Machine Learning-Augmented Algorithms: Theory and Applications in Energy and Sustainability

Thesis by
Nicolas Henry Christianson

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

Caltech

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2025
Defended May 14, 2025

Nicolas Henry Christianson
ORCID: 0000-0001-8330-8964

ACKNOWLEDGEMENTS

The work in this thesis would not have been possible without the mentorship, collaboration, and support of many people.

I am most grateful to my advisors, Adam Wierman and Steven Low, for their generous support and mentorship during the last five years. Adam and Steven represent academia at its finest—they are uncompromisingly supportive of their students, deeply knowledgeable and yet perpetually curious, and passionate about driving real-world impact through their work. Throughout my graduate study, Adam's constant presence as a mentor, keen eye for interesting research questions, and creativity in translating insight into impact have been instrumental to my success as a graduate student, and have profoundly shaped how I understand the endeavor of building a sustainable and exciting research program. Likewise, Steven's deep expertise in energy systems and eagerness to offer his time—whether to provide general advice, or to dive into the mathematical details of whatever problem I find myself working on—have been invaluable to the development of my research trajectory during my PhD. It has been a distinct privilege to be their student, and I can only aspire to one day become as effective and supportive an advisor as they have been.

I have been very lucky to collaborate with many wonderful and talented people during my PhD, both at Caltech and at other institutions. During my first year at Caltech, Lucien Werner involved me in some discussions about robust multi-stage electricity markets, and the ideas we discussed soon blossomed into my first paper; I am very grateful for the time he spent to help me orient myself in the world of power systems and graduate research. Around the same time, I began working with Daan Rutten and Debankur Mukherjee at Georgia Tech on the design of learning-augmented algorithms for smoothed online optimization. This was a very fun collaboration, and crucially, it gave me the opportunity to build the conceptual foundation in online and learning-augmented algorithms that would serve me well for many more projects during my PhD.

One of the most formative—and productive—collaborations during my PhD was with Adam Lechowicz and Mohammad Hajiesmaili (among others) at UMass Amherst. The several papers we have written together over the last few years have been a lot of fun to work on, and it's been very exciting to work together to bridge learning-augmented algorithm design and sustainable datacenter operation. I'm also very grateful for the times they've hosted me in Amherst for visits and talks—these visits have been both highly productive and a wonderful excuse to escape the heat of California for the autumnal beauty of New England.

I must thank Weiwei Yang for hosting me in Redmond for an internship at Microsoft Research during the summer of 2023, and Baosen Zhang for his mentorship and collaboration on the project I began there. That summer marked a critical turning point in my research agenda as I began to think more about questions related to risk, uncertainty, and reliability in machine learning. My time in Redmond gave me the opportunity to take a step back, read and think deeply, and explore these new themes that would characterize the second half of this thesis.

Beyond these general themes, the second half of my PhD was largely characterized by collaborations with Christopher Yeh. Chris's interest and expertise in uncertainty quantification was influential in my beginning to explore these areas, and I have had a lot of fun working (and living) with him over the last few years. Chris has always been willing to listen to my worries, productively critique my research ideas, and work through the night together to make a paper deadline. He is a deeply curious person, a true believer in the cause of sustainability, and a wonderful friend.

I must also acknowledge Bo Sun, whose familiarity with online conversion problems has been of great help for several papers we have worked on together; Eitan Levin, with whom I have not coauthored, but whose mathematical expertise has been invaluable whenever I find myself out of my depth; and the many other individuals that I collaborated with or otherwise discussed research with during my time at Caltech, who are too numerous to list here.

My partner, Sophie Pesek, has been a constant source of love and support throughout the highs and lows of my PhD, from starting graduate school remotely during COVID, to the busyness and stress of the job market. Through it all, she has been a voice of reason, helping me work through challenges, sending me snacks during late nights, and proofreading my papers. Perhaps the most important lesson she has taught me is that the solution to most roadblocks in research is to go for a hike and look at the problem with fresh eyes later. Her commitment to working on important problems that serve the earth and its people is a daily inspiration to me, and I aspire to bring such a clarity of purpose to my own work.

Finally, none of the work in this thesis or anything that preceded it would have been possible without the love, support, and guidance of my family. My parents were, and continue to be, my primary inspiration for pursuing a scientific career, and they have supported my journey at each step of the way, even when it involved moving to the West Coast for my PhD. My father, David, is my role model for how to build and pursue one's own research agenda, striving to advance the frontiers of knowledge, while doing it with a pure love for the science. My mother, Anastasia, embodies a persistence in the face of adversity, a willingness to embrace new challenges, and a focus on the human impact of her work which I aspire to match. My sister, Karen, serves as my inspiration for taking ownership of one's path and authentically exploring one's interests, regardless of how untrodden the path may be. I have become the person I am today because of them, and their example will continue to inspire me for many years to come.

ABSTRACT

Uncertainty poses a significant challenge for decision-makers in energy and sustainability domains. The ongoing energy transition—characterized by increasing penetrations of variable renewable generation, deployment of novel grid assets like battery energy storage systems, and growing risks from climate-driven natural disasters—introduces new, multifaceted uncertainties that traditional operational methods struggle to accommodate. While artificial intelligence (AI) and machine learning (ML) hold significant promise for navigating this transition and improving the efficiency of energy system operation, their direct deployment to high-stakes energy and sustainability problems presents substantial risks. In particular, current AI/ML tools typically lack guarantees on reliability, robustness, and safety, and thus pose a risk of poor performance or catastrophic failure if deployed in the real world. To make progress on decarbonization while maintaining reliability, new approaches are needed to enable the design of AI- and ML-augmented algorithms that achieve near-optimal performance while providing rigorous guarantees on robustness and reliability when deployed in real-world energy and sustainability problems.

This thesis addresses this challenge from two complementary perspectives, seeking to bridge the gap between theoretical algorithmic insights and practical impact. In the first part, we develop *learning-augmented algorithms* that integrate black-box AI/ML "advice" into online optimization problems while ensuring provable, worst-case performance guarantees. We propose algorithms for several classes of problems—including cases with convex costs, nonconvex costs, and long-term deadline constraints—that obtain the provably optimal tradeoff between exploiting good AI performance and worst-case robustness. We demonstrate these algorithms' ability to improve operational efficiency in energy and sustainability domains through case studies on cogeneration power plant operation under high renewables penetration and carbon-aware workload shifting for geographically-distributed datacenters.

In the second part of this thesis, we move beyond the "black box" model of AI/ML to explore how risk-awareness and reliability can be integrated as primary design criteria in AI/ML model training and algorithm development more generally. We consider this objective along several avenues, introducing new theoretical and methodological approaches for risk-aware optimization and uncertainty quantification, designing new mechanisms for pricing general forms of uncertainty in electricity markets, and developing new frameworks for training machine learning models

with provable reliability guarantees. Throughout, we emphasize connections with and applications to energy and sustainability problems ranging from grid-scale battery storage operation to power grid contingency analysis. Together, these approaches highlight the challenges facing and benefits to risk- and reliability-aware learning and decision-making.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] N. Christianson, T. Handina, and A. Wierman, "Chasing Convex Bodies and Functions with Black-Box Advice," in *Proceedings of the Thirty Fifth Conference on Learning Theory*, PMLR, Jun. 2022, pp. 867–908. [Online]. Available: https://proceedings.mlr.press/v178/christianson22a.html.
N.H.C. led the conception of the project and the development of the theoretical results and algorithms, and led the writing of the manuscript.

[2] D. Rutten, N. Christianson, D. Mukherjee, and A. Wierman, "Smoothed Online Optimization with Unreliable Predictions," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 7, no. 1, 12:1–12:36, Mar. 2023. DOI: 10.1145/3579442. [Online]. Available: https://dl.acm.org/doi/10.1145/3579442.
N.H.C. participated in the conception of the project and development of the theoretical results and algorithms, conceived and conducted the simulations, and participated in the writing of the manuscript.

[3] N. Christianson, J. Shen, and A. Wierman, "Optimal Robustness-Consistency Tradeoffs for Learning-Augmented Metrical Task Systems," in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, PMLR, Apr. 2023, pp. 9377–9399. [Online]. Available: https://proceedings.mlr.press/v206/christianson23a.html.
N.H.C. led the conception of the project and the development of the theoretical results and algorithms, and led the writing of the manuscript.

[4] N. Christianson, C. Yeh, T. Li, M. Hosseini, M. T. Rad, A. Golmohammadi, and A. Wierman, "Robust Machine-Learned Algorithms for Efficient Grid Operation," *Environmental Data Science*, vol. 4, e24, Apr. 2025, ISSN: 2634-4602. DOI: 10.1017/eds.2024.28. [Online]. Available: https://www.cambridge.org/core/journals/environmental-data-science/article/robust-machinelearned-algorithms-for-efficient-grid-operation/0E29B936A8FEE565B169F70372B7F9DE.
N.H.C. led the conception of the project and development of the theoretical results and algorithms, conceived and conducted the simulations, and led the writing of the manuscript.

[5] A. Lechowicz, N. Christianson, J. Zuo, N. Bashir, M. Hajiesmaili, A. Wierman, and P. Shenoy, "The Online Pause and Resume Problem: Optimal Algorithms and An Application to Carbon-Aware Load Shifting," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 7, no. 3, 45:1–45:32, Dec. 2023. DOI: 10.1145/3626776. [Online]. Available: https://doi.org/10.1145/3626776.
N.H.C. participated in the conception of the project and development of the

theoretical results and algorithms, participated in the development of the simulations, and participated in the writing of the manuscript.

[6] A. Lechowicz, N. Christianson, B. Sun, N. Bashir, M. Hajiesmaili, A. Wierman, and P. Shenoy, "Learning-Augmented Competitive Algorithms for Spatiotemporal Online Allocation with Deadline Constraints," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 9, no. 1, 8:1–8:49, Mar. 2025. DOI: 10.1145/3711701. [Online]. Available: https://dl.acm.org/doi/10.1145/3711701.
N.H.C. participated in the conception of the project and development of the theoretical results and algorithms, and participated in the writing of the manuscript.

[7] N. Christianson, B. Sun, S. Low, and A. Wierman. "Risk-Sensitive Online Algorithms." Accepted for Presentation at the Conference on Learning Theory 2024: https://proceedings.mlr.press/v247/christianson24a.html. arXiv: 2405.09859 [cs], [Online]. Available: http://arxiv.org/abs/2405.09859.
N.H.C. led the conception of the project and development of the theoretical results and algorithms, and led the writing of the manuscript.

[8] C. Yeh*, N. Christianson*, A. Wu, A. Wierman, and Y. Yue. "End-to-End Conformal Calibration for Optimization Under Uncertainty." arXiv: 2409.20534 [cs, math], [Online]. Available: http://arxiv.org/abs/2409.20534.
N.H.C. co-led the conception of the project and development of the theoretical results and algorithms, participated in the conception and conduction of the simulations, and co-led the writing of the manuscript.

[9] L. Werner*, N. Christianson*, A. Zocca, A. Wierman, and S. Low, "Pricing Uncertainty in Stochastic Multi-Stage Electricity Markets," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, Dec. 2023, pp. 1580–1587. DOI: 10.1109/CDC49753.2023.10384022. [Online]. Available: https://ieeexplore.ieee.org/document/10384022.
N.H.C. co-led the conception of the project and development of the theoretical results and algorithms, and co-led the writing of the manuscript. © 2023 IEEE.

[10] N. Christianson, W. Cui, S. Low, W. Yang, and B. Zhang, "Fast and Reliable $N - k$ Contingency Screening with Input-Convex Neural Networks," in *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*, PMLR, Jun. 2025, pp. 527–539. [Online]. Available: https://proceedings.mlr.press/v283/christianson25a.html.
N.H.C. led the conception of the project and development of the theoretical results and algorithms, conceived and conducted the simulations, and led the writing of the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

Uncertainty poses a pervasive and growing challenge in the modern world, with volatility, emerging risks, and complex, unpredictable system dynamics impacting domains ranging from financial markets to supply chains. In few areas are the challenges presented by uncertainty so evident as in energy and sustainability: decision-making entities like energy system and resource operators must plan for and operate in the face of high-dimensional, multi-faceted uncertainties like short- and long-term trends in electricity demand, natural disaster risks, and market participants' strategic behavior. The threat of climate change and the ongoing transition to a decarbonized economy intensify these challenges: numerous factors, including a dramatic increase in the penetration of nondispatchable, variable renewable generation like wind and solar, emergence of novel asset types such as grid-scale battery storage, increasing electricity demand due to electrification, and the rising risks of wildfires and other natural disasters, all drive increased complexity and uncertainty [1–3]. This poses a significant challenge for decision-makers of many kinds—including small-scale energy resource operators, large-scale grid operators, and operators of other energy-intensive infrastructure like datacenters—who must plan and operate their systems as efficiently as possible in the face of these growing uncertainties while meeting sustainability goals and continuing to ensure system reliability.

The recent groundbreaking developments in Artificial Intelligence (AI) and Machine Learning (ML) hold significant promise for addressing this growing uncertainty and improving decision-making performance. Indeed, AI and ML have achieved significant improvements upon state-of-the-art and human-level performance in application domains such as autonomous driving [4] and game playing [5–7], and AI tools are used widely in real-world use cases including programming [8] and social media content moderation [9]. These advancements in AI and ML are also beginning to make an impact in the domains of energy and sustainability: real-world deployments of AI have been used to optimize the energy efficiency of data center cooling systems [10] and the profitability of grid-scale battery energy storage systems [11]. In addition, multiple regional power system operators in the United

States and abroad are actively exploring its application to large-scale power grid planning and operation [12–15].

However, naively deploying AI and ML to high-stakes, safety-critical, and societally-impactful domains such as energy and sustainability comes with significant risks. Modern AI and ML tools have failure modes which are often not well understood [16], including susceptibility to adversarial attacks [17], poor generalization when faced with distribution shift and new problem domains [18, 19], and a propensity to hallucinate false and potentially dangerous information [20]. Such failures could cause very costly and even catastrophic downside risks in energy and sustainability applications, such as spikes in cost or emissions, constraint violations, or even blackouts. While a great deal of effort has been expended in recent years to understand the mechanisms underlying these failure modes and develop new AI and ML models with improved safety and reliability, there remains a critical lack of algorithmic and methodological approaches enabling both *rigorous, theoretical* guarantees on the safety and reliability of AI and ML in high-stakes application domains, as well as *near-optimal performance* of these models for decision-making objectives. The lack of such principled, reliable, and optimal methods for training and deploying AI and ML models significantly hinders the deployment of AI for real-world problems in energy and sustainability; while leveraging AI in these applications could enable improved efficiency and a more rapid energy transition, AI lacks the requisite guarantees for these applications, where reliability is paramount.

Moreover, the growing appetite for larger and better AI models is currently exacerbating existing challenges in energy and sustainability. The training and deployment of these growing models is expected to fuel an exponential increase in the electricity consumption of datacenters over the next decade [21]. Such a dramatic increase in electricity demand risks progress on decarbonizing the electricity sector, as power system operators must build and maintain fossil fuel generation assets to keep up with demand [22]. The rapid growth in datacenter demand has sustainability ramifications beyond just carbon emissions: modern datacenters consume a significant volume of water for cooling [23], and power they consume both from grid assets and local backup generation can emit harmful air pollutants [24]. This feedback loop—whereby the rise of AI accentuates our current energy and sustainability challenges—makes clear the need for new tools to help power system and datacenter operators absorb this increased demand through more efficient utilization of renewable energy and existing grid assets. While AI itself can help toward this end, it must be developed and deployed in a reliable manner to avoid further intensifying these challenges.

Thus, to realize the promise of AI and ML as transformative technologies while addressing the pressing societal challenges posed by energy and sustainability, there is an imminent need to develop new, theoretically-grounded approaches for training and deploying *reliable* AI and ML tools. In particular, a new generation of *AI-* and *ML-augmented algorithms* is needed to bridge the divide between the excellent performance of modern AI/ML tools and the rigorous, theoretical guarantees on reliability offered by classical algorithms and decision-making frameworks in these high-stakes energy and sustainability problems.

1.1 Challenges and Prior Work

There has been a great deal of progress in recent years on the development of fundamental algorithms and mechanisms for the planning and operation of sustainable systems in energy and beyond, and new techniques to train and deploy AI and ML tools in these applications. However, significant challenges hinder the reliable, widespread deployment of AI and ML in such real-world, high-stakes, and safety-critical energy and sustainability tasks. This is due both to the fundamental difficulty of the underlying problems—which often exhibit complex, constrained structure—as well as fundamental technical challenges that arise in the design of algorithms and AI/ML-driven tools with rigorous guarantees in these settings.

Unifying Challenges in Energy and Sustainability:
Hard Constraints, Intertemporal Coupling, and Operational Reliability

Many problems in energy systems and sustainability exhibit complex, constrained problem structure that significantly complicates their efficient and reliable solution. For instance, energy grid operation problems feature many different kinds of constraints, including nonconvex power flow feasibility constraints [25], high-dimensional security and contingency constraints ensuring resilience to asset failures [26, 27], and intertemporal constraints that bind decisions across time, like ramp constraints [28–30], state-of-charge constraints for energy storage [31], and deadline constraints for deferrable loads [32–34]. In addition, problems like energy resource operation and datacenter operation often involve costs which depend on decisions made across multiple timesteps, such as ramp costs for modifying decisions or turning assets on and off [35–38].

This complex structure poses two distinct types of challenge for reliable system operation. First, it may be computationally difficult to obtain a decision that feasibly satisfies the desired constraints; for example, the high-dimensional structure of the constraint set for security-constrained optimal power flow quickly leads to intractability for large-scale power grids [26]. Second, the coupling between uncertainty and the intertemporal structure of these problems poses its own set of challenges. This structure means that earlier decisions can significantly influence the cost or feasibility of later decisions, even while future conditions (such as renewable energy availability) are not yet known. Amid this uncertainty, decision-makers face strict reliability needs of multiple kinds: assets must be dispatched feasibly, sufficient energy supply must be available to meet demand, and workload or electric load deadlines must be met. As such, decision-makers must carefully quantify and plan for future uncertainty during operation, to ensure that their current decisions do not lead to higher costs or infeasible operating conditions in the future. They must also plan for many kinds of uncertainty, ranging from short-term electricity price and demand variability to longer-term trends in the grid makeup, consumer energy demand, and natural disaster risk.

Many different approaches have been proposed to address these challenges, including heuristic methods and convex relaxations to solve computationally difficult power systems problems [39, 40], numerous frameworks in robust and stochastic optimization to ensure feasibility and control risk in the face of uncertainty [41–49], and online algorithms that accommodate intertemporal structure like ramp costs and constraints [29, 30, 37, 38]. However, these kinds of approaches are limited by their conservativeness: to ensure robust and reliable operation, they must generally sacrifice performance. Furthermore, they are typically not designed to fully take advantage of real-world data.

Data-driven AI and ML techniques have the potential to significantly improve performance in these problems, and have already seen great success in applications such as grid carbon intensity forecasting [50, 51], contingency screening [52, 53], and power flow optimization [54–57]. However, the previously-mentioned failure modes of AI and ML also pose significant risks for their deployment to real-world energy and sustainability applications, with adversarial attacks [58], distribution shift [59], and hallucination [60] all posing their own risks for feasibility and performance. These risks are particularly significant given the societally-critical nature of the energy domain. Thus, to take advantage of the typically excellent performance of AI/ML tools in energy and sustainability problems, we must ensure that they can be deployed with rigorous, provable guarantees on reliability.

One can employ two approaches to this end. On the one hand, one might seek to take advantage of existing, already-trained AI/ML tools, and process them through a separate algorithm to *"robustify"* their behavior. On the other hand, one might instead seek to train new models which are *reliable by design*, promoting some desired reliability notion to a primary design criterion during model training. This thesis will consider both approaches, and we will briefly discuss each of these two paradigms in the following sections.

Leveraging Untrusted, Black-Box AI/ML for Decision-Making

Ongoing developments in AI and ML have led to a wealth of models achieving excellent performance for many different applications; of particular note are large foundation models which can perform well *across* domains. However, nearly all of these models are "black boxes," in the sense that very little can be understood about how they produce decisions and their failure modes. Recent work has sought to understand these failure modes better through means such as empirical and theoretical analyses of adversarial robustness [61, 62], and to propose new methods to train models that are robust to adversarial perturbations [63, 64], and distribution shift [65]. These training methodologies cannot in general *guarantee* that the resulting model produces decisions that are reliable in a worst-case sense. While tools like neural network verification [66–68] can rigorously certify such reliability in some applications, these tools cannot scale to the general forms of multi-stage, online, and constrained decision-making problems in energy and sustainability to which we might seek to apply AI and ML.

A recent line of work on *algorithms with predictions*, or *learning-augmented algorithms*, has sought to investigate how such untrusted, black-box AI and ML models can be effectively utilized for decision-making problems. In this paradigm, a decision-maker seeks to exploit the predictions or "advice" offered by a model when they are useful, obtaining cost close to that of the AI/ML model in this case—i.e., *consistency*—while maintaining rigorous, worst-case performance guarantees if the model performs poorly—called *robustness*. The design of robust and consistent algorithms was first studied by [69] for the online caching problem, and by [70] for the problems of ski rental and non-clairvoyant job scheduling. Since these initial works, more than 200 papers have been written on the design and analysis of learning-augmented algorithms for online and offline problems including peak-aware energy scheduling [71], mechanism design [72], bipartite matching [73], and electric vehicle charging [74].[1]

However, there are significant challenges facing the development of learning-augmented algorithms for more general decision-making problems. Specifically, in complex, high-dimensional, and multi-stage decision-making problems—representative of important energy and sustainability problems like energy resource dispatch—it is challenging to design algorithms that *optimally* trade off between consistency (exploiting AI/ML advice) and robustness (worst-case performance). For instance, prior to the work presented in this thesis, previous learning-augmented algorithms for the *metrical task systems* problem, a general form of online optimization with switching costs, simply applied off-the-shelf algorithms that do not exploit problem structure [75]. As such, these previous approaches are suboptimal, and cannot fully leverage the power of good AI/ML performance. Thus, a key challenge for such general problems is designing algorithms that *leverage problem structure* to obtain better performance bounds.

Another significant challenge is the design of learning-augmented algorithms which can accommodate complex *intertemporal structure*, such as problems with both long-term deadline constraints and switching/ramp costs. This is a challenge for online algorithm design even without considering the incorporation of AI/ML advice: prior work in the literature on the design and competitive analysis of algorithms has only considered either switching costs [76, 77] or long-term deadline constraints [78, 79]. However, these features are both critical for practical applications like spatial and temporal load shifting for sustainable datacenters. Thus, designing algorithms which can make optimal decisions in the face of this varied problem structure, and that can furthermore leverage the advice of AI/ML while maintaining worst-case guarantees, is an important problem.

Risk-, Uncertainty-, and Reliability-Aware Methods

The approaches discussed in the previous section are useful, as they enable robustly leveraging black-box AI/ML models for decision-making; in particular, *any* existing AI/ML model can be used as the “advice” in a learning-augmented algorithm, and no *a priori* guarantees are needed. However, these approaches still introduce some conservativeness, as they require post-processing the model or its outputs in

[1] See https://algorithms-with-predictions.github.io/ for a repository of papers related to this framework.

some fashion to obtain the desired, e.g., robustness guarantee. To obtain optimal performance, it would be better to train a new, problem-specific AI/ML model to be *reliable by design* by, e.g., enforcing some notion of model reliability during training. Such enforcement can often be done in an approximate fashion: for instance, constraint satisfaction can be promoted by adding a penalty to the training loss, and adversarial robustness can be promoted via adversarial training [63]. Such notions of reliability can also, in some cases, be enforced *provably* during training through means like feasibility enforcement layers [56, 80] and certified training [64, 81, 82]. However, for broader notions of reliability such as uncertainty calibration, risk control, and satisfaction of more general constraints, provably enforcing these properties during training remains an open challenge. New strategies to this end would enable the design of new methods for learning over the set of *provably reliable models*, allowing for better performance in applications where both performance and reliability are critical.

This particular challenge in learning evokes a complementary, yet broader theme spanning beyond AI and ML: how should we integrate risk, uncertainty, and reliability as first-class design criteria when developing algorithms, mechanisms, and decision-making frameworks? And how do these considerations impact the design and analysis of optimal strategies? This is a rich theme that connects to many different problems, including risk-sensitive (online) learning [83–87], risk- and uncertainty-aware optimization and control [46, 47, 49, 88, 89], and uncertainty quantification [90–92]. Despite the wide range of prior research on this general theme, many questions remain open in specific applications regarding how to optimally *utilize* and *account for* risk and uncertainty in decision-making problems.

1.2 Contributions of This Thesis

Motivated by the aforementioned challenges, this thesis considers the following central question:

How can we design theoretically-grounded *AI- and ML-augmented algorithms* to enable the safe and reliable deployment of modern AI/ML tools to critical problems in energy and sustainability?

We structure our investigation in two parts. The first considers the design of *learning-augmented algorithms* which leverage the advice of black-box AI and ML tools to improve performance while maintaining worst-case guarantees for a collection of general online optimization problems. The second part takes a broader view of algorithms, machine learning, and reliability, considering both the design of risk-, uncertainty-, and reliability-aware machine learning models, as well as algorithmic questions around the integration of risk and uncertainty in online decision-making and electricity market operation.

Part I: Algorithms with Black-Box AI/ML Advice

In the first part of this thesis, we examine the question of designing *learning-augmented algorithms*, or algorithms that leverage the advice of potentially unreliable "black-box" AI or ML models for online decision-making problems while maintaining worst-case performance guarantees. Specifically, we seek algorithms with two kinds of performance guarantees: *robustness*, a worst-case multiplicative cost guarantee relative to the offline optimal algorithm (i.e., a competitive ratio), and *consistency*, a multiplicative cost guarantee relative to the performance of the AI/ML model itself. We consider the design of robust and consistent algorithms in several problem settings built on the general theme of online optimization with switching costs; for each setting, new algorithmic insights are needed to enable optimal performance. We motivate and evaluate our algorithms throughout with applications in energy and sustainability.

We begin this agenda in Chapter 2 by considering the problem of convex function chasing, where an online decision-maker seeks to minimize the total cost of making and switching between decisions in a normed vector space. This problem framework models general classes of online optimization problems with ramping costs such as datacenter operation, where servers can turn on and off (at some expense, due to overhead) in order to meet workload demand and minimize energy expenditure [37, 38, 93]. The decision-maker seeks to obtain cost close to the advice of a black-box AI/ML algorithm when it performs well—i.e., consistency—as well as worst-case robustness when the advice performs poorly.

We begin by considering learning-augmented algorithms that deterministically "switch" between following the decisions of the advice and those of a chosen baseline algorithm. We show that, in general, no algorithms in this class can simultaneously be robust while obtaining consistency less than 3; that is, any switching algorithm that has bounded worst-case robustness must pay at least 3 times the cost of the advice in the worst case. To break through this fundamental limit, we propose three novel algorithms that exploit the problem's convexity to enable better performance.

The first, AOBD, achieves a consistency of $(1+\epsilon)$ and robustness of $2+\frac{2}{\epsilon}$ when the decision space is the real line $\mathbb{R}$, which we prove is the optimal tradeoff amongst *all* algorithms in this setting. We then move to higher-dimensional decision spaces, proposing an algorithm, INTERP, that achieves $(\sqrt{2}+\epsilon)$-consistency and $O(\frac{C}{\epsilon^2})$-robustness for any $\epsilon > 0$, where C is the competitive ratio of any chosen baseline algorithm for convex function chasing or a special case thereof. Finally, we propose an algorithm BDINTERP that achieves $(1+\epsilon)$-consistency and $O(\frac{CD}{\epsilon})$-robustness when the decision space has diameter bounded by D.

In Chapter 3, we go beyond the convex setting of the previous chapter, considering the design of learning-augmented algorithms for *metrical task systems* (MTS), a broad generalization of convex function chasing where cost functions can be nonconvex and the switching costs are a general metric. In this setting, we propose a randomized algorithm, DART, that for any $\epsilon > 0$, obtains $(1+\epsilon)$-consistency together with a robustness of $2^{O(1/\epsilon)}$ relative to any chosen baseline algorithm. We further prove that this exponential tradeoff is necessary: *any* $(1+\epsilon)$-consistent algorithm for learning-augmented MTS must have a robustness of at least $2^{\Omega(1/\epsilon)}$. However, we show that in several important special cases of MTS, DART achieves better robustness: when the metric space has bounded diameter D, DART achieves robustness $\frac{1}{\epsilon}$ relative to the baseline algorithm (with an additive term of $\frac{D}{\epsilon}$), and in the setting of the celebrated k-server problem, DART can achieve robustness $\frac{k}{\epsilon}$. Notably, these results follow from specialized analyses, and do not require any modification of the DART algorithm itself.

Because DART is a randomized algorithm, we then turn to the question of whether we can design a deterministic algorithm to match its performance. We propose a new algorithm, DETROBUSTML, that essentially matches the robustness-consistency tradeoff achieved by DART when provided with an *a priori* bound on the diameter of the decision space; however, DETROBUSTML (and, in fact, any deterministic algorithm) cannot obtain the diameter-independent optimality guarantees achieved by DART. We conclude the chapter with a case study, evaluating both DART and DETROBUSTML on a realistic model of cogeneration power plant operation on high-renewables power grids. Our experimental results demonstrate the substantial value of our learning-augmented algorithms to bridge the excellent performance of machine-learned approaches with the reliability of standard dispatch methods.

In Chapter 4, we take a step back from learning-augmented algorithm design, and instead focus on a more fundamental question in online algorithm design.

Specifically, we ask: how can we design optimal algorithms for online optimization with switching costs when there is a *deadline constraint* present? This problem is motivated by the general task of carbon-aware temporal load shifting, where a datacenter operator may choose to pause a workload during times when the local grid energy has high carbon intensity, in order to take advantage of lower-carbon energy in the future. In this setting, there is both an overhead cost for saving or restoring the state of a workload—a switching cost—as well as a deadline by which time the workload must be completed—a deadline constraint. While previous online algorithms problems have involved either of these features independently, their combination presents unique challenges for optimal algorithm design.

To this end, we introduce the "online pause and resume" problem, where a decision-maker is faced with a sequence of prices of length T, and after each price has been revealed must decide whether to accept it. The decision-maker's goal is to choose the k lowest (or highest) prices, while also accounting for the cost of switching between decisions (i.e., each time they start or stop purchasing/selling). In this problem, the prices represent the carbon intensity of the grid, the switching costs represent the overhead of starting or stopping the workload, and k represents the size of the workload that must be completed by time T. We propose a framework of *double-threshold* algorithms for this problem, which utilize two distinct threshold rules to decide whether a decision-maker should accept or reject a price depending on what their previous decision was. We further show that these algorithms, in both the minimization and maximization variants of the problem, obtain the *provably optimal* competitive ratio amongst all deterministic algorithms. We conclude the chapter with a case study applying our double threshold algorithmic framework to a model of carbon-aware temporal workload shifting with real carbon intensity traces, showing that our approach improves significantly upon existing baselines.

We conclude the first part of this thesis with Chapter 5, which returns to the design of learning-augmented algorithms in a problem setting that builds on the previous chapters. Specifically, this chapter is motivated by the challenge of carbon-aware spatiotemporal workload shifting in datacenters, where a company operating geographically distributed datacenters seeks to shift a workload across time and between regions to take advantage of low-carbon energy, while accommodating both the overhead costs of load shifting (both temporally and spatially), and the workload deadline. To model this application, we introduce the problem of spatiotemporal online allocation with deadline constraints (SOAD), in which a decision-maker seeks to complete a workload with deadline T by allocating it amongst the points of a general metric space (X, d). At each time, the decision-maker is faced with a cost function representing the cost to service the workload at each point of the metric space (e.g., the carbon intensity at each datacenter location), and they must decide where to shift the workload, incurring a switching cost $d(\cdot, \cdot)$ that reflects the overhead of both local (temporal) shifting and global (spatial) shifting. This problem setting unites the general metric space setting of MTS considered in Chapter 3 and the deadline-constrained structure considered in Chapter 4, and formalizes the open challenge of designing algorithms for this more complex setting.

We first propose a randomized pseudo-cost minimization algorithm (PCM) for SOAD, which chooses decisions at each time by minimizing the difference between the instantaneous cost and a carefully-designed *pseudo-cost* term, which promotes satisfaction of the deadline constraint over time. We show that this algorithm obtains a competitive ratio that is optimal up to a logarithmic factor in the size of the metric space X, which results from the use of randomized metric embeddings to accommodate general metrics. We then propose a learning-augmented algorithm, ST-CLIP, which builds on this pseudo-cost minimization framework by including a *consistency constraint* that enables taking advantage of black-box AI/ML advice. We prove that ST-CLIP obtains the optimal tradeoff between consistency and robustness up to a logarithmic factor in the size of the metric space X, again resulting from the metric embedding. Finally, we perform extensive evaluations of our algorithms in a case study on carbon-aware spatiotemporal workload shifting leveraging real data on carbon traces, cluster traces, and throughput between datacenters. Our experimental results demonstrate the significant value of learning-augmented algorithms for reducing the carbon emissions of compute workloads.

Part II: Beyond the Black Box: New Frontiers in Uncertainty, Risk, and Reliability

In the second part of this thesis, we move beyond the black-box AI/ML advice framework and consider the question of designing *risk-*, *uncertainty-*, and *reliability-aware* machine learning models and algorithms. In particular, we are motivated by two complementary, yet related questions. First, how should we train machine learning models in a manner that enforces reliability, risk, and uncertainty guarantees? Second, how can we optimally incorporate risk and uncertainty into algorithms and mechanisms for online and multi-stage decision-making? Though the latter question is not directly related to the broader theme of machine learning, it has important implications for the *downstream* application of machine learning tools resulting from the former question—for instance, how to incorporate the uncertainty estimates of a machine learning model into online decision-making. While we cannot possibly address all of the possible connections between these two avenues here, in this part we will survey several research directions that they motivate, drawing attention to potential connections where they arise.

We begin this part in Chapter 6, where we study the design of *risk-sensitive online algorithms*, with risk-sensitive notions of performance used in their design and competitive analysis. This paradigm is motivated in part by the *randomized* learning-augmented algorithms presented in Chapters 3 and 5, and the question of whether such randomization can expose decision-makers to potential downside risk in the form of large costs (even with small probability). To this end, we introduce the CVaR_δ-competitive ratio (δ-CR) using the conditional value-at-risk of an algorithm's cost, which measures the expectation of the $(1-\delta)$-fraction of worst outcomes against the offline optimal cost, and use this measure to study three online optimization problems: continuous-time ski rental, discrete-time ski rental, and one-max search. These problems, which are prototypical problems in online optimization, serve as building blocks for more complex problems like MTS, and also have connections with real applications such as dynamic power management [94], peak-aware economic dispatch in microgrids [95], and energy trading [96].

We develop optimal and near-optimal algorithms for each of these problems, finding that the structure of the optimal δ-CR and algorithm varies significantly between problems. We first prove that the optimal δ-CR for continuous-time ski rental is $2 - 2^{-\Theta(\frac{1}{1-\delta})}$, obtained by an algorithm described by a delay differential equation. In contrast, in discrete-time ski rental with buying cost B, there is an abrupt phase transition at $\delta = 1 - \Theta(\frac{1}{\log B})$, after which the classic deterministic strategy is optimal. Similarly, one-max search exhibits a phase transition at $\delta = \frac{1}{2}$, after which the classic deterministic strategy is optimal; we also obtain an algorithm that is asymptotically optimal as $\delta \downarrow 0$ that arises as the solution to a delay differential equation. These results highlight a fundamental limit to the value of randomization when decision-makers are risk-sensitive, and serve as a theoretical foundation for potential future work integrating risk senstivity into learning-augmented algorithms in more general settings.

Whereas Chapter 6 focuses primarily on how to make optimal decisions with risk-sensitive objectives, Chapter 7 considers the complementary question of how to best *quantify uncertainty* for optimal decision-making. Ensuring robust performance in risk-aware decision-making problems requires well-calibrated estimates of uncertainty, which can be difficult to achieve with neural networks. While a number of uncertainty quantification methods have been proposed in recent years to enable the calibration of machine-learned uncertainty estimates, these estimates are still typically learned in a manner that is *separate* from the downstream decision-making problem. That is, conventional approaches for decision-making and optimization under uncertainty typically separate the *estimation* of uncertainty from the *optimization* using it. This significantly hinders performance in high-dimensional settings, where there can be many valid uncertainty estimates, each with its own performance profile—i.e., not all uncertainty is equally valuable for downstream decision-making.

To address this challenge, Chapter 7 develops an end-to-end framework to learn uncertainty sets for conditional robust optimization problems in a way that is informed by the downstream decision-making loss, with robustness and calibration guarantees provided by conformal prediction. The end-to-end nature of this methodology ensures that uncertainty sets are both calibrated, and allowed to focus on regions of the parameter space that matter the most for decision-making performance. We specifically propose to represent general families of convex uncertainty sets with partially input-convex neural networks, which are learned and calibrated as part of our framework. We perform extensive experiments comparing our end-to-end framework against conventional two-stage "estimate-then-optimize" methods on the problems of energy storage arbitrary and portfolio optimization, finding that our framework yields significantly improved performance while maintaining the uncertainty calibration guarantees needed for robust operation.

In Chapter 8, we move from the machine learning questions of the previous chapter to their downstream implications for the operation of energy systems. Specifically, we ask: given an uncertainty set (say, for example, learned in the end-to-end fashion proposed in the previous chapter) that we would like to use for robust electricity market dispatch, or given some other risk-sensitive or stochastic dispatch method, how should we design prices to support an efficient market equilibrium? To this end, we propose a pricing mechanism for multi-stage electricity markets that does not explicitly depend on the choice of dispatch procedure or optimization method. Our approach can accommodate a wide range of methodologies for power system economic dispatch under uncertainty, including multi-interval dispatch, multi-settlement markets, scenario-based dispatch, chance-constrained dispatch policies, and robust optimization-based dispatch. We prove that our pricing scheme provides both *ex-ante* and *ex-post* dispatch-following incentives for participants by simultaneously supporting per-stage and ex-post competitive equilibria. In numerical experiments on a ramp-constrained test system, we demonstrate the benefits of scheduling under uncertainty and show how our price decomposes into components corresponding to energy, intertemporal coupling, and uncertainty.

Finally, in Chapter 9, we return to the theme of reliability-aware learning, motivated by the problem of contingency screening in power grids. Power system operators must ensure that dispatch decisions remain feasible in case of grid outages, or *contingencies*, to prevent cascading failures and ensure reliable operation. However, checking the feasibility of all $N - k$ contingencies—every possible simultaneous failure of k grid components—is computationally intractable even for small k. As such, system operators must use heuristic screening methods that might not include all relevant contingencies, which can generate false negatives where unsafe scenarios are misclassified as safe. In this final chapter, we propose to use input-convex neural networks (ICNNs) for contingency screening. We show that ICNN reliability can be determined by solving a convex optimization problem, and by scaling model weights using this problem as a differentiable optimization layer during training, we can learn an ICNN classifier that is both data-driven and has provably guaranteed reliability—i.e., that guarantees a zero false negative rate. We evaluate this methodology in a case study of $N-2$ contingency screening on the IEEE 39-bus test network, where it yields substantial (10-20 $\times$) speedups over exhaustive contingency screening while maintaining excellent classification accuracy. We further show that the learned ICNNs can also be used to speed up the solution of the security-constrained DC optimal power flow problem, providing comparable speedups while maintaining excellent cost and feasibility.

Notably, the training methodology we propose in this final chapter is similar in spirit to that proposed for end-to-end learning of calibrated uncertainty sets in Chapter 7. Both frameworks employ differentiable optimization, input-convex neural networks, and an "end-to-end" flavor whereby the model is transformed to ensure some notion of reliability—in Chapter 7, uncertainty set calibration, and in Chapter 9, zero false negative rate. Both frameworks benefit from the model being able to focus its learning capacity on regions of the input-output space which are *most relevant*, enabling better performance while preserving the desired reliability guarantees. This commonality poses the interesting question, for future work, of whether the framework can be generalized to allow for the enforcement of more general notions of reliability during machine learning model training.

Part I

Algorithms with Black-Box AI/ML Advice

Chapter 2

CHASING CONVEX BODIES AND FUNCTIONS WITH BLACK-BOX ADVICE

We consider the problem of convex function chasing with black-box advice, where an online decision-maker aims to minimize the total cost of making and switching between decisions in a normed vector space, aided by black-box advice such as the decisions of a machine-learned algorithm. The decision-maker seeks cost comparable to the advice when it performs well, known as *consistency*, while also ensuring worst-case *robustness* even when the advice is adversarial. We first consider the common paradigm of algorithms that switch between the decisions of the advice and a competitive algorithm, showing that no algorithm in this class can improve upon 3-consistency while staying robust. We then propose three novel algorithms that bypass this limitation by exploiting the problem's convexity. The first, Adaptive Online Balanced Descent, obtains an optimal tradeoff of $(1+\epsilon)$-consistency and $O(\frac{1}{\epsilon})$-robustness in the one-dimensional setting. The second, Interp, achieves $(\sqrt{2}+\epsilon)$-consistency and $O(\frac{C}{\epsilon^2})$-robustness for any $\epsilon > 0$, where C is the competitive ratio of an algorithm for convex function chasing or a subclass thereof. The third, BdInterp, achieves $(1+\epsilon)$-consistency and $O(\frac{CD}{\epsilon})$-robustness when the problem has bounded diameter D.

This chapter is primarily based on the following paper:

[1] N. Christianson, T. Handina, and A. Wierman, "Chasing Convex Bodies and Functions with Black-Box Advice," in *Proceedings of the Thirty Fifth Conference on Learning Theory*, PMLR, Jun. 2022, pp. 867–908. [Online]. Available: https://proceedings.mlr.press/v178/christianson22a.html.

 In addition, the results on the one-dimensional setting presented in Section 2.4 are adapted from the paper

[1] D. Rutten, N. Christianson, D. Mukherjee, and A. Wierman, "Smoothed Online Optimization with Unreliable Predictions," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 7, no. 1, 12:1–12:36, Mar. 2023. DOI: 10.1145/3579442. [Online]. Available: https://dl.acm.org/doi/10.1145/3579442.

2.1 Introduction

We study the problem of convex function chasing (CFC), in which a player chooses decisions $\mathbf{x}_t$ online from a normed vector space $\mathcal{X} = (X, \|\cdot\|)$ in order to minimize the total cost $\sum_{t=1}^{T} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\|$, where each f_t is a convex "hitting" cost function that is revealed prior to the player's selection of $\mathbf{x}_t$, and the term $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|$ penalizes changing decisions between rounds. A number of subclasses of CFC have been discussed in the literature, characterized by various restrictions on the class of cost functions f_t. Of particular note is the special case of convex body chasing (CBC), in which each cost function f_t is the $\{0, \infty\}$ indicator of a convex set K_t, so that each decision $\mathbf{x}_t$ must reside strictly within K_t. Algorithms for CFC and its special cases are judged on the basis of their competitive ratio, i.e., the worst-case ratio in cost between the algorithm and the hindsight optimal sequence of decisions (Definition 2.2.1).

Convex body chasing and function chasing were introduced by [77] as continuous versions of several fundamental problems in online algorithms, including Metrical Task Systems [76] and the k-server problem [97]. CFC has also been studied recently as the problem of "smoothed online convex optimization" (SOCO), introduced by [38]. The basic premise of CFC/SOCO, of choosing decisions online to optimize per-round costs with minimal movement between decisions, has seen wide application in a number of domains, including datacenter load-balancing [38] and right-sizing [37, 93], electric vehicle charging [98], and control [99, 100].

In high-dimensional settings, the performance of algorithms for CBC and CFC can be arbitrarily poor: [77] showed a $\sqrt{d}$ lower bound on the competitive ratio of any algorithm for CBC (and thus CFC) in d-dimensional Euclidean space, which [101] extended to an $\Omega(\max\{\sqrt{d}, d^{1-\frac{1}{p}}\})$ lower bound in $\mathbb{R}^d$ with the ℓ^p norm. Prospects are poor even for subclasses of CFC with additional restrictions on the functions f_t. For instance, CFC with *α-polyhedral* cost functions, i.e., where each f_t has a unique minimizer away from which it grows with slope at least $\alpha > 0$, has been studied widely in the SOCO literature. State-of-the-art algorithms in this setting achieve competitive ratio $O(\alpha^{-1})$, which grows arbitrarily large in the $\alpha \to 0$ limit [102, 103].

The modern tools of machine learning wield great promise for improving upon these pessimistic performance guarantees. That is, for practical applications, there is often large amounts of data recorded from past problem instances, enabling the training of machine learning models that can outperform traditional, conservative online algorithms. However, these machine-learned algorithms are "black boxes," in the sense that they lack rigorous, worst-case performance guarantees. Such black-box algorithms might *typically* outperform robust online algorithms, but their lack of uncertainty quantification can lead to arbitrarily poor performance in the worse case, if they are deployed on held-out problem instances or under distribution shift.

Thus, a natural question arises: *is it possible to develop algorithms that achieve both the worst-case guarantees of traditional online algorithms for* CFC *and the average-case performance of machine-learned algorithms or other sources of black-box "advice"?*

These desiderata are naturally encoded in the notions of *robustness* and *consistency* introduced by [69] in the context of competitive caching. In this framework, a *consistent* algorithm is one with a competitive ratio with respect to the black-box advice, implying that when the advice is accurate, the algorithm will perform well; on the other hand, a *robust* algorithm is one that has a finite competitive ratio, regardless of advice performance. Our goal is to develop algorithms with tunable robustness and consistency guarantees, so that a decision-maker can decide in advance the tradeoff they wish to make between exploiting good advice performance and ensuring worst-case robustness in the case that advice performs poorly.

Contributions

We answer the question above by proposing novel algorithms with tunable robustness and consistency bounds for CFC and any subclass thereof. In particular, we reduce the general problem of designing robust and consistent algorithms for CFC to the design of *bicompetitive meta-algorithms* (Definitions 2.2.4, 2.2.5), which are unified "recipes" for combining black-box advice with a robust algorithm in a manner that guarantees a competitive ratio with respect to both ingredients. These "recipes" are very general—they can be used to combine advice with *any* algorithm for any subclass of CFC to obtain a customized robustness and consistency guarantee for that subclass without explicit knowledge of the algorithm or advice design.

More specifically, our contributions are twofold. We first consider the class of "switching" algorithms, which switch between the decisions of the advice and a robust algorithm. This class of algorithms has received considerable attention in the literature on robustness and consistency, and in particular, nearly all prior algorithms for CFC with black-box advice in dimension greater than one have been switching algorithms. We prove a fundamental limit on the robustness and consistency of any switching algorithm for CFC, showing that *no* switching algorithm for CFC can improve on 3-consistency while obtaining finite robustness (Theorem 2.3.3). We give a switching meta-algorithm Switch (Algorithm 1) achieving this fundamental limit, obtaining $(3 + O(\epsilon))$-consistency and $O(\frac{C}{\epsilon^2})$-robustness for any $\epsilon > 0$, where C is the competitive ratio of any algorithm for CFC or a subclass thereof. We further show that the fundamental limit on switching algorithms can be broken in the special case of nested CBC, in which successive bodies are nested. In this setting, we provide an algorithm NestedSwitch (Algorithm 2) achieving $(1+\epsilon)$-consistency along with $O(\frac{d}{\epsilon})$-robustness for nested CBC in d dimensions (Proposition 2.3.4).

Second, galvanized by the limitations of switching algorithms, we develop algorithms exploiting the convexity of the CFC problem to obtain improved robustness and consistency bounds. We first propose a (non-meta) algorithm for the one-dimensional setting, Adaptive Online Balanced Descent (Algorithm 3), that achieves $(1 + \epsilon)$-consistency and $O(\frac{1}{\epsilon})$-robustness when the decision space is $X = \mathbb{R}$, and we show that this is the asymptotically optimal tradeoff in this setting. We then propose a meta-algorithm Interp (Algorithm 4) that, for any normed vector space $\mathcal{X}$ of arbitrary dimension, given a C-competitive algorithm for a subclass of CFC, achieves $(\mu(\mathcal{X}) + \epsilon)$-consistency and $O(\frac{C}{\epsilon^2})$-robustness for any desired $\epsilon > 0$, where $\mu(\mathcal{X})$ is a geometric constant depending on the structure of the decision space that is $\sqrt{2}$ when $\mathcal{X}$ is a Hilbert space and that is strictly less than 3 in any ℓ^p space, $p \in (1, \infty)$ (Theorem 2.4.4). Moreover, under the additional assumption that the advice and the C-competitive algorithm are never farther apart than some distance D, we give a meta-algorithm BdInterp (Algorithm 5) that achieves $(1 + \epsilon)$-consistency and $O(\frac{CD}{\epsilon})$-robustness (Theorem 2.4.5). In particular, BdInterp gives nearly-optimal consistency and robustness for the problem of CFC with α-polyhedral cost functions when $D = O(1)$ in α.

A key feature of our results is their generality: our main results on bicompetitive meta-algorithms (Proposition 2.3.2, Theorems 2.4.4, 2.4.5) hold in vector spaces with any norm and arbitrary, even infinite, dimension. This enables application to problems where the decisions $\mathbf{x}_t$ are infinite-dimensional objects such as probability measures, which could arise in settings such as iterated games. Moreover, these meta-algorithms enable the design of customized robust and consistent algorithms for *any* subclass of CFC, since they are agnostic to the specific algorithms used. We illustrate this by giving specific robustness and consistency results for the cases of CFC, CBC, and CFC restricted to α-polyhedral hitting cost functions; we give further examples in Section 2.G.

Related work

Our work contributes to the literatures on CBC, CFC, and SOCO as well as the emerging literature on online algorithms with black-box advice. We discuss each in turn below.

CBC, CFC, and SOCO. The problems of convex body chasing and function chasing were introduced by Friedman and Linial [77], who gave a competitive algorithm for CBC in 2-dimensional Euclidean space. The problem in general dimension d has been largely settled in the last few years. In the setting where subsequent bodies are nested, Argue, Bubeck, et al. [104] gave an $O(d \log d)$-competitive algorithm in any norm, and Bubeck et al. [101] later gave an $O(\min\{d, \sqrt{d \log T}\})$-competitive algorithm in the Euclidean setting that uses the geometric Steiner point of the convex bodies. Later, Argue, Gupta, et al. [105] and Sellke [106] concurrently obtained $O(d)$-competitive algorithms for general CFC. The latter work builds upon the methods of Bubeck et al. [101], developing a "functional" Steiner point algorithm that is d-competitive for CBC and $(d+1)$-competitive for CFC in any normed space, matching the lower bound of d in the ℓ^∞ norm setting.

Several special cases of CFC/SOCO with restrictions on hitting cost structure have been studied in the literature to the end of obtaining "dimension-free" competitive ratios for these subclasses. Chen et al. [102] obtained the first such bound for the subclass of CFC where hitting cost functions f_t are α-polyhedral, which we call αCFC. The authors propose an algorithm, "Online Balanced Descent" (OBD), which achieves a competitive ratio $O(\frac{1}{\alpha})$. This upper bound has been successively refined, with the most recent entry a simple greedy algorithm from Zhang et al. [103] that achieves competitive ratio $\max\{1, \frac{2}{\alpha}\}$ in any normed vector space of arbitrary (even infinite) dimension by moving to the minimizer of each hitting cost function. The $O(\frac{1}{\alpha})$ upper bound has been broken by Lin [107] in the finite-dimensional Euclidean setting with an $O(\frac{1}{\alpha^{1/2}})$-competitive algorithm, Greedy OBD, that is

Problem	Algorithm Name	Comp. Ratio	Setting
CFC	Functional Steiner Point	$d+1$	$\mathbb{R}^d$ with any norm
CBC	Functional Steiner Point	d	$\mathbb{R}^d$ with any norm
αCFC	Greedy	$\max\left\{1, \frac{2}{\alpha}\right\}$	Any normed vector space
αCFC	Greedy OBD	$O\left(\frac{1}{\alpha^{1/2}}\right)$	$\mathbb{R}^d$ with ℓ^2 norm

Table 2.1: Competitive ratios for state-of-the-art algorithms for CFC, CBC, and αCFC.

optimal within the class of memoryless, rotation- and scale-invariant algorithms. Another subclass of CFC that has received attention is that with (κ, γ)-well-centered hitting cost functions, which generalize well-conditioned functions. Argue, Gupta, and Guruganesh [108] propose an algorithm achieving an $O(2^{\gamma/2}\kappa)$ competitive ratio for this subclass, and an improved algorithm achieving competitive ratio $O(\sqrt{\kappa})$ for the particular class of κ-well-conditioned functions along with a nearly matching $\Omega(\kappa^{1/3})$ lower bound. We summarize the state-of-the-art algorithms and competitive ratios that we refer to in our later results in Table 2.1, giving an extended version of the table in Section 2.A, Table 2.2.

Online Algorithms with Black-Box Advice. The idea of using machine-learned, black-box advice to improve online algorithms was first proposed by Mahdian et al. [109] to design algorithms for online ad allocation, load balancing, and facility location. Formal notions of *robustness* and *consistency* were later coined by Lykouris and Vassilvtiskii [69] in the context of designing learning-augmented algorithms for caching. The last few years have seen a surge in the application of the robustness and consistency paradigm in designing online algorithms augmented with black-box advice for a multitude of problems, for example ski rental and non-clairvoyant scheduling [70, 110], energy generation scheduling [71], bidding and bin-packing [111], and Q-learning [112].

Closest to our work are the recent papers of Antoniadis et al. [75] and Rutten, Christianson, et al. [113]. The former considers the problem of designing algorithms for *metrical task systems* (MTS) with black-box advice. MTS can be thought of as (non-convex) function chasing on general metric spaces, and hence their results also give robustness/consistency guarantees for CFC. They apply two classical results on combining k-server algorithms [114, 115] and on combining MTS algorithms via k-experts algorithms [116, 117] to devise, in our parlance, bicompetitive meta-algorithms for MTS. In particular, their first algorithm switches between the advice and a C-competitive algorithm for MTS, and achieves 9-consistency and $9C$-robustness. They also propose a randomized switching algorithm that, under the assumption that the metric space has bounded diameter D, obtains cost bounded in expectation by $\min\{(1+\epsilon)\mathrm{C}_{\mathrm{ADV}}+O(\frac{D}{\epsilon}),(1+\epsilon)C\cdot\mathrm{C}_{\mathrm{OPT}}+O(\frac{D}{\epsilon})\}$, where $\mathrm{C}_{\mathrm{ADV}}$ is the cost of the advice and $\mathrm{C}_{\mathrm{OPT}}$ is the optimal cost. However, the large $O(\frac{D}{\epsilon})$ additive factors in their result preclude $(1+\epsilon)$-consistency, since when $\mathrm{C}_{\mathrm{ADV}}=O(1)$, the consistency bound will be $1+\epsilon+\Omega(\frac{D}{\epsilon})$. This is to be expected, since as we show in Section 2.3, no deterministic switching algorithm can improve on 3-consistency while having finite robustness. Moreover, their results due not allow tuning robustness and consistency, i.e., neither algorithm allows trading-off robustness in order to obtain consistency arbitrarily close to 1.

On the other hand, [113] considers the problem of CFC with α-polyhedral hitting costs (αCFC), but with the convexity assumption dropped from the hitting costs f_t. They obtain a $(1+\epsilon)$-consistent, $2^{\tilde{O}(\frac{1}{\alpha\epsilon})}$-robust algorithm in this setting, together with a lower bound showing that this exponential tradeoff between robustness and consistency is necessary due to their non-convex setting. Their algorithm is a switching algorithm and crucially depends on the α-polyhedral structure of the hitting cost functions, and hence cannot be extended to general CFC. The authors also propose an algorithm for CFC in the 1-dimensional case (where $X=\mathbb{R}$) that achieves $(1+\epsilon)$-consistency and $O(\frac{1}{\epsilon^2})$ robustness, and they prove a lower bound of $(1+\epsilon)$-consistency and $\Omega(\frac{1}{\epsilon})$ robustness on *any* algorithm for CFC with black-box advice.[1] They leave open the broader problem of developing robust and consistent algorithms for CFC and its many subclasses in the higher-dimensional setting.

Notation

Throughout this chapter, X refers to a real vector space of arbitrary dimension. When a norm $\|\cdot\|$ is distinguished, $B(\mathbf{x},r)$ is the closed $\|\cdot\|$-ball of radius $r\geq 0$ centered at $\mathbf{x}$, and $\Pi_K\mathbf{x}$ is a metric projection of the point $\mathbf{x}\in X$ onto a closed convex set K. For $\mathbf{x},\mathbf{y}\in X$, we define $[\mathbf{x},\mathbf{y}]\coloneqq\{\mathbf{z}\in X:\mathbf{z}=\lambda\mathbf{x}+(1-\lambda)\mathbf{y},\lambda\in[0,1]\}$ as the convex span of $\mathbf{x}$ and $\mathbf{y}$. The non-negative reals are denoted by $\mathbb{R}_+$, and for $T\in\mathbb{N}$, we write $[T]\coloneqq\{1,\ldots,T\}$. Asymptotic notation involving the variable $\epsilon>0$ reflects the asymptotic regime $\epsilon\to 0$.

[1] The algorithm for the 1-dimensional setting proposed in [113] is the "Adaptive Online Balanced Descent" algorithm we describe in Algorithm 3. In Section 2.4, we provide a refined analysis to show that, in fact, this algorithm is $(1+\epsilon)$-consistent and $O(\frac{1}{\epsilon})$-robust, matching the lower bound.

2.2 Preliminaries

We consider the general problem of *convex function chasing* (CFC) on a real normed vector space $\mathcal{X} = (X, \|\cdot\|)$. In particular, we make no assumption on either the dimension of $\mathcal{X}$ or on the choice of norm $\|\cdot\|$. In CFC, a decision-maker begins at some initial point $\mathbf{x}_0 \in X$, and at each time $t \in \mathbb{N}$ is handed a convex function $f_t : X \to \mathbb{R}_+$ and must choose some $\mathbf{x}_t \in X$, paying both the *hitting cost* $f_t(\mathbf{x}_t)$, as well as the *movement* or *switching cost* $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|$ induced by the norm. Crucially, $\mathbf{x}_t$ is chosen prior to the revelation of any future cost functions f_k, $k > t$, i.e., decisions are made *online*. The game ends at some time $T \in \mathbb{N}$, which is unknown to the decision-maker in advance. We refer to a tuple $(\mathbf{x}_0, f_1, \dots, f_T)$ as an *instance* of the CFC problem. The total cost incurred by the decision-maker on a problem instance is $\sum_{t=1}^{T} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\|$.

Informally, an *online algorithm* for CFC is an algorithm that, on a given instance of CFC, produces decisions online. We denote by $\textsc{Alg}_t$ the t^{th} decision made by an online algorithm $\textsc{Alg}$; by convention, $\textsc{Alg}_0 \coloneqq \mathbf{x}_0$, the starting point of the instance. Then the cost $\mathrm{C}_{\textsc{Alg}}$ incurred by $\textsc{Alg}$ on an instance is

$$\mathrm{C}_{\textsc{Alg}} = \sum_{t=1}^{T} f_t(\textsc{Alg}_t) + \|\textsc{Alg}_t - \textsc{Alg}_{t-1}\|.$$

We also introduce the partial cost notation $\mathrm{C}_{\textsc{Alg}}(t, t') = \sum_{i=t}^{t'} f_i(\textsc{Alg}_i) + \|\textsc{Alg}_i - \textsc{Alg}_{i-1}\|$, defined for $1 \le t \le t' \le T$. We refer to the set of all online algorithms for CFC as $\mathcal{A}_{\mathsf{CFC}}$.

We typically compare online algorithms for CFC against $\textsc{Opt}$, the offline optimal algorithm that chooses the hindsight optimal sequence of decisions for any problem instance. Its cost is the optimal value of the following convex program:

$$\mathrm{C}_{\textsc{Opt}} = \mathrm{C}_{\textsc{Opt}}(\mathbf{x}_0, f_1, \dots, f_T) \coloneqq \min_{\mathbf{x}_1, \dots, \mathbf{x}_T \in X} \sum_{t=1}^{T} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\|$$

and its decisions are determined by the optimal solution. To evaluate the performance of an online algorithm for CFC, we consider the *competitive ratio*, which measures the worst case ratio in costs between an algorithm and $\textsc{Opt}$. In the following, we define both the conventional competitive ratio as well as a generalization that allows for comparing against arbitrary benchmark algorithms.

Definition 2.2.1. *Let* $\text{ALG}^{(1)}$ *be an online algorithm for* CFC*, and let* $\text{ALG}^{(2)}$ *be another (not necessarily online) algorithm for* CFC.[2] $\text{ALG}^{(1)}$ *is defined to be* ***C*-competitive with respect to** $\text{ALG}^{(2)}$ *if, regardless of problem instance,* $\mathrm{C}^{(1)}_{\text{ALG}} \leq C \cdot \mathrm{C}^{(2)}_{\text{ALG}}$*. In particular, if* $\text{ALG}^{(2)} = \text{OPT}$*, we simply say that* $\text{ALG}^{(1)}$ *is* ***C*-competitive***, or has* ***competitive ratio*** C.

Subclasses of Convex Function Chasing

CFC is a broad set of problems and many subclasses have received attention in the literature. We consider several subclasses of the general CFC problem in this work, distinguished by different assumptions on the hitting cost functions. In this section, we briefly define the subclasses of CFC which we refer to in our later results in the main text. We give more detailed definitions of these and several other subclasses of CFC in Section 2.A.

Convex Body Chasing. In the problem of convex body chasing (CBC), the decision-maker must choose each decision $\mathbf{x}_t$ from a convex body $K_t \subseteq X$ that is revealed online. This can be seen as a special case of CFC where f_t is 0 on K_t and ∞ elsewhere; see Section 2.A for more details on this equivalence. A notable special case of CBC is the problem of *nested* convex body chasing (NCBC), in which subsequent bodies are nested, i.e., $K_t \supseteq K_{t+1}$ for each t. We define $\mathcal{A}_{\mathsf{CBC}}$ as the set of all online algorithms for CBC that are *feasible*, i.e., that produce decisions within the convex body K_t at each time. We define $\mathcal{A}_{\mathsf{NCBC}}$ similarly as the set of *feasible* online algorithms for NCBC.

α-Polyhedral Convex Function Chasing. Several subclasses of CFC have been studied in the literature with hitting cost functions f_t restricted so as to enable dimension-free competitive ratios. One of the most well-studied such subclasses is the problem of α-polyhedral convex function chasing (αCFC), e.g., [102, 103], in which each hitting cost function f_t is restricted to be globally α-polyhedral, meaning intuitively that it has a unique minimizer, away from which it grows with slope at least $\alpha > 0$.

Definition 2.2.2. *Let* $(X, \|\cdot\|)$ *be a normed vector space, and let* $\alpha > 0$*. A function* $f : X \to \mathbb{R}_+$ *is* ***globally*** $\boldsymbol{\alpha}$***-polyhedral*** *if it has unique minimizer* $\mathbf{x}^* \in X$*, and in*

[2] Like OPT, the decision of $\text{ALG}^{(2)}$ at some time t is allowed to depend on problem instance data revealed after time t.

addition,

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) + \alpha\|\mathbf{x} - \mathbf{x}^*\| \qquad \textit{for all } \mathbf{x} \in X.$$

Using Black-Box Advice: Robustness, Consistency, and Bicompetitive Analysis

In this work, we seek algorithms for CFC and its subclasses that can exploit the good performance of a black-box advice algorithm, such as a reinforcement learning model, while maintaining rigorous worst-case performance guarantees. More specifically, we strive for algorithms that can obtain cost not much worse than optimal when the black-box advice is perfect, yet which have uniformly bounded competitive ratio when the advice is arbitrarily bad or even adversarial. This dual objective is naturally formulated in terms of *robustness* and *consistency*, which were introduced by [69] and are defined as follows.

Definition 2.2.3. *Let* ALG *be an online algorithm for* CFC, *and let* ADV *be a black-box advice algorithm.* ALG *is said to be* c***-consistent*** *if it is* c*-competitive with respect to* ADV. *On the other hand,* ALG *is defined to be* r***-robust*** *if it is* r*-competitive, independent of the performance of* ADV.

Our precise goal is to design algorithms achieving $(1 + \epsilon)$-consistency and $R(\epsilon)$-robustness for CFC and its subclasses, where $\epsilon > 0$ is a hyperparameter chosen by the decision-maker that encodes confidence in the advice. The dependence of the robustness $R(\epsilon)$ on ϵ anticipates a tradeoff between exploiting advice and worst-case robustness. We ideally seek algorithms with robustness $R(\epsilon)$ as small as possible, so that the tradeoff between consistency and robustness is tight.

Our methodology for designing robust and consistent algorithms is very general, in the sense that we do not restrict to any special cases of CFC and do not consider in our analysis the explicit behavior of the advice or of any specific algorithm for CFC. This is in contrast to the work of [113], whose main robustness and consistency guarantees depend crucially upon the α-polyhedral setting. Rather, we will primarily approach the task of designing robust and consistent algorithms via a more general problem of designing *bicompetitive meta-algorithms* for CFC, which, informally, are "recipes" for combining two CFC algorithms to produce a single algorithm with competitive guarantees with respect to both input algorithms. More formally, we give the following definitions.

Definition 2.2.4. *An online algorithm* $\textsc{Alg}$ *for* CFC *is* (c,r)***-bicompetitive*** *with respect to a pair of algorithms* $(\textsc{Alg}^{(1)}, \textsc{Alg}^{(2)})$ *if* $\textsc{Alg}$ *is simultaneously* c*-competitive with respect to* $\textsc{Alg}^{(1)}$ *and* r*-competitive with respect to* $\textsc{Alg}^{(2)}$*. Equivalently, the cost of* $\textsc{Alg}$ *can be bounded as*

$$\mathrm{C}_{\textsc{Alg}} \le \min\left\{c \cdot \mathrm{C}_{\textsc{Alg}^{(1)}}, r \cdot \mathrm{C}_{\textsc{Alg}^{(2)}}\right\}.$$

Definition 2.2.5. *A* ***meta-algorithm*** $\textsc{Meta}$ *for* CFC *is a mapping* $\textsc{Meta} : \mathcal{A}_{\mathsf{CFC}} \times \mathcal{A}_{\mathsf{CFC}} \to \mathcal{A}_{\mathsf{CFC}}$*. That is,* $\textsc{Meta}$ *takes as input two online algorithms for* CFC *and returns a single online algorithm for the problem.* $\textsc{Meta}$ *is said to be* (c,r)***-bicompetitive*** *if its output is always* (c,r)*-bicompetitive with respect to its inputs.*

It follows immediately from the previous two definitions that if $\textsc{Meta}$ is (c,r)-bicompetitive, $\textsc{Adv}$ is the advice, and $\textsc{Rob}$ is a b-competitive algorithm for (a subclass of) CFC, then $\textsc{Meta}(\textsc{Adv}, \textsc{Rob})$ is c-consistent and rb-robust. We discuss this observation in more detail in Section 2.A. Thus bicompetitive meta-algorithms give a general approach for designing robust and consistent algorithms for CFC and its subclasses.

The idea of approaching robust and consistent algorithm design via the design of bicompetitive meta-algorithms has been considered to some extent in the literature on other online problems, e.g., in the work of [75] on combining algorithms for MTS. To our knowledge, however, our specific terminology has not seen wide use in the literature.

2.3 Warmup: Switching Algorithms and Their Fundamental Limits

A natural first approach for designing bicompetitive meta-algorithms for CFC is to consider the class of switching algorithms, whose decisions switch between two other algorithms:

Definition 2.3.1. *A meta-algorithm* $\textsc{Meta}$ *is a* ***switching meta-algorithm*** *if, at each time* t*, the decision* $\textsc{Meta}_t(\textsc{Alg}^{(1)}, \textsc{Alg}^{(2)})$ *made by* $\textsc{Meta}$ *resides in the set* $\{\textsc{Alg}_t^{(1)}, \textsc{Alg}_t^{(2)}\}$.

Switching algorithms have garnered significant attention in the literature on robustness and consistency in recent years, e.g., [71, 75, 113, 118]. In particular, the only robust and consistent algorithms for CFC or subclasses thereof in general dimension are the switching algorithms of [75] for MTS and [113] for αCFC.

Algorithm 1: $\textsc{Switch}(\textsc{Adv}, \textsc{Rob}; b, \delta)$

Input: Algorithms $\textsc{Adv}, \textsc{Rob} \in \mathcal{A}_{\mathsf{CFC}}$; hyperparameters $b > 1$, $\delta \in (0, 1]$
Output: Decisions $\mathbf{x}_1, \ldots, \mathbf{x}_T$ chosen online

$i \leftarrow 0$
while problem instance has not ended **do**
  if $i \equiv 0 \mod 2$ **then**
    $\mathbf{x}_t \leftarrow \textsc{Adv}_t$ until the last time t that $\mathrm{C}_{\textsc{Adv}}(1, t) \leq b^i$
    $i \leftarrow i + 1$
  else
    $\mathbf{x}_t \leftarrow \textsc{Rob}_t$ until the last time t that $\mathrm{C}_{\textsc{Rob}}(1, t) \leq \delta b^i$
    $i \leftarrow i + 1$
end

In Algorithm 1, we generalize these prior algorithms to the general CFC setting, proposing a meta-algorithm $\textsc{Switch}$ which takes as hyperparameters $b > 1$ and $\delta \in (0, 1]$. In the following proposition, we show that $\textsc{Switch}$ obtains a tunable bicompetitive bound.

Proposition 2.3.2. *Suppose $\textsc{Adv}, \textsc{Rob}$ are algorithms for* CFC *and* $\mathrm{C}_{\textsc{Rob}} \geq 1$. *Then the switching meta-algorithm $\textsc{Switch}$ (Algorithm 1) is $\left(3 + O(\epsilon), 5 + O(\frac{1}{\epsilon^2})\right)$-bicompetitive with respect to the inputs $(\textsc{Adv}, \textsc{Rob})$, where $\epsilon > 0$ is an algorithm hyperparameter.*

Note that, in order to reduce the two hyperparameters b, δ used by $\textsc{Switch}$ to a single hyperparameter ϵ as in the statement of Proposition 2.3.2, we can simply introduce an auxiliary variable γ and make the substitutions $\delta \leftarrow b\gamma^2 - b^{-1}$, $b \leftarrow \sqrt{\gamma^{-2} + 1}$, and $\gamma \leftarrow \sqrt{\frac{\epsilon}{4}}$. Our proof of Proposition 2.3.2 follows closely that of [75, Theorem 1], extending it via the recent result of [118, Theorem 5] on linear search with a “hint”; we present a proof in Section 2.B.

If $\textsc{Adv}$ is an advice algorithm and $\textsc{Rob}$ is a C-competitive algorithm for (a subclass of) CFC, then $\textsc{Switch}$ yields $(3 + O(\epsilon))$-consistency and $(5 + O(\frac{1}{\epsilon^2}))C$-robustness. Notably, this does not appear to allow for *arbitrary* consistency: specifically, $\textsc{Switch}$ cannot attain consistency less than 3 while maintaining finite robustness. This limitation is unsurprising, since an identical lower bound holds on the algorithm for linear search on which $\textsc{Switch}$ is based [118, Theorem 7], which can be extended to a lower bound on the bicompetitiveness of switching meta-algorithms. It is natural, then, to ask whether this lower bound also applies to the robustness and consistency of CFC algorithms. That is, can we devise a switching algorithm that, so long as it is provided with some non-adversarial, competitive algorithm ROB, beats 3-consistency while staying robust?

In the following theorem, which we prove in Section 2.B, we show that robustness and consistency also face this fundamental limit: any algorithm that switches between black-box advice and an advice-agnostic competitive algorithm cannot beat 3-consistency while preserving robustness. We prove the theorem in the ℓ^2 setting, though the result extends to other norms such as the ℓ^∞ norm.

Theorem 2.3.3. *Consider the ℓ^2 norm setting. Let* ADV *be an advice algorithm, and let* ROB *be any (deterministic) competitive algorithm for* CBC *that is advice-agnostic. Let* ALG *be an online algorithm that deterministically switches between* ADV *and* ROB*. If* ALG *is c-consistent with $c < 3$, then* ALG *cannot have finite robustness.*

This lower bound implies that to obtain finite robustness alongside consistency $c < 3$ for general CFC, one must venture beyond the realm of switching algorithms. This is exactly the focus of Section 2.4, where we approach this task by exploiting the convexity of CFC. First, though, we ask: are there any special cases of CFC in which switching algorithms *can* obtain $(1 + \epsilon)$-consistency and finite robustness for any $\epsilon > 0$? The answer is affirmative for αCFC [113], and as we show in the next proposition, such an algorithm also exists for NCBC. Specifically, we propose an algorithm, NESTEDSWITCH (Algorithm 2), which can achieve a $(1 + \epsilon)$-consistent, $O(\frac{d}{\epsilon})$-robust tradeoff for NCBC by using a simple threshold-based rule for switching between the advice and the Steiner point algorithm of [101]. We prove Proposition 2.3.4 in Section 2.B.

Proposition 2.3.4. *Consider the problem of* NCBC *on $(\mathbb{R}^d, \|\cdot\|_{\ell^2})$, where the initial body K_1 resides in some ball $B(\mathbf{y}, r)$ of radius r containing $\mathbf{x}_0$, and $\mathrm{C}_{\mathrm{OPT}} \geq 1$. If* ROB *is the Steiner point algorithm ([101]) that chooses the Steiner point of K_t at each time t, then* NESTEDSWITCH *(Algorithm 2) is $(1+\epsilon)$-consistent and $\left(1 + \frac{1}{\epsilon}\right) r(d+2)$-robust, where $\epsilon > 0$ is a hyperparameter.*

2.4 Beyond Switching Algorithms: Exploiting Convexity to Break 3-Consistency

In this section, we present our main results: three novel algorithms that transcend the limitations of switching algorithms by exploiting the convexity of CFC. The key insight that enables this improved performance is that *hedging* between ADV and some more robust decision—i.e., choosing a decision that is a convex combination

Algorithm 2: $\textsc{NestedSwitch}(\textsc{Adv}, \textsc{Rob}; \epsilon, r)$

Input: Algorithms $\textsc{Adv}, \textsc{Rob} \in \mathcal{A}_{\mathsf{NCBC}}$; hyperparameters $\epsilon, r > 0$
Output: Decisions $\mathbf{x}_1 \in K_1, \dots, \mathbf{x}_T \in K_T$ chosen online

```
for t = 1, 2, ..., T do
    Observe K_t, x̃_t := Adv_t, and s_t := Rob_t
    if ε · C_Adv(1, t) ≥ r(d + 2) then
        x_t ← s_t
    else
        x_t ← x̃_t
end
```

of $\textsc{Adv}$ and some robust strategy $\textsc{Rob}$, or that approaches but does not reach $\textsc{Adv}$ unless it is sufficiently good—allows for more nuanced algorithmic behavior than switching permits.

AOBD: An Optimal Memoryless Algorithm in One Dimension

We begin this section by considering the restricted setting where the vector space is $X = \mathbb{R}$; the results in this subsection are adapted from Rutten, Christianson, et al. [113]. We may assume without loss of generality that the norm is the absolute value $|\cdot|$, since all norms on $\mathbb{R}$ are a scalar multiple of this one. Note that despite its simplicity, the one-dimensional setting has seen significant study in the literature on CFC/SOCO as well as their application to, e.g., sustainable datacenter operation [37, 119, 120]. We will also make the temporary assumption that each f_t has a finite minimizer, though this assumption can be lifted (see the discussion following Theorem 2.4.1).

We propose in Algorithm 3 a strategy which we call *Adaptive Online Balanced Descent (AOBD)*, that builds on the classic Online Balanced Descent framework [102] to incorporate black-box advice. Note that this algorithm is not a meta-algorithm, since it only requires the input of a single advice algorithm; rather, it acts to directly "robustify" the decisions made by the provided advice, without incorporating the decisions of some additional robust strategy. Intuitively, AOBD seeks to follow the advice as closely as possible while approximately balancing the incurred hitting cost with the movement cost. The extent to which the hitting cost and movement cost can differ is governed by two confidence hyperparameters $\overline{\beta} > 1 > \underline{\beta} > 0$; intuitively, if the ratio $\overline{\beta}/\underline{\beta}$ is larger, the algorithm will have more leeway to follow the advice closely, even if this leads to an imbalance between movement and hitting cost (this corresponds to an increase in the width of the

Algorithm 3: Adaptive Online Balanced Descent AOBD($\textsc{Adv}$; $\underline{\beta}, \overline{\beta}$)

Input: Algorithm $\textsc{Adv}$; hyperparameters $\overline{\beta} > 1 > \underline{\beta} > 0$
Output: Decisions $x_1 \in \mathbb{R}, \ldots, x_T \in \mathbb{R}$ chosen online

1 **for** $t = 1, 2, \ldots, T$ **do**
2   Observe f_t and $\tilde{x}_t := \textsc{Adv}_t$
3   Choose some $v_t \in \arg\min_{v \in \mathbb{R}} f_t(v)$
4   Define $x(\lambda) := (1 - \lambda)x_{t-1} + \lambda v_t$ for $\lambda \in [0, 1]$
5   Choose $\underline{\lambda} \in [0, 1]$ such that $|x(\underline{\lambda}) - x_{t-1}| = \underline{\beta} f_t(x(\underline{\lambda}))$, or $\underline{\lambda} = 1$ if no such $\underline{\lambda}$ exists
6   Choose $\overline{\lambda} \in [0, 1]$ such that $|x(\overline{\lambda}) - x_{t-1}| = \overline{\beta} f_t(x(\overline{\lambda}))$, or $\overline{\lambda} = 1$ if no such $\overline{\lambda}$ exists
7     $\hat{\lambda} \leftarrow \arg\min_{\lambda \in [\underline{\lambda}, \overline{\lambda}]} |x(\lambda) - \tilde{x}_t|$
8     $x_t \leftarrow x(\hat{\lambda})$
9 **end**

interval $[\underline{\lambda}, \overline{\lambda}]$ employed in line 7 of the algorithm). Determining the values of $\underline{\lambda}$ and $\overline{\lambda}$ can be done efficiently using binary search.

In the following theorem, we characterize the robustness and consistency of AOBD.

Theorem 2.4.1. *The algorithm AOBD (Algorithm 3) is* $\max\left\{\frac{1+\underline{\beta}}{1-\underline{\beta}}, \frac{1+\overline{\beta}}{\overline{\beta}}\right\}$*-consistent and* $\max\left\{1 + 2\overline{\beta}, 1 + \underline{\beta}^{-1}\right\}$*-robust. In particular, if* $\underline{\beta} = \frac{\epsilon}{2+\epsilon}$ *and* $\overline{\beta} = \frac{1}{\epsilon}$ *for some* $\epsilon \in (0, 1)$*, then AOBD is* $(1 + \epsilon)$*-consistent and* $\left(2 + \frac{2}{\epsilon}\right)$*-robust.*

A proof of this theorem is presented in Section 2.C. Before proceeding, we make two brief remarks. First, note that our analysis is a refined version of the original analysis in the paper [113], which enables us to obtain an improved robustness-consistency tradeoff. Second, note that the assumption we made that each f_t has a finite minimizer can be easily relaxed by redefining $x(\lambda)$ in any case that $v_t \in \{\pm\infty\}$ so that the algorithm "moves toward" v_t until the relevant balance condition holds.

A natural question is whether the tradeoff between robustness and consistency incurred by AOBD is necessary, or whether any better tradeoff can be obtained. As we show in the following theorem, AOBD achieves a tradeoff between robustness and consistency that is *asymptotically optimal* in ϵ.

Theorem 2.4.2. *Let* $\epsilon \in (0, \frac{1}{2})$*, and let* $\textsc{Alg}$ *be any algorithm for* CFC *on* $\mathbb{R}$ *with advice* $\textsc{Adv}$*. If* $\textsc{Alg}$ *is* $(1 + \epsilon)$*-consistent, then it must be at least* $\frac{1}{2\epsilon}$*-robust.*

A proof of this theorem is presented in Section 2.C. This result, together with the upper bound obtained by AOBD, implies that the optimal tradeoff between consistency and robustness for CFC on $\mathbb{R}$ is $(1+\epsilon)$-consistency and $\Theta(\frac{1}{\epsilon})$-robustness. Having established such an optimal tradeoff in the one-dimensional setting, one might wonder whether a similar algorithmic strategy might enable near-optimal tradeoffs between consistency and robustness in more general settings—i.e., whether it is possible to incorporate black-box advice into the Online Balanced Descent framework [102] in higher-dimensional vector spaces. Unfortunately, it is not likely that such a straightforward extension of AOBD can obtain good robustness-consistency tradeoffs: [113, Theorem 15] shows that any *memoryless* and *symmetry-invariant* algorithm (which includes natural generalizations of the Online Balanced Descent framework) cannot in general obtain both robustness and nontrivial consistency. As such, more sophisticated algorithms that effectively utilize memory are needed to take advantage of black-box advice while preserving robustness in high-dimensional settings.

Interp: a $(\sqrt{2}+\epsilon, O(\epsilon^{-2}))$-Bicompetitive Meta-Algorithm

We now return to the setting of general dimension and arbitrary convex functions f_t, and the task of designing bicompetitive meta-algorithms. We preface our algorithmic discussion with some definitions from the geometry of real normed vector spaces $\mathcal{X} = (X, \|\cdot\|)$ which we employ in the proposed algorithm's statement and performance bound. We present abridged introductions of these notions here, giving more detail in Section 2.D.

We begin by introducing the *rectangular constant* $\mu(\mathcal{X})$ of a normed vector space $\mathcal{X}$, which is bounded between $\sqrt{2}$ and 3, with $\mu(\mathcal{X}) = \sqrt{2}$ when $\mathcal{X}$ is Hilbert and $\mu(\ell^p) < 3$ for any $p \in (1, \infty)$ ([121, 122]). Next, we define the *radial retraction*.

Definition 2.4.3 ([123])**.** *On a normed vector space $\mathcal{X} = (X, \|\cdot\|)$, the **radial retraction** $\rho(\cdot\,; r) : X \to B(\mathbf{0}, r)$ is the metric projection onto the closed ball of radius $r \geq 0$:*

$$\rho(\mathbf{x}; r) = \begin{cases} \mathbf{x} & \text{if } \|\mathbf{x}\| \leq r \\ r\frac{\mathbf{x}}{\|\mathbf{x}\|} & \text{if } \|\mathbf{x}\| > r. \end{cases}$$

On a fixed normed space $\mathcal{X}$, the collection of radial retractions $\rho(\cdot\,; r)$ with $r > 0$ share a Lipschitz constant, which we call $k(\mathcal{X})$. It is known that $1 \leq k(\mathcal{X}) \leq 2$ ([124]), and moreover $k(\mathcal{X}) \leq \mu(\mathcal{X})$ (Section 2.D, Proposition 2.D.6).

Algorithm 4: $\textsc{Interp}(\textsc{Adv}, \textsc{Rob}; \epsilon, \gamma, \delta)$

Input: Algorithms $\textsc{Adv}, \textsc{Rob}$; hyperparameters $\epsilon > 0$ and $\gamma > 0, \delta > 0$ satisfying $2\gamma + 2\delta = \epsilon$

Output: Decisions $\mathbf{x}_1, \dots, \mathbf{x}_T$ chosen online

1 **for** $t = 1, 2, \dots, T$ **do**
2   Observe f_t, $\tilde{\mathbf{x}}_t := \textsc{Adv}_t$, and $\mathbf{s}_t := \textsc{Rob}_t$
3   **if** $\mathrm{C}_{\textsc{Rob}}(1,t) \geq \delta \cdot \mathrm{C}_{\textsc{Adv}}(1,t)$ **then**
4         $\mathbf{x}_t \leftarrow \tilde{\mathbf{x}}_t$
5   **else**
6         $\mathbf{y}_t \leftarrow \mathbf{s}_{t-1} + \rho\left(\tilde{\mathbf{x}}_t - \mathbf{s}_{t-1}; \|\mathbf{x}_{t-1} - \mathbf{s}_{t-1}\|\right)$
7         $\mathbf{z}_t \leftarrow \mathbf{s}_{t-1} + \rho\left(\mathbf{y}_t - \mathbf{s}_{t-1}; \max\{\|\mathbf{y}_t - \mathbf{s}_{t-1}\| - \gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t), 0\}\right)$
8         $\mathbf{x}_t \leftarrow \mathbf{s}_t + \rho(\tilde{\mathbf{x}}_t - \mathbf{s}_t; \|\mathbf{z}_t - \mathbf{s}_{t-1}\|)$
9 **end**

With these definitions at our disposal, we now proceed to the main result of this section. We propose Algorithm 4, a meta-algorithm Interp that takes as input two algorithms Adv, Rob for (a subclass of) CFC, and hyperparameters $\epsilon, \gamma, \delta > 0$ satisfying $2\gamma + 2\delta = \epsilon$. Interp works as follows: at each time t, if the cost of Rob so far is a substantial fraction of the cost of Adv, then Interp can move to $\textsc{Adv}_t$ while staying competitive with respect to Rob, and it does so (line 4). Otherwise, Interp moves to a point $\mathbf{x}_t$ determined by the series of radial projections (lines 6, 7, and 8), which intuitively guide Interp to take a "greedy step" toward the decision made by Rob while still remaining close enough to Adv so as to maintain a consistency guarantee.

We characterize the bicompetitive performance of Interp in the following theorem, which holds in any normed vector space $\mathcal{X}$ of arbitrary dimension.

Theorem 2.4.4. *Interp (Algorithm 4) is*

$$\left(\mu(\mathcal{X}) + \epsilon, 1 + \frac{k(\mathcal{X})}{\gamma} + \frac{\mu(\mathcal{X}) + \epsilon + 1 + \frac{k(\mathcal{X})}{\gamma}}{\delta}\right)\text{–bicompetitive}$$

with respect to $(\textsc{Adv}, \textsc{Rob})$. *With* γ, δ *chosen optimally, the bound is* $(\mu(\mathcal{X}) + \epsilon, O(\epsilon^{-2}))$.

In particular, if Adv is advice and Rob is C-competitive for (a subclass of) CFC*, then Interp is $(\mu(\mathcal{X}) + \epsilon)$-consistent and $O(C\epsilon^{-2})$-robust.*

Notably, Interp (Algorithm 4) strictly improves on the 3-consistent lower bound for switching meta-algorithms in any ℓ^p space with $1 < p < \infty$, in which it holds that

$\mu(\ell^p) < 3$. Moreover, it obtains consistency $(\sqrt{2}+\epsilon)$ in any Hilbert space. We prove Theorem 2.4.4 and give details regarding optimal selection of the parameters γ, δ in Section 2.E. The proof employs two potential function arguments with different potential functions for the bounds with respect to Adv and Rob. Moreover, the generality of the theorem's setting requires the development of several geometric results characterizing the radial projection and its relation to the rectangular constant in arbitrary-dimensional normed vector spaces, which we present in Section 2.E prior to the main proof. These results are crucial for enabling Interp's robustness and consistency in the general setting and elucidate the presence of the constants $\mu(\mathcal{X})$ and $k(\mathcal{X})$ in its bicompetitive bound.

We also detail robustness and consistency corollaries of Theorem 2.4.4 for multiple subclasses of CFC in Section 2.G. In particular, Theorem 2.4.4 and Table 2.1 imply an algorithm for CFC and CBC on $\mathbb{R}^d$ with any norm that is $(\mu(\mathbb{R}^d, \|\cdot\|) + \epsilon)$-consistent and $O(\frac{d}{\epsilon^2})$-robust; we also obtain an algorithm for αCFC that is $(\mu(\mathcal{X}) + \epsilon)$-consistent and $O(\frac{1}{\alpha\epsilon^2})$-robust for αCFC on any normed vector space $\mathcal{X}$.

Attaining $(1+\epsilon)$-Consistency in Bounded Instances with BdInterp

In the preceding section, we proved that in the Hilbert space setting, Interp (Algorithm 4) obtains consistency $(\sqrt{2}+\epsilon)$ while remaining competitive with respect to Rob. While this is a significant improvement on the limit of 3-consistency faced by switching algorithms, the question remains: can we devise an algorithm that achieves $(1+\epsilon)$-consistency and $R(\epsilon) < \infty$ competitiveness with respect to Rob for *any* $\epsilon > 0$, in *any* normed vector space? In this section, we provide a simple sufficient condition under which this is possible: if there exists some constant $D \in \mathbb{R}_+$ for which $\|\text{Adv}_t - \text{Rob}_t\| \leq D$ for all t, then there is a meta-algorithm that is $(1+\epsilon, O(\frac{D}{\epsilon}))$-bicompetitive with respect to (Adv, Rob). We call this condition *D-boundedness* of Adv and Rob; it arises naturally in a number of settings, for example in any CBC instance in which the diameter of each body K_t is bounded by D.

We present the algorithm achieving this bicompetitive bound, BdInterp, in Algorithm 5. Just like Interp, BdInterp takes as input two algorithms Adv, Rob for (a subclass of) CFC, and hyperparameters $\epsilon, \gamma, \delta > 0$ satisfying $2\gamma + 2\delta = \epsilon$. At a high level, BdInterp operates similarly to Interp, though it takes smaller greedy steps toward Rob, enabling it to maintain $(1+\epsilon)$-consistency. Specifically, BdInterp works as follows: if the cost of Rob is a sufficient fraction of the cost of Adv, then

Algorithm 5: BdInterp(Adv, Rob; ϵ, γ, δ)

Input: Algorithms Adv, Rob; hyperparameters $\epsilon > 0$ and $\gamma > 0, \delta > 0$ satisfying $2\gamma + 2\delta = \epsilon$

Output: Decisions $\mathbf{x}_1, \ldots, \mathbf{x}_T$ chosen online

```
1 for t = 1, 2, ..., T do
2     Observe f_t, x̃_t := Adv_t, and s_t := Rob_t
3     if C_Rob(1, t) ≥ δ · C_Adv(1, t) then
4         x_t ← x̃_t
5     else
6         ν ← ||x_{t-1} − s_{t-1}|| / ||x̃_{t-1} − s_{t-1}|| if x̃_{t-1} ≠ s_{t-1}, otherwise ν ← 0
7         y_t ← ν x̃_t + (1 − ν) s_t
8         x_t ← s_t + ρ (y_t − s_t; max{||y_t − s_t|| − γ · C_Adv(t, t), 0})
9 end
```

BdInterp moves to $\textsc{Adv}_t$ (line 4). Otherwise, it selects an auxiliary point $\mathbf{y}_t$ as the point along the segment $[\mathbf{s}_t, \tilde{\mathbf{x}}_t]$ with the same relative position as $\mathbf{x}_{t-1}$ on the segment $[\mathbf{s}_{t-1}, \tilde{\mathbf{x}}_{t-1}]$ (line 7), and then chooses $\mathbf{x}_t$ by taking a greedy step toward $\mathbf{s}_t$ from $\mathbf{y}_t$ (line 8).

We present the performance result for BdInterp in Theorem 2.4.5; like Theorem 2.4.4, the result holds in any normed vector space $\mathcal{X}$ of arbitrary dimension.

Theorem 2.4.5. *Suppose that Adv and Rob are D-bounded, i.e., $\|\textsc{Adv}_t - \textsc{Rob}_t\| \leq D$ for all $t \in [T]$; and assume that $\mathrm{C}_{\textsc{Rob}} \geq 1$. Then BdInterp (Algorithm 5) is*

$$\left(1 + \epsilon, D + \frac{D}{\gamma} + \frac{1 + \epsilon}{\delta}\right)\text{–bicompetitive}$$

with respect to (Adv, Rob). With γ, δ chosen optimally, the bound is $(1 + \epsilon, O(\frac{D}{\epsilon}))$.

In particular, if Adv is advice and Rob is C-competitive for (a subclass of) CFC*, then BdInterp is $(1 + \epsilon)$-consistent and $O(\frac{CD}{\epsilon})$-robust.*

Remarkably, Theorem 2.4.5 states that BdInterp not only improves on the consistency of Interp, but it also strictly improves upon Interp's $O(\epsilon^{-2})$ competitiveness with respect to Rob when $D = o(\epsilon^{-1})$. Moreover, BdInterp substantially improves on the randomized switching algorithm of [75] in the D-bounded setting, providing a tunable robustness and consistency guarantee with no additive factor in the consistency term. We give a proof of Theorem 2.4.5, as well as details on optimal parameter selection, in Section 2.F. The argument follows a similar line of reasoning as that of our proof of Theorem 2.4.4, though in the proof of competitiveness with respect to ROB (i.e., robustness), we employ a novel potential function constructed via the ratio between the respective distances of $\mathbf{x}_t$ and $\tilde{\mathbf{x}}_t$ to $\mathbf{s}_t$.

We detail robustness and consistency corollaries of Theorem 2.4.5 for multiple subclasses of CFC in Section 2.G. In particular, Theorem 2.4.5 and Table 2.1 imply an algorithm for CFC and CBC with any norm that is $(1+\epsilon)$-consistent and $O(\frac{dD}{\epsilon})$-robust on D-bounded instances. We also obtain an algorithm for αCFC in the D-bounded finite-dimensional Euclidean setting that achieves $(1+\epsilon)$-consistency and $O(\frac{D}{\alpha^{1/2}\epsilon})$-robustness. This latter bound is nearly tight for αCFC: Theorem 2.4.2 gives a lower bound of $\Omega(\frac{1}{\epsilon})$ for any $(1+\epsilon)$-consistent αCFC algorithm when $\alpha = 2\epsilon$. When $D = O(1)$, Theorem 2.4.5 gives us $(1+\epsilon)$-consistency and $O(\frac{1}{\alpha^{1/2}\epsilon})$-robustness, leaving a gap of just $O(\alpha^{-1/2})$ between the upper and lower bounds. We leave to future work the question of whether these upper and lower bounds can be made tight and whether the factor of D (and more generally the D-boundedness assumption) can be dropped; this latter question will be studied in Chapter 3 of this thesis.

2.5 Conclusion

In this chapter, we examine the question of integrating black-box advice into algorithms for convex function chasing using the notions of robustness and consistency from the literature on online algorithms with machine-learned advice. We first propose an algorithm that switches between the decisions of an arbitrary C-competitive algorithm ROB and the advice, showing that it obtains $(3+O(\epsilon))$-consistency and finite robustness for any $\epsilon > 0$. We moreover show that this is optimal, in the sense that *no* switching algorithm can improve upon 3-consistency while maintaining finite robustness. We then move beyond switching algorithms, and propose three algorithms, AOBD, INTERP, and BDINTERP, which obtain improved robustness and consistency guarantees by exploiting the convexity inherent in the CFC problem. In particular, AOBD obtains the optimal tradeoff of $(1+\epsilon)$-consistency and $O(\frac{1}{\epsilon})$-robustness in the one-dimensional setting, INTERP obtains $(\sqrt{2}+\epsilon)$-consistency and $O(\frac{C}{\epsilon^2})$-robustness in the general Hilbert space setting, and under the additional assumption of D-boundedness, BDINTERP can obtain $(1+\epsilon)$-consistency and $O(\frac{CD}{\epsilon})$-robustness.

Several interesting questions remain open for future work: in particular, (a) the question of whether $(1+\epsilon)$-consistency and finite robustness can be obtained for general CFC without the D-boundedness assumption, and (b) the question of tight

Problem	State-of-the-Art Competitive Ratio	Setting
CFC	$d + 1$	$\mathbb{R}^d$ with any norm
CBC	d	$\mathbb{R}^d$ with any norm
kCBC	$2k + 1$	$\mathbb{R}^d$ with ℓ^2 norm
αCFC	$\max\left\{1, \frac{2}{\alpha}\right\}$	Any normed vector space
αCFC	$O\left(\frac{1}{\alpha^{1/2}}\right)$	$\mathbb{R}^d$ with ℓ^2 norm
(κ, γ)CFC	$(2 + 2\sqrt{2})2^{\gamma/2}\kappa$	$\mathbb{R}^d$ with ℓ^2 norm

Table 2.2: Competitive ratios for state-of-the-art algorithms on various subclasses of CFC.

lower bounds on robustness and consistency for CFC and its many subclasses. Specifically, for the case of CFC in general, we pose the question of whether $(1 + \epsilon)$-consistency is possible together with $O(\frac{d}{\epsilon})$-robustness, or even whether the dependence on ϵ and d can be further improved in the robustness bound. We will provide more insight on these questions in the subsequent chapter of this thesis.

Appendix

In these appendix sections, we provide additional background on CFC and its special cases, proofs of the theoretical results in the main body of the chapter, and geometric details that are leveraged in our design and analysis of Interp (Algorithm 2.4.4).

2.A Preliminaries

In this section, we provide more detailed definitions of the subclasses of convex function chasing considered in this work, including both those introduced in Section 2.2 as well as several additional special cases which we refer to in the robustness and consistency results given in Section 2.G. We also review state-of-the-art competitive algorithms for each subclass, which we summarize in Table 2.2, which is an extended version of Table 2.1 in the main text. We then elaborate on the claim made in Section 2.2 that a (c, r)-bicompetitive meta-algorithm for CFC, along with a b-competitive algorithm for a subclass of CFC, together yield a c-consistent and rb-robust algorithm for that subclass.

Subclasses of CFC

Convex body chasing. In the problem of convex body chasing (CBC) on a normed vector space $(X, \|\cdot\|)$, at each time t a decision-maker is given a convex body $K_t \subseteq X$ and faces the requirement that their decision $\mathbf{x}_t$ must reside within K_t. A further special case of CBC is the problem of *nested* convex body chasing (NCBC), in which subsequent bodies are nested, i.e., $K_t \supseteq K_{t+1}$ for each t. We define the set of all online algorithms which are *feasible* for CBC, i.e., which produce decisions residing within the convex body K_t at each time, as $\mathcal{A}_{\mathsf{CBC}}$. We define $\mathcal{A}_{\mathsf{NCBC}}$ similarly as the set of all online algorithms which are feasible for NCBC. [106] proved that an algorithm based on a functional generalization of the Steiner point of a convex body achieves competitive ratio d for CBC and $d+1$ for general CFC in $\mathbb{R}^d$ equipped with any norm.

The problem of convex body chasing can easily be seen as a special case of CFC in which each hitting cost f_t is the $\{0, \infty\}$ indicator of the convex set K_t. That is,

$$f_t(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in K_t \\ \infty & \text{otherwise.} \end{cases}$$

As noted in [106], we need not even require hitting costs to take infinite values to recover convex body chasing from function chasing. Indeed, restricting to the finite-dimensional setting,[3] consider f_t defined as

$$f_t(\mathbf{x}) = 3 \cdot d(\mathbf{x}, K_t) = 3 \min_{\mathbf{y} \in K_t} \|\mathbf{x} - \mathbf{y}\|.$$

Then any algorithm $\textsc{Alg} \in \mathcal{A}_{\mathsf{CFC}}$ yields a set of decisions $\textsc{Alg}_1, \dots, \textsc{Alg}_T$ on the instance $(\mathbf{x}_0, f_1, \dots, f_T)$; and moreover, $\textsc{Alg}$ can be transformed into an algorithm $\textsc{Alg}' \in \mathcal{A}_{\mathsf{CFC}}$ with strictly improved cost, and which in particular incurs no hitting cost, by setting

$$\textsc{Alg}'_t = \begin{cases} \textsc{Alg}_t & \text{if } \textsc{Alg}_t \in K_t \\ \Pi_{K_t} \textsc{Alg}_t & \text{otherwise.} \end{cases}$$

Clearly each decision $\textsc{Alg}'_t$ resides in the convex body K_t; thus $\textsc{Alg}'$ is a feasible online algorithm for CBC, i.e., $\textsc{Alg}' \in \mathcal{A}_{\mathsf{CBC}}$. Moreover, the cost of $\textsc{Alg}'$ on the CBC instance is identical to its cost for the corresponding CFC instance, since it incurs no hitting cost. It follows that the competitive ratio of $\textsc{Alg}'$ for the CBC problem is at most the competitive ratio of $\textsc{Alg}$ as a CFC algorithm, since $\textsc{Opt}$ for a CBC instance and its corresponding CFC instance always coincide. In short, a C-competitive algorithm for CFC is also C-competitive for CBC.

[3] This restriction is natural because no algorithm can be competitive for CBC in the infinite-dimensional setting. We impose the restriction in order to ensure existence of a metric projection onto K_t.

Remark 2.A.1. *The preceding line of reasoning can be extended to show that a C-competitive algorithm for* CFC *gives a C-competitive algorithm for* CFC *with strict decision constraints, i.e., where at time t, the decision $\mathbf{x}_t$ must both reside in some convex body K_t and also incurs a convex hitting cost $f_t(\mathbf{x}_t)$.*

Chasing low-dimensional convex bodies. A special case of convex body chasing that has received significant attention is the problem of chasing low-dimensional bodies in higher-dimensional space: indeed, the seminal work of [77] began by addressing the problem of chasing lines in the Euclidean plane. [125] later presented a 3-competitive algorithm for chasing lines in $(\mathbb{R}^d, \|\cdot\|_{\ell^2})$, and most recently [108] gave an algorithm that is $(2k+1)$-competitive for chasing convex bodies lying in k-dimensional affine subspaces, regardless of the dimension d of the underlying Euclidean space. Motivated by this last result, we define the problem of k-dimensional convex body chasing (kCBC), comprised of all instances of CBC in which each body K_t lies within an affine subspace of dimension at most k—i.e., $\dim \operatorname{aff} K_t \leq k$ for all t.

α-polyhedral convex function chasing. A class of functions that has been studied extensively in the literature on online optimization with switching costs (SOCO) is the class of globally α-polyhedral functions, e.g., [102, 103], which are defined as follows.

Definition 2.A.2. *Let $(X, \|\cdot\|)$ be a normed vector space, and let $\alpha > 0$. A function $f : X \to \mathbb{R}_+$ is **globally α-polyhedral** if it has unique minimizer $\mathbf{x}^* \in X$, and in addition,*

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) + \alpha\|\mathbf{x} - \mathbf{x}^*\| \qquad \textit{for all } \mathbf{x} \in X.$$

Roughly speaking, a globally α-polyhedral function has a unique minimizer, away from which it grows with slope at least α. For a fixed $\alpha > 0$, we define the problem of *α-polyhedral convex function chasing* (αCFC) comprised of all those problem instances of CFC in which, in addition to being convex, each function f_t is also globally α-polyhedral. αCFC has been widely studied due to its admitting algorithms with "dimension-free" competitive ratios: [103] showed that a greedy algorithm that simply moves to the minimizer $\mathbf{x}_t^*$ of each function f_t is $\max\{1, \frac{2}{\alpha}\}$-competitive for αCFC in any normed vector space. In the setting of $\mathbb{R}^d$ with the ℓ^2 norm, [107] gave an algorithm augmenting the Online Balanced Descent algorithm of [102] to achieve a competitive ratio of $O(\frac{1}{\alpha^{1/2}})$.

(κ, γ)-well-centered convex function chasing. Another class of functions that has received attention in the design of algorithms for subclasses of CFC with dimension-free competitive ratios is the set of (κ, γ)-well-centered functions, introduced by [108]:

Definition 2.A.3. *Let $(X, \|\cdot\|)$ be a normed vector space, and let $\kappa, \gamma \geq 1$. A function $f : \mathbb{R}^d \to \mathbb{R}_+$ with minimizer $\mathbf{x}^*$ is (κ, γ)-**well-centered** if there exists some $a > 0$ such that*

$$\frac{a}{2}\|\mathbf{x} - \mathbf{x}^*\|^\gamma \leq f(\mathbf{x}) \leq \frac{a\kappa}{2}\|\mathbf{x} - \mathbf{x}^*\|^\gamma \qquad \textit{for all } \mathbf{x} \in X.$$

Intuitively, the growth rate of a (κ, γ)-well-centered function away from its minimizer (as measured with the "distance" $\|\cdot\|^\gamma$) is bounded above and below, and the ratio of these bounds is at most κ. For fixed $\kappa, \gamma \geq 1$, we define the problem of (κ, γ)-*well-centered convex function chasing* ((κ, γ)CFC) comprised of all those problem instances of CFC in which each f_t is (κ, γ)-well-centered. [108] showed that the "Move towards Minimizer" algorithm is $(2 + 2\sqrt{2})2^{\gamma/2}\kappa$-competitive for (κ, γ)CFC on $\mathbb{R}^d$ equipped with the ℓ^2 norm.

Bicompetitive meta-algorithms give robust and consistent algorithms

In this section, we briefly justify the claim that if Meta is a (c, r)-bicompetitive meta-algorithm for CFC, Adv is an advice algorithm, and Rob is a b-competitive online algorithm for a subclass of CFC, then Meta(Adv, Rob) is c-consistent and rb-robust for that subclass. This is straightforward to see for non-CBC subclasses, or more generally, for any subclass of CFC which does not involve hard constraints on the decisions $\mathbf{x}_t$. In particular, Rob being b-competitive means that $\mathrm{C}_{\mathrm{Rob}} \leq b \cdot \mathrm{C}_{\mathrm{Opt}}$, and so (c, r)-bicompetitiveness of Meta implies that both $\mathrm{C}_{\mathrm{Meta(Adv,Rob)}} \leq c \cdot \mathrm{C}_{\mathrm{Adv}}$ and $\mathrm{C}_{\mathrm{Meta(Adv,Rob)}} \leq r \cdot \mathrm{C}_{\mathrm{Rob}} \leq rb \cdot \mathrm{C}_{\mathrm{Opt}}$, as desired.

The only subclasses that require more careful justification are those, such as CBC, with hard constraints on the decisions. However, so long as the advice always gives feasible decisions—e.g., in the CBC case, $\mathrm{Adv} \in \mathcal{A}_{\mathrm{CBC}}$, so $\mathrm{Adv}_t \in K_t$ for each t—then we can obtain the same result by applying the reasoning from Section 2.A on equivalent CFC reformulations of instances with hard constraints. That is, on any instance of the subclass, we must simply run Meta(Adv, Rob) on its equivalent reformulation as a CFC instance, and we thereby obtain the same guarantees of c-consistency and rb-robustness for the subclass.

2.B Switching Algorithms

Proof of Proposition 2.3.2

The proof follows the argument of [118, Theorem 5] and uses a similar line of reasoning as [75, Theorems 1, 18] in applying an algorithm for linear search to CFC.

Each value of i encountered in the execution of Algorithm 1 is taken to refer to a *phase* of the algorithm; every decision $\mathbf{x}_t$ made during a particular value of i is said to take place during the i^{th} phase. Our strategy will be to bound the cost that the algorithm incurs in each phase i, including the cost it takes to switch from the last decision of the previous phase $i-1$.

As a base case, consider $i = 0$. The total cost incurred by the algorithm during this phase is bounded by $b^i = b^0 = 1$, by line 4 of the algorithm.

Now consider phase $i > 0$, and assume that i is odd; after proving the cost bound for phase i in the odd case, we will state the corresponding bound for the even case, which follows a nearly identical argument. Let $\underline{t}$ be the last timestep in the $(i-1)^{\text{th}}$ phase—that is, $\underline{t}$ is defined such that $C_{\text{Adv}}(1, \underline{t}+1) > b^{i-1}$. We will assume that $C_{\text{Adv}}(1, \underline{t}) \leq b^{i-1}$, i.e., the algorithm makes at least one decision during phase $(i-1)$, selecting $\mathbf{x}_{\underline{t}} = \text{Adv}_{\underline{t}}$; but the upper bound we obtain will also apply to the case where the $(i-1)^{\text{th}}$ phase is vacuous. Let $\bar{t}$ be the last timestep corresponding to phase i, i.e., $\bar{t} \geq \underline{t}$ is defined such that $C_{\text{Rob}}(1, \bar{t}+1) > \delta b^i$. If $\bar{t} = \underline{t}$, then clearly no decisions are made during phase i, so no cost is incurred during this phase. On the other hand, if $\bar{t} > \underline{t}$, then certainly $C_{\text{Rob}}(1, \bar{t}) \leq \delta b^i$, so the cost incurred by Switch during phase i, starting from its position at time $\underline{t}$, can be bounded as

$$
\begin{aligned}
\mathrm{C}_{\textsc{Switch}}(\underline{t}+1,\bar{t}) &= \sum_{t=\underline{t}+1}^{\bar{t}} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| \\
&= f_{\underline{t}+1}(\mathbf{x}_{\underline{t}+1}) + \|\mathbf{x}_{\underline{t}+1} - \mathbf{x}_{\underline{t}}\| + \sum_{t=\underline{t}+2}^{\bar{t}} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| \\
&\le f_{\underline{t}+1}(\mathbf{x}_{\underline{t}+1}) + \|\mathbf{x}_{\underline{t}+1} - \mathbf{x}_0\| + \|\mathbf{x}_{\underline{t}} - \mathbf{x}_0\| + \sum_{t=\underline{t}+2}^{\bar{t}} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| \\
&= f_{\underline{t}+1}(\textsc{Rob}_{\underline{t}+1}) + \|\textsc{Rob}_{\underline{t}+1} - \mathbf{x}_0\| + \|\textsc{Adv}_{\underline{t}} - \mathbf{x}_0\| \\
&\qquad + \sum_{t=\underline{t}+2}^{\bar{t}} f_t(\textsc{Rob}_t) + \|\textsc{Rob}_t - \textsc{Rob}_{t-1}\| \\
&\le \mathrm{C}_{\textsc{Rob}}(1,\bar{t}) + \mathrm{C}_{\textsc{Adv}}(1,\underline{t}) \\
&\le \delta b^i + b^{i-1},
\end{aligned}
$$

where the first two bounds use the triangle inequality, and the last bound follows by construction of $\bar{t}$ and $\underline{t}$. By a very similar argument, if i is even, we can bound the cost incurred by Switch during phase i as $\delta b^{i-1} + b^i$. Then the total cost expenditure of Switch through the end of some phase $N > 0$ is at most

$$
\begin{aligned}
& b^0 + \sum_{i=0}^{\lfloor \frac{N-1}{2} \rfloor} (\delta b^{2i+1} + b^{2i}) + \sum_{j=1}^{\lfloor \frac{N}{2} \rfloor} (\delta b^{2j-1} + b^{2j}) \\
= & \begin{cases} b^N + 2\sum_{i=0}^{\frac{N}{2}-1} (b^{2i} + \delta b^{2i+1}) & \text{if } N \text{ is even} \\ \delta b^N + 2b^{N-1} + 2\sum_{i=0}^{\frac{N-3}{2}} (b^{2i} + \delta b^{2i+1}) & \text{if } N \text{ is odd.} \end{cases}
\end{aligned}
\tag{2.1}
$$

Suppose then that the instance ends at time T *during* phase N. We break into cases depending on the value of N.

First, if $N = 0$, then clearly $\mathrm{C}_{\textsc{Switch}} = \mathrm{C}_{\textsc{Adv}}$, so Switch is 1-competitive with respect to Adv. Moreover, since $\mathrm{C}_{\textsc{Adv}} \le b^0 = 1$, then by the assumption in the proposition statement that $\mathrm{C}_{\textsc{Rob}} \ge 1$, it follows that Switch is at most 1-competitive with respect to Rob.

Second, if $N = 1$, then $\mathrm{C}_{\textsc{Switch}} \le 2b^0 + \mathrm{C}_{\textsc{Rob}} = 2 + \mathrm{C}_{\textsc{Rob}}$. Since $\mathrm{C}_{\textsc{Adv}} > 1$ and $\mathrm{C}_{\textsc{Rob}} \le \delta b$ (due to the instance ending at phase $N = 1$), this means that Switch is at most $(2+\delta b)$-competitive with respect to Adv. Moreover, by assumption $\mathrm{C}_{\textsc{Rob}} \ge 1$, Switch is at most 3-competitive with respect to Rob.

Next, suppose $N > 1$ and N is even. Then we have

$$\mathrm{C}_{\textsc{Switch}} \leq 2 \sum_{i=0}^{\frac{N}{2}-1} (b^{2i} + \delta b^{2i+1}) + \mathrm{C}_{\textsc{Adv}}$$

which follows by applying (2.1) to bound cost through phase $(N-1)$, and bounding the remaining cost by $\delta b^{N-1} + \mathrm{C}_{\textsc{Adv}}$, i.e., the cost to switch back to Adv and follow it until the instance ends. Then note that $\mathrm{C}_{\textsc{Adv}} \geq b^{N-2}$ by definition of phase; introducing the substitution $2k := N - 2$, we find that the competitive ratio of Switch with respect to Adv is bounded as

$$\begin{aligned}
\frac{\mathrm{C}_{\textsc{Switch}}}{\mathrm{C}_{\textsc{Adv}}} &\leq 1 + 2\frac{\sum_{i=0}^{k}(b^{2i} + \delta b^{2i+1})}{b^{2k}} \\
&= 1 + 2\left(\frac{b^{2k+2}-1}{b^{2k}(b^2-1)} + \delta b \frac{b^{2k+2}-1}{b^{2k}(b^2-1)}\right) \\
&\leq 1 + 2\left(\frac{b^2}{b^2-1} + \delta \frac{b^3}{b^2-1}\right).
\end{aligned}$$

On the other hand, we know that $\mathrm{C}_{\textsc{Adv}} \leq b^N$ and $\mathrm{C}_{\textsc{Rob}} \geq \delta b^{N-1}$, so by similar reasoning the competitive ratio of Switch with respect to Rob is bounded as

$$\begin{aligned}
\frac{\mathrm{C}_{\textsc{Switch}}}{\mathrm{C}_{\textsc{Rob}}} &\leq \frac{b}{\delta} + 2\frac{\sum_{i=0}^{k}(b^{2i} + \delta b^{2i+1})}{\delta b^{2k+1}} \\
&\leq \frac{b}{\delta} + 2\left(\frac{b}{\delta(b^2-1)} + \frac{b^2}{b^2-1}\right).
\end{aligned}$$

Finally, consider $N > 1$ for odd N. Then

$$\mathrm{C}_{\textsc{Switch}} \leq 2 \sum_{i=0}^{\frac{N-1}{2}} b^{2i} + \sum_{i=0}^{\frac{N-3}{2}} \delta b^{2i+1} + \mathrm{C}_{\textsc{Rob}}.$$

Noting that $\mathrm{C}_{\textsc{Rob}} \leq \delta b^N$, $\mathrm{C}_{\textsc{Adv}} \geq b^{N-1}$, and making the substitution $2k = N - 1$, we obtain that the competitive ratio of Switch with respect to Adv is bounded as

$$\begin{aligned}
\frac{\mathrm{C}_{\textsc{Switch}}}{\mathrm{C}_{\textsc{Adv}}} &\leq \delta b + 2\frac{\sum_{i=0}^{k} b^{2i} + \sum_{i=0}^{k-1} \delta b^{2i+1}}{b^{2k}} \\
&\leq \delta b + 2\left(\frac{b^2}{b^2-1} + \delta \frac{b}{b^2-1}\right).
\end{aligned}$$

On the other hand, we know that $\mathrm{C}_{\textsc{Rob}} \geq \delta b^{N-2} = \delta b^{2k-1}$. Thus the competitive ratio of Switch with respect to Rob is bounded as

$$\begin{aligned}\frac{\mathrm{C}_{\textsc{Switch}}}{\mathrm{C}_{\textsc{Rob}}} &\leq 1 + 2\frac{\sum_{i=0}^{k} b^{2i} + \sum_{i=0}^{k-1} \delta b^{2i+1}}{\delta b^{2k-1}} \\ &\leq 1 + 2\left(\frac{1}{\delta}\frac{b^3}{b^2-1} + \frac{b^2}{b^2-1}\right).\end{aligned}$$

Combining these various cases, we obtain that Switch is

$$\left(1 + 2\left(\frac{b^2}{b^2-1} + \delta\frac{b^3}{b^2-1}\right), 1 + 2\left(\frac{b^2}{b^2-1} + \frac{1}{\delta}\frac{b^3}{b^2-1}\right)\right)\text{–bicompetitive}$$

with respect to $(\textsc{Adv}, \textsc{Rob})$. Introducing an auxiliary parameter γ and making the substitutions $\delta \leftarrow b\gamma^2 - b^{-1}$, $b \leftarrow \sqrt{\gamma^{-2}+1}$, and $\gamma \leftarrow \sqrt{\frac{\epsilon}{4}}$, we arrive at the bicompetitive bound in terms of ϵ stated in the proposition. □

Proof of Theorem 2.3.3

We consider the setting of $\mathbb{R}^d$ with the ℓ^2 norm, where the advice Adv is adversarial and Rob is an *arbitrary* b-competitive algorithm for CBC, with $b < \infty$; Alg is any algorithm that switches between Rob and Adv. For simplicity of presentation, we will assume that $\sqrt{d}$ is an integer. Rob is assumed to be advice-agnostic, i.e., the behavior of Adv does not impact the decisions made by Rob (nor does the behavior of Alg, since Alg itself depends on both Adv and Rob). We construct a lower bound in the spirit of the standard example of chasing faces of the hypercube. At a high level, the CBC instance we construct has two phases: the first is comprised of multiple subphases in which an affine subspace is chosen adversarially and is repeatedly served until Rob has "almost" stopped moving. This phase lasts either until $3\sqrt{d}$ subphases have concluded, or until the first time that Alg coincides with Adv at the end of a subphase, whichever happens sooner. If the former holds, i.e., if Alg ends each of the $3\sqrt{d}$ subphases at Rob, then the instance is done. Otherwise, the second phase begins: there are a few different cases, but generally, the same affine subspace is served repeatedly while Adv slowly drifts away from Rob until Alg switches back to Rob. Then, the final body is simply the last advice decision as a singleton, forcing Alg to move back to the advice, and the instance concludes.

We now describe the lower bound in more specific detail. Since Rob is advice-agnostic, we may begin by describing its behavior before specifying the behavior of Adv and reasoning about the switching algorithm Alg. We denote by $\mathbf{e}_j \in \mathbb{R}^d$,

$j \in [d]$ the j^{th} standard unit basis vector, which is 1 in its j^{th} entry and 0 elsewhere. Choose any $\delta > 0$. The starting position is $\mathbf{x}_0 = \mathbf{0}$.

Phase one. At time $t = 1$, the served body K_1 is the hyperplane forcing the first coordinate to be $z_1 := 1$:

$$K_1 = \left\{\mathbf{x} : \mathbf{x}^\top \mathbf{e}_1 = z_1\right\}.$$

This same hyperplane K_1 is then repeatedly served until the time m_1 at which Rob is almost stationary. That is, the time $m_1 < \infty$ is chosen to satisfy the property that the cumulative cost incurred by Rob after time m_1, if K_1 were repeated indefinitely thereafter, is bounded above by δ. Such a time m_1 must exist, since Rob is b-competitive, the offline optimal cost for the instance comprised of repeated K_1s is 1, and the tail of a convergent series converges to zero. At time m_1, if Alg's decision coincides with that of Adv, i.e., if $\textsc{Alg}_{m_1} = \textsc{Adv}_{m_1}$ (note we will define the behavior of Alg and Adv later on), then we say that phase one is complete and we move on to phase two below. Otherwise, we continue to the next subphase in phase one as follows.

Let $z_2 := -\operatorname{sgn}(\textsc{Rob}_{m_1,2})$ be the negative of the sign of Rob's 2$^{\text{nd}}$ entry at time m_1 (defaulting to 1 if $\textsc{Rob}_{m_1,2} = 0$). At time $t = m_1 + 1$, we serve a new affine subspace K_2 defined as

$$K_2 = \left\{\mathbf{x} : \mathbf{x}^\top \mathbf{e}_i = z_i, i = 1, 2\right\}.$$

Note that this forces Rob to incur cost at least 1 at time $m_1 + 1$. This same body is repeated until the time m_2 at which Rob is almost stationary. That is, just as before, m_2 is defined as the time at which, if K_2 were repeated indefinitely from time $m_2 + 1$ onward, Rob would incur total cost no more than δ after time m_2. For the same reason as before, $m_2 < \infty$ is certain to exist by b-competitiveness of Rob. If $\textsc{Alg}_{m_2} = \textsc{Adv}_{m_2}$, then we say that phase one is complete and move on to phase two below. Otherwise, we continue to the next subphase in phase one.

The remaining subphases in phase one are constructed similarly: for each $j = 3, \ldots, 3\sqrt{d}$, we define $z_j := -\operatorname{sgn}(\textsc{Rob}_{m_{j-1},j})$ to be the negative of the sign of Rob's j^{th} entry at time m_{j-1}, and at time $t = m_{j-1} + 1$, we serve a new affine subspace K_j defined as

$$K_j = \left\{\mathbf{x} : \mathbf{x}^\top \mathbf{e}_i = z_i, i = 1, \ldots, j\right\}, \tag{2.2}$$

which forces Rob to incur cost at least 1. This body K_j is then repeated until the time m_j at which Rob is almost stationary, i.e., after which it would incur cumulative cost no more than δ, were K_j to be repeated indefinitely. Then, if $\textsc{Alg}_{m_j} = \textsc{Adv}_{m_j}$, we say that phase one is complete and move on to phase two below. Otherwise, we remain in phase one and repeat this step with an incremented value of j. Once the subphase corresponding to $j = 3\sqrt{d}$ is completed, then the instance is concluded without moving on to phase two.

Behavior of the advice. We specify the behavior of $\textsc{Adv}$ based on the behavior of $\textsc{Rob}$ on the (possibly) counterfactual instance wherein phase one runs to termination without moving to phase two. That is, let $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{m_{3\sqrt{d}}}$ be the decisions of $\textsc{Rob}$ on an auxiliary CBC instance where K_1 is served from time 1 through m_1, K_2 is served from time $m_1 + 1$ through m_2, and so on, terminating with $K_{3\sqrt{d}}$ being served from time $m_{3\sqrt{d}-1} + 1$ through $m_{3\sqrt{d}}$. Then define

$$\mathbf{a} = \operatorname*{arg\,max}_{\mathbf{x} \in \{\pm 1\}^{d-3\sqrt{d}}} \min_{j=1,\ldots,3\sqrt{d}} \|\mathbf{x} - \mathbf{r}_{m_j, 3\sqrt{d}+1:}\|_{\ell^2}, \tag{2.3}$$

where $\mathbf{r}_{m_j,3\sqrt{d}+1:}$ is the vector obtained by dropping the first $3\sqrt{d}$ entries in $\mathbf{r}_{m_j}$. Thus, $\mathbf{a}$ is the corner of the hypercube $\{\pm 1\}^{d-3\sqrt{d}}$ that is farthest (in ℓ^2) from any of the subvectors comprised of the last $d - 3\sqrt{d}$ entries of the decisions $\mathbf{r}_{m_1}, \ldots, \mathbf{r}_{m_{3\sqrt{d}}}$ made by $\textsc{Rob}$ at the conclusion of the phase one subphases. Then we define the advice's phase one behavior simply as follows: at time 1, the advice immediately moves to the point

$$\hat{\mathbf{a}} = (z_1, \ldots, z_{3\sqrt{d}}, a_1, \ldots, a_{d-3\sqrt{d}}),$$

and it remains there until phase one is completed.

Phase two. Fix $\epsilon > 0$. Suppose that phase one terminates at time m_j, where $j < 3\sqrt{d}$ (since if $j = 3\sqrt{d}$, then the instance ends without moving on to phase two). Thus it is the case that $\textsc{Alg}_{m_j} = \textsc{Adv}_{m_j} = \hat{\mathbf{a}}$, and $\textsc{Rob}_{m_j} = \mathbf{r}_{m_j}$. Then the instance splits into two cases:

1.) Suppose that $\|\mathbf{a} - \mathbf{r}_{m_j,3\sqrt{d}+1:}\|_{\ell^2} \geq \sqrt{d - 3\sqrt{d}}$, and define $\mathbf{v} = \frac{\mathbf{a} - \mathbf{r}_{m_j,3\sqrt{d}+1:}}{\|\mathbf{a} - \mathbf{r}_{m_j,3\sqrt{d}+1:}\|_{\ell^2}}$. Then at each time $t = m_j + 1, \ldots, m_j + k$ (where k will be defined later), we serve the body K_j again. By our selection of m_j, the robust algorithm $\textsc{Rob}$ will remain δ-close to its decision $\textsc{Rob}_{m_j}$, since we are simply continuing to serve the same body. However, at each of these times, we make the advice move to the point

$$\textsc{Adv}_t = \left(z_1, \ldots, z_{3\sqrt{d}}, a_1 + (t - m_j)\epsilon v_1, \ldots, a_{d-3\sqrt{d}} + (t - m_j)\epsilon v_{d-3\sqrt{d}}\right).$$

That is, at each time $t = m_j + 1, \ldots, m_j + k$, the advice takes a step of length ϵ in the direction $\mathbf{v}$ in its last $d - 3\sqrt{d}$ coordinates. Then k is chosen such that

$m_j + k$ is the first time after m_j at which $\textsc{Alg}_{m_j+k} = \textsc{Rob}_{m_j+k}$, i.e., the first time at which the algorithm switches back to Rob after following the advice. Note that $k < \infty$, by the assumption that Alg has finite robustness. Then the final body is chosen as

$$K_{\text{fin}} = \left\{\textsc{Adv}_{m_j+k}\right\} = \left\{\left(z_1, \ldots, z_{3\sqrt{d}}, a_1 + k\epsilon v_1, \ldots, a_{d-3\sqrt{d}} + k\epsilon v_{d-3\sqrt{d}}\right)\right\},$$

which allows the advice to stay put while Rob and Alg must move back to coincide with it.

2.) Suppose that $\|\mathbf{a} - \mathbf{r}_{m_j,3\sqrt{d}+1:}\|_{\ell^2} < \sqrt{d - 3\sqrt{d}}$. Since $\mathbf{a}$ maximizes the objective of (2.3), then it must hold that

$$\min_{i=1,\ldots,3\sqrt{d}} \| - \mathbf{a} - \mathbf{r}_{m_i,3\sqrt{d}+1:}\|_{\ell^2} < \sqrt{d - 3\sqrt{d}}. \tag{2.4}$$

Let i^* be the minimizing index in (2.4); note that $i^* \neq j$, since otherwise,

$$\begin{aligned} 2\sqrt{d - 3\sqrt{d}} &= \|2\mathbf{a}\| \\ &\leq \|\mathbf{a} - \mathbf{r}_{m_j,3\sqrt{d}+1:}\|_{\ell^2} + \|\mathbf{a} + \mathbf{r}_{m_j,3\sqrt{d}+1:}\|_{\ell^2} \quad \text{by the triangle inequality} \\ &< 2\sqrt{d - 3\sqrt{d}} \end{aligned}$$

giving a contradiction. Then the instance splits into two further subcases:

(a) Suppose that $i^* < j$. Then, just as in case 1.), at each time $t = m_j + 1, \ldots, m_j + k$, we serve the body K_j again. At each of these times, we make the advice move to the point

$$\textsc{Adv}_t = \left(z_1, \ldots, z_{3\sqrt{d}}, a_1 + (t - m_j)\epsilon v_1, \ldots, a_{d-3\sqrt{d}} + (t - m_j)\epsilon v_{d-3\sqrt{d}}\right),$$

where $\mathbf{v} = \frac{\mathbf{a}-\mathbf{r}_{m_j,3\sqrt{d}+1:}}{\|\mathbf{a}-\mathbf{r}_{m_j,3\sqrt{d}+1:}\|_{\ell^2}}$ just as in case 1.). Just as in case 1.), k is chosen such that $m_j + k$ is the first time after m_j at which $\textsc{Alg}_{m_j+k} = \textsc{Rob}_{m_j+k}$, i.e., the first time at which the algorithm switches back to Rob after following the advice. Then the final body is chosen as

$$\begin{aligned} K_{\text{fin}} &= \left\{\textsc{Adv}_{m_j+k}\right\} \\ &= \left\{\left(z_1, \ldots, z_{3\sqrt{d}}, a_1 + k\epsilon v_1, \ldots, a_{d-3\sqrt{d}} + k\epsilon v_{d-3\sqrt{d}}\right)\right\}, \end{aligned}$$

which allows the advice to stay put while Rob and Alg must move back to coincide with it.

(b) Suppose that $i^* > j$. Then for each $l = j+1, \ldots, i^*$, serve the body K_l as defined in (2.2) from time $m_{l-1}+1$ through m_l, while keeping the advice at the same point $\hat{\mathbf{a}}$. Since Rob is advice agnostic and this sequence of bodies coincides with the remainder of the phase one sequence of bodies, it will be the case that $\textsc{Rob}_{m_{i^*}} = \mathbf{r}_{m_{i^*}}$. Then, finally, we split into two further subcases.

(i) If $\textsc{Alg}_{m_{i^*}} = \textsc{Rob}_{m_{i^*}} = \mathbf{r}_{m_{i^*}}$, then simply choose the final body as

$$K_{\text{fin}} = \{\hat{\mathbf{a}}\},$$

which allows the advice to stay put while Rob and Alg must move back to coincide with it.

(ii) If $\textsc{Alg}_{m_{i^*}} = \textsc{Adv}_{m_{i^*}} = \hat{\mathbf{a}}$, then proceed similarly to subcase (a): for each time $t = m_{i^*}+1, \ldots, m_{i^*}+k$, we serve the body K_{i^*} again and make the advice move to the point

$$\textsc{Adv}_t = \left(z_1, \ldots, z_{3\sqrt{d}}, a_1 + (t-m_j)\epsilon v_1, \ldots, a_{d-3\sqrt{d}} + (t-m_j)\epsilon v_{d-3\sqrt{d}}\right),$$

where this time $\mathbf{v} = \frac{\mathbf{a}-\mathbf{r}_{m_{i^*},3\sqrt{d}+1:}}{\|\mathbf{a}-\mathbf{r}_{m_{i^*},3\sqrt{d}+1:}\|_{\ell^2}}$. Just as in subcase (a), k is chosen such that $m_{i^*}+k$ is the first time after m_{i^*} at which $\textsc{Alg}_{m_{i^*}+k} = \textsc{Rob}_{m_{i^*}+k}$, i.e., the first time at which the algorithm switches back to Rob after following the advice starting from time m_{i^*}. Then the final body is simply chosen as

$$\begin{aligned} K_{\text{fin}} &= \left\{\textsc{Adv}_{m_j+k}\right\} \\ &= \left\{\left(z_1, \ldots, z_{3\sqrt{d}}, a_1 + k\epsilon v_1, \ldots, a_{d-3\sqrt{d}} + k\epsilon v_{d-3\sqrt{d}}\right)\right\}, \end{aligned}$$

which allows the advice to stay put while forcing Rob and Alg to move to coincide with it.

Cost analysis. Let us now tally costs for each of the cases of the instance to prove the result.

Consider the initial case where the instance never makes it out of phase one; this means that $3\sqrt{d}$ subphases occur in phase one, and Alg finishes each subphase at the Rob decision. Since $\|\mathbf{r}_{m_j} - \mathbf{r}_{m_{j-1}}\|_{\ell^2} \geq 1$ for each $j = 1, \ldots, 3\sqrt{d}$ (where $\mathbf{r}_{m_0} := \mathbf{r}_0 = \mathbf{x}_0$), this means that Alg incurs cost at least $3\sqrt{d}$, whereas the advice, which moves immediately to $\hat{\mathbf{a}} \in \{\pm 1\}^d$ and stays there throughout the entire instance, incurs cost $\sqrt{d}$. Thus Alg is at least 3-consistent, and we are done.

Now, we turn to each of the cases within which the instance makes it to phase two. First, consider case 1.). Since $\textsc{Alg}_{m_j} = \textsc{Adv}_{m_j} = \hat{\mathbf{a}}$, the cost incurred by $\textsc{Alg}$ through time m_j is at least $\sqrt{d}$. Then from time m_j to $m_j + k - 1$ while $\textsc{Alg}$ is following the advice, $\textsc{Alg}$ incurs cost $\|(k-1)\epsilon\mathbf{v}\|_{\ell^2} = (k-1)\epsilon$. At time $m_j + k$, $\textsc{Alg}$ switches back to $\textsc{Rob}$, incurring cost at least

$$\begin{aligned}
\|\textsc{Adv}_{m_j+k-1} - \textsc{Rob}_{m_j+k}\|_{\ell^2} &\geq \|\textsc{Adv}_{m_j+k-1} - \textsc{Rob}_{m_j}\|_{\ell^2} - \|\textsc{Rob}_{m_j} - \textsc{Rob}_{m_j+k}\|_{\ell^2} \\
&\geq \|\textsc{Adv}_{m_j+k-1} - \textsc{Rob}_{m_j}\|_{\ell^2} - \delta \\
&\geq \|\textsc{Adv}_{m_j+k-1,3\sqrt{d}+1:} - \textsc{Rob}_{m_j,3\sqrt{d}+1:}\|_{\ell^2} - \delta \\
&= \|(\mathbf{a} + (k-1)\epsilon\mathbf{v}) - \mathbf{r}_{m_j,3\sqrt{d}+1:}\|_{\ell^2} - \delta \\
&= \|\mathbf{a} - \mathbf{r}_{m_j,3\sqrt{d}+1:}\|_{\ell^2} + (k-1)\epsilon - \delta \\
&\geq \sqrt{d - 3\sqrt{d}} + (k-1)\epsilon - \delta. \qquad (2.5)
\end{aligned}$$

Finally, by an analogous argument to (2.5), to switch back to $\textsc{Adv}$, $\textsc{Alg}$ incurs a cost of at least $\sqrt{d-3\sqrt{d}} + k\epsilon - \delta$. In sum, $\textsc{Alg}$ incurs a total cost of $\sqrt{d} + 2\sqrt{d-3\sqrt{d}} + (3k-2)\epsilon - 2\delta$. On the other hand, $\textsc{Adv}$ incurs a total cost of $\sqrt{d} + k\epsilon$. Then the consistency of $\textsc{Alg}$ is

$$\frac{\sqrt{d} + 2\sqrt{d-3\sqrt{d}} + (3k-2)\epsilon - 2\delta}{\sqrt{d} + k\epsilon}$$

which can be made arbitrarily close to 3 by choosing ϵ and δ small and taking d arbitrarily large.

Next, let us move to case 2.). First, we set up some preliminaries. Let us call $\rho = \sqrt{d-3\sqrt{d}} - \|\mathbf{a} - \mathbf{r}_{m_j,3\sqrt{d}+1:}\|_{\ell^2}$, and note that $\rho > 0$. Since $\|\mathbf{a} - \mathbf{r}_{m_j,3\sqrt{d}+1:}\|_{\ell^2} \geq \|-\mathbf{a} - \mathbf{r}_{m_{i^*},3\sqrt{d}+1:}\|_{\ell^2}$, we have that

$$\|-\mathbf{a} - \mathbf{r}_{m_{i^*},3\sqrt{d}+1:}\|_{\ell^2} \leq \sqrt{d-3\sqrt{d}} - \rho \qquad (2.6)$$

and hence

$$\begin{aligned}
\|\mathbf{r}_{m_{i^*},3\sqrt{d}+1:}\|_{\ell^2} &\geq \|-\mathbf{a}\| - \|-\mathbf{a} - \mathbf{r}_{m_{i^*},3\sqrt{d}+1:}\|_{\ell^2} \\
&\geq \rho. \qquad (2.7)
\end{aligned}$$

Then

$$
\begin{aligned}
\|\hat{\mathbf{a}} - \mathbf{r}_{m_{i^*}}\|_{\ell^2} &\geq \|\mathbf{a} - \mathbf{r}_{m_{i^*}, 3\sqrt{d}+1:}\|_{\ell^2} && (2.8)\\
&\geq \left\|\mathbf{a} - \Pi_{B(-\mathbf{a}, \sqrt{d-3\sqrt{d}-\rho})}\mathbf{a}\right\|_{\ell^2} && (2.9)\\
&= \left\|2\mathbf{a} - \Pi_{B(\mathbf{0}, \sqrt{d-3\sqrt{d}-\rho})}2\mathbf{a}\right\|_{\ell^2} && (2.10)\\
&= \left\|2\mathbf{a} - (\sqrt{d-3\sqrt{d}} - \rho)\frac{\mathbf{a}}{\|\mathbf{a}\|_{\ell^2}}\right\|_{\ell^2} && (2.11)\\
&= \left\|\mathbf{a} + \rho\frac{\mathbf{a}}{\|\mathbf{a}\|_{\ell^2}}\right\|_{\ell^2}\\
&= \sqrt{d-3\sqrt{d}} + \rho && (2.12)
\end{aligned}
$$

where $\Pi_K\mathbf{x}$ denotes the projection of the point $\mathbf{x}$ onto the convex body K, (2.8) follows from $\mathbf{a}$ and $\mathbf{r}_{m_{i^*}, 3\sqrt{d}+1:}$ being subvectors of $\hat{\mathbf{a}}$ and $\mathbf{r}_{m_{i^*}}$, respectively, (2.9) follows from (2.6) and non-expansivity of the projection, (2.10) follows from translation, (2.11) applies the fact that the projection onto an origin-centered ball is just a radial projection, and (2.12) follows from $\|\mathbf{a}\|_{\ell^2} = \sqrt{d-3\sqrt{d}}$.

Now, let us consider the subcases, starting with subcase (a). Since $i^* < j$, we know that $\textsc{Alg}_{m_{i^*}} = \textsc{Rob}_{m_{i^*}} = \mathbf{r}_{m_{i^*}}$. Then by (2.7), $\textsc{Alg}$ incurs cost at least ρ to get to $\mathbf{r}_{m_{i^*}}$, and by (2.12) it incurs another cost of at least $\sqrt{d-3\sqrt{d}} + \rho$ to get to $\textsc{Adv}_{m_j} = \hat{\mathbf{a}}$. From time m_j to $m_j + k - 1$ while $\textsc{Alg}$ is following the advice, $\textsc{Alg}$ incurs cost $(k-1)\epsilon$. Then at time $m_j + k$, $\textsc{Alg}$ switches back to $\textsc{Rob}$, and by a similar analysis to that in (2.5) done for case 1.), it incurs cost at least $\sqrt{d-3\sqrt{d}} - \rho + (k-1)\epsilon - \delta$ to do so. Finally, to switch back to $\textsc{Adv}$, $\textsc{Alg}$ incurs a cost of at least $\sqrt{d-3\sqrt{d}} - \rho + k\epsilon - \delta$. Then in sum, $\textsc{Alg}$ has incurred a total cost of $3\sqrt{d-3\sqrt{d}} + (3k-2)\epsilon - 2\delta$ in this instance case. On the other hand, $\textsc{Adv}$ incurs a total cost of $\sqrt{d} + k\epsilon$, so $\textsc{Alg}$ has consistency

$$
\frac{3\sqrt{d-3\sqrt{d}} + (3k-2)\epsilon - 2\delta}{\sqrt{d} + k\epsilon},
$$

which can be made arbitrarily close to 3 by choosing ϵ and δ small and taking d arbitrarily large.

Now, we move to subcase (b), beginning first with (i). $\textsc{Alg}$ spends $\sqrt{d}$ to get to $\textsc{Adv}_{m_j} = \hat{\mathbf{a}}$ in the first place, and then by (2.12) it spends cost at least $\sqrt{d-3\sqrt{d}} + \rho$ to get to $\textsc{Rob}_{m_{i^*}}$ at time m_{i^*}. Finally, it spends at least another $\sqrt{d-3\sqrt{d}} + \rho$ to get back to $\hat{\mathbf{a}}$ for the final timestep. Thus in sum, $\textsc{Alg}$ incurs cost $\sqrt{d} + 2(\sqrt{d-3\sqrt{d}} + \rho)$, whereas Adv incurs cost $\sqrt{d}$, giving a consistency of

$$\frac{\sqrt{d} + 2(\sqrt{d - 3\sqrt{d}} + \rho)}{\sqrt{d}},$$

which even for arbitrarily small $\rho > 0$ can be made arbitrarily close to 3 by choosing d sufficiently large.

Finally, we consider scenario (ii) in subcase (b). Alg first spends $\sqrt{d}$ to get to $\text{Adv}_{m_j} = \hat{\mathbf{a}}$, and then from time $m_{i^*} + 1$ through $m_{i^*} + k - 1$ it incurs cost $(k-1)\epsilon$ to follow the advice. Using (2.12) and reasoning analogous to that in (2.5), Alg incurs cost $\sqrt{d - 3\sqrt{d}} + \rho + (k-1)\epsilon - \delta$ to switch back to Rob at time $m_{i^*} + k$, and finally, it incurs cost $\sqrt{d - 3\sqrt{d}} + \rho + k\epsilon - \delta$ to switch back to the advice in the final timestep. Thus in sum, Alg incurs cost at least $\sqrt{d} + 2(\sqrt{d - 3\sqrt{d}} + \rho) + (3k-2)\epsilon - 2\delta$, while Adv incurs cost $\sqrt{d} + k\epsilon$. Thus Alg has consistency

$$\frac{\sqrt{d} + 2(\sqrt{d - 3\sqrt{d}} + \rho) + (3k-2)\epsilon - 2\delta}{\sqrt{d} + k\epsilon},$$

which, even for very small $\rho > 0$, can be made arbitrarily close to 3 by choosing ϵ, δ small and taking d sufficiently large.

□

Proof of Proposition 2.3.4

Let us first recall Theorem 2.1 of [101], which characterizes the cost incurred by moving to the Steiner point of each nested body.

Theorem 2.B.1 ([101, Theorem 2.1]). *Let $\mathbf{x}_0 = \mathbf{0}$ and $K_1 \subseteq B(\mathbf{0}, r)$ for some $r > 0$. Then following the Steiner point of each nested body K_t incurs total movement cost no more than rd.*

We now prove Proposition 2.3.4. For clarity, we abbreviate NestedSwitch in this proof as NS.

If NS only ever follows Adv, then $\epsilon \cdot C_{\text{Adv}} < r(d+2)$ and $C_{\text{NS}} = C_{\text{Adv}}$, so $C_{\text{NS}} \leq \frac{r(d+2)}{\epsilon}$. Thus NS is 1-competitive with respect to Adv and $\frac{r(d+2)}{\epsilon}$-robust, since $C_{\text{Opt}} \geq 1$.

On the other hand, if NS only ever follows Rob, then $C_{\text{NS}} = C_{\text{Rob}}$ and $\epsilon \cdot C_{\text{Adv}} \geq r(d+2)$. Since Rob just follows the Steiner point of each nested body, we have

$\mathrm{C}_{\textsc{Rob}} \leq r + rd$, where the rd comes from Theorem 2.B.1 and the extra factor of r arises from the triangle inequality applied to the $t = 1$ movement:

$$\|\mathbf{s}_1 - \mathbf{x}_0\|_{\ell^2} \leq \|\mathbf{s}_1 - \mathbf{y}\|_{\ell^2} + \|\mathbf{y} - \mathbf{x}_0\|_{\ell^2} \leq \|\mathbf{s}_1 - \mathbf{y}\|_{\ell^2} + r.$$

Thus $\mathrm{C}_{\mathrm{NS}} = \mathrm{C}_{\textsc{Rob}} \leq r(d+1) \leq (1+\epsilon)\mathrm{C}_{\textsc{Adv}}$, and the desired robustness also holds.

Finally, suppose NS switches to $\textsc{Rob}$ at time $t \in [T]$; i.e., $\mathrm{NS}_1 = \textsc{Adv}_1, \ldots, \mathrm{NS}_{t-1} = \textsc{Adv}_{t-1}, \mathrm{NS}_t = \textsc{Rob}_t, \ldots, \mathrm{NS}_T = \textsc{Rob}_T$. We know that

$$\mathrm{C}_{\mathrm{NS}}(1, t-1) = \mathrm{C}_{\textsc{Adv}}(1, t-1) < \frac{r(d+2)}{\epsilon}$$

and since $K_t \subseteq B(\mathbf{y}, r)$,

$$\mathrm{C}_{\mathrm{NS}}(t, t) = \|\textsc{Rob}_t - \textsc{Adv}_{t-1}\|_{\ell^2} \leq 2r$$

and finally

$$\mathrm{C}_{\mathrm{NS}}(t+1, T) = \mathrm{C}_{\textsc{Rob}}(t+1, T) \leq rd.$$

Thus in sum,

$$\mathrm{C}_{\mathrm{NS}} \leq \frac{r(d+2)}{\epsilon} + rd + 2r = \left(1 + \frac{1}{\epsilon}\right) r(d+2).$$

This gives both the robustness and consistency bounds, since $\epsilon \cdot \mathrm{C}_{\textsc{Adv}} \geq r(d+2)$ and $\mathrm{C}_{\textsc{Opt}} \geq 1$. $\square$

2.C Proofs for the One-Dimensional Setting

Proof of Theorem 2.4.1

Consistency bound. We begin by proving the competitive bound with respect to $\textsc{Adv}$—i.e., consistency—via a potential function argument. Define the potential function $\phi_t = |x_t - \tilde{x}_t|$ and denote $\Delta\phi_t = \phi_t - \phi_{t-1}$. For each time $t \in [T]$, there are four different possible cases: (1) the algorithm already began at the minimizer v_t, (2) the algorithm moves all the way to the advice $\tilde{x}_t$, which is equivalent to the condition that $\tilde{x}_t$ is contained in the interval between $x(\underline{\lambda})$ and $x(\overline{\lambda})$ by line 7 of the algorithm, (3) the algorithm moves "past" the advice, and (4) the algorithm moves toward but does not reach the advice.

(1) Suppose that $x_{t-1} = v_t$. Then $x_t = v_t$, so there is no movement cost, and $f_t(x_t) = f_t(v_t) \leq f_t(\tilde{x}_t)$. As such,

$$\begin{aligned}\Delta\phi_t &= |v_t - \tilde{x}_t| - |v_t - \tilde{x}_{t-1}| \\ &\leq |\tilde{x}_t - \tilde{x}_{t-1}| \\ &\leq f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}| - f_t(x_t),\end{aligned}$$

which, rearranging, implies

$$f_t(x_t) + |x_t - x_{t-1}| + \Delta\phi_t \le f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}|. \tag{2.13}$$

(2) Suppose that $x_t = \tilde{x}_t$, i.e., AOBD moves directly to the advice. Thus $f_t(x_t) = f_t(\tilde{x}_t)$, and

$$\begin{aligned}\Delta\phi_t &= -|x_{t-1} - \tilde{x}_{t-1}| \\ &\le |\tilde{x}_t - \tilde{x}_{t-1}| - |x_t - x_{t-1}| \\ &= f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}| - f_t(x_t) - |x_t - x_{t-1}|,\end{aligned}$$

which, rearranging, implies

$$f_t(x_t) + |x_t - x_{t-1}| + \Delta\phi_t \le f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}|. \tag{2.14}$$

(3) Suppose that $x_t \ne \tilde{x}_t$ lies between $\tilde{x}_t$ and v_t, and that x_{t-1} is on the same side of v_t as $\tilde{x}_t$; these assumptions encode the condition that the algorithm spends at least part of its turn moving "past" the advice toward the minimizer, i.e., moving from x_{t-1} to x_t requires some movement away from $\tilde{x}_t$. Without loss of generality, we can assume that $\tilde{x}_t < x_t \le v_t$ and $x_{t-1} < v_t$. Because v_t is a minimizer of f_t, f_t is nonincreasing on the interval $[x_{t-1}, v_t]$, so it must be the case that $x(\underline{\lambda}) \le x(\overline{\lambda})$. Because $\tilde{x}_t < x_t$, line 7 of the algorithm implies that $x_t = x(\underline{\lambda})$, so $|x_t - x_{t-1}| \le \underline{\beta} f_t(x_t)$.[4] As a result,

$$\begin{aligned}\Delta\phi_t &= |x_t - \tilde{x}_t| - |x_{t-1} - \tilde{x}_{t-1}| \\ &\le |\tilde{x}_t - \tilde{x}_{t-1}| + |x_t - x_{t-1}| \\ &\le f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}| - f_t(x_t) + |x_t - x_{t-1}| && \text{since } f_t(x_t) \le f_t(\tilde{x}_t) \\ &\le f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}| - (1 - \underline{\beta}) f_t(x_t) \\ &\le f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}| - \frac{1 - \underline{\beta}}{1 + \underline{\beta}} \left(f_t(x_t) + |x_t - x_{t-1}| \right),\end{aligned}$$

which, rearranging, implies

$$f_t(x_t) + |x_t - x_{t-1}| + \Delta\phi_t \le \frac{1 + \underline{\beta}}{1 - \underline{\beta}} \left(f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}| \right). \tag{2.15}$$

[4] If the balancing equality in line 5 of the algorithm does not hold for any $\underline{\lambda} \in [0, 1]$, then this implies that $|x(\lambda) - x_{t-1}| \le \underline{\beta} f_t(x(\lambda))$ for all $\lambda \in [0, 1]$—and in particular for $\underline{\lambda} = 1$—since the movement cost term is increasing in λ and the hitting cost term is nonincreasing in λ.

(4) Assume without loss of generality that $\tilde{x}_t \leq v_t$. Suppose that x_t lies outside of the interval $[\tilde{x}_t, v_t)$, and that the ray $\overrightarrow{x_{t-1}x_t}$ contains $\tilde{x}_t$; these assumptions encode the condition that the algorithm spends its turn moving toward the advice $\tilde{x}_t$—i.e., that $|x_t - \tilde{x}_t| < |x_{t-1} - \tilde{x}_t|$—and that it does not reach $\tilde{x}_t$. We break into two further cases:

(a) Suppose that $x_t = v_t$. Clearly $f_t(x_t) \leq f_t(\tilde{x}_t)$. Then

$$\begin{aligned}\Delta\phi_t &= |x_t - \tilde{x}_t| - |x_{t-1} - \tilde{x}_{t-1}| \\ &= |\tilde{x}_t - x_{t-1}| - |x_t - x_{t-1}| - |x_{t-1} - \tilde{x}_{t-1}| \qquad (2.16) \\ &\leq |\tilde{x}_t - \tilde{x}_{t-1}| - |x_t - x_{t-1}| \\ &\leq f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}| - f_t(x_t) - |x_t - x_{t-1}|,\end{aligned}$$

where (2.16) follows from the assumption that x_t is closer to $\tilde{x}_t$ than x_{t-1} is. Rearranging, we have

$$f_t(x_t) + |x_t - x_{t-1}| + \Delta\phi_t \leq f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}|. \qquad (2.17)$$

(b) Suppose that $x_t \neq v_t$; this implies by the definition of $x(\lambda)$ and line 6 of the algorithm that $\overline{\lambda} < 1$. Moreover, because increasing λ moves $x(\lambda)$ toward $\tilde{x}_t$, yet $x_t \neq \tilde{x}_t$, it must be the case that $x_t = x(\overline{\lambda})$, and since $\overline{\lambda} < 1$, we have the equality $|x_t - x_{t-1}| = \overline{\beta} f_t(x_t)$. As a result, Thus

$$\begin{aligned}\Delta\phi_t &= |x_t - \tilde{x}_t| - |x_{t-1} - \tilde{x}_{t-1}| \\ &= |\tilde{x}_t - x_{t-1}| - |x_t - x_{t-1}| - |x_{t-1} - \tilde{x}_{t-1}| \\ &\leq |\tilde{x}_t - \tilde{x}_{t-1}| - |x_t - x_{t-1}| \\ &\leq f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}| - \frac{\overline{\beta}}{1+\overline{\beta}}\left(f_t(x_t) + |x_t - x_{t-1}|\right),\end{aligned}$$

which, rearranging, implies

$$f_t(x_t) + |x_t - x_{t-1}| + \Delta\phi_t \leq \frac{1+\overline{\beta}}{\overline{\beta}}\left(f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}|\right). \qquad (2.18)$$

Summing the various cases (2.13), (2.14), (2.15), (2.17), and (2.18) over time, we obtain the claimed consistency bound:

$$\sum_{t=1}^{T} f_t(x_t) + |x_t - x_{t-1}| \leq \max\left\{\frac{1+\underline{\beta}}{1-\underline{\beta}}, \frac{1+\overline{\beta}}{\overline{\beta}}\right\} \sum_{t=1}^{T} f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}|.$$

Robustness bound. We now move to proving the robustness of AOBD, once again using a potential function argument. Let $o_1, \ldots, o_T \in \mathbb{R}$ denote the decisions of the offline optimal algorithm, and define the potential function $\phi_t = |x_t - o_t|$, with $\Delta\phi_t = \phi_t - \phi_{t-1}$. We break into two different cases.

(1) Suppose that $f_t(x_t) \leq f_t(o_t)$. Note that, by construction of the algorithm, $|x_t - x_{t-1}| \leq \overline{\beta} f_t(x_t)$. Let $c > \overline{\beta} - 1$, and observe that

$$\begin{aligned}
\Delta\phi_t &= |x_t - o_t| - |x_{t-1} - o_{t-1}| \\
&\leq |o_t - o_{t-1}| + |x_t - x_{t-1}| \\
&\leq (1+c)\left(f_t(o_t) + |o_t - o_{t-1}|\right) - (1+c) f_t(x_t) + |x_t - x_{t-1}| \\
&\leq (1+c)\left(f_t(o_t) + |o_t - o_{t-1}|\right) - (1+c-\overline{\beta}) f_t(x_t) \\
&\leq (1+c)\left(f_t(o_t) + |o_t - o_{t-1}|\right) - \frac{1+c-\overline{\beta}}{1+\overline{\beta}}\left(f_t(x_t) + |x_t - x_{t-1}|\right),
\end{aligned}$$

which, rearranging, gives

$$f_t(x_t) + |x_t - x_{t-1}| + \frac{1+\overline{\beta}}{1+c-\overline{\beta}}\Delta\phi_t \leq \frac{(1+\overline{\beta})(1+c)}{1+c-\overline{\beta}}\left(f_t(o_t) + |o_t - o_{t-1}|\right).$$

Selecting $c = 2\overline{\beta}$, we obtain

$$f_t(x_t) + |x_t - x_{t-1}| + \Delta\phi_t \leq (1+2\overline{\beta})\left(f_t(o_t) + |o_t - o_{t-1}|\right). \tag{2.19}$$

(2) Suppose that $f_t(x_t) > f_t(o_t)$; this implies that $x_t \neq v_t$, so in particular, the balance condition holds for some $\beta \in [\underline{\beta}, \overline{\beta}]$: $|x_t - x_{t-1}| = \beta f_t(x_t)$. In addition, $f_t(x_t) > f_t(o_t)$ implies that AOBD must have moved closer to o_t during its turn. Thus

$$\begin{aligned}
\Delta\phi_t &= |x_t - o_t| - |x_{t-1} - o_{t-1}| \\
&= |o_t - x_{t-1}| - |x_t - x_{t-1}| - |x_{t-1} - o_{t-1}| \\
&\leq |o_t - o_{t-1}| - |x_t - x_{t-1}| \\
&= |o_t - o_{t-1}| - \frac{\beta}{1+\beta}\left(f_t(x_t) + |x_t - x_{t-1}|\right) \\
&\leq f_t(o_t) + |o_t - o_{t-1}| - \frac{\underline{\beta}}{1+\underline{\beta}}\left(f_t(x_t) + |x_t - x_{t-1}|\right),
\end{aligned}$$

which, rearranging, yields

$$f_t(x_t) + |x_t - x_{t-1}| + \Delta\phi_t \leq \frac{1+\underline{\beta}}{\underline{\beta}}\left(f_t(o_t) + |o_t - o_{t-1}|\right). \tag{2.20}$$

Summing the cases (2.19) and (2.20) over time, we obtain the claimed robustness bound:

$$\sum_{t=1}^{T} f_t(x_t) + |x_t - x_{t-1}| \leq \max\left\{1 + 2\overline{\beta}, 1 + \underline{\beta}^{-1}\right\} \sum_{t=1}^{T} f_t(\tilde{x}_t) + |\tilde{x}_t - \tilde{x}_{t-1}|.$$

□

Proof of Theorem 2.4.2

Let x_t denote the decisions of ALG, and $\tilde{x}_t$ the decisions of ADV. Consider a problem instance where $\tilde{x}_0 = x_0 = 0$, $f_1(x) = 2\epsilon|x-1|$, $\tilde{x}_1 = 1$, and the time horizon is $T = 2$. We distinguish two cases:

(1) Suppose that ALG chooses $x_1 \geq \frac{1}{2}$. Let $f_2(x) = |x|$, with advice $\tilde{x}_2 = 0$. Then the optimal solution stays at 0 for the entire instance, incurring a cost of 2ϵ. On the other hand, ALG incurs a cost of at least 1, so the competitive ratio is at least $\frac{1}{2\epsilon}$.

(2) Suppose that ALG chooses $x_1 < \frac{1}{2}$. Let $f_2(x) = |x-1|$, with advice $\tilde{x}_2 = 1$. Then ADV incurs total cost 1, while ALG incurs a cost of $1 + 2\epsilon(1 - x_1) > 1 + \epsilon$; thus, ALG is not $(1+\epsilon)$-consistent.

□

2.D Background from the Geometry of Normed Vector Spaces

In this appendix section, we introduce some notions and results from the literature on the geometry of normed vector spaces, expanding on the brief definitions of the rectangular constant and the radial retraction given in the main text in Section 9. In the following definitions and results, $\mathcal{X} = (X, \|\cdot\|)$ is an arbitrary real normed vector space.

We begin by defining Birkhoff-James orthogonality, which generalizes the usual Hilbert space orthogonality.

Definition 2.D.1 ([126, p.169]; [127, p.265]). *$\mathbf{x} \in X$ is **Birkhoff-James orthogonal** to $\mathbf{y} \in X$, denoted $\mathbf{x} \perp \mathbf{y}$, if $\|\mathbf{x}\| \leq \|\mathbf{x} + \lambda\mathbf{y}\|$ for all $\lambda \in \mathbb{R}$.*

Note that, unlike orthogonality in Hilbert spaces, Birkhoff-James orthogonality is not generally symmetric. However, it is homogeneous.

Lemma 2.D.2 ([127, p.265]; [121, Remark 1]). *If $\mathbf{x} \perp \mathbf{y}$, then $a\mathbf{x} \perp b\mathbf{y}$ for all $a, b \in \mathbb{R}$.*

Using Birkhoff-James orthogonality, we can formally define define the first constant we introduced in Section 9: the rectangular constant. It is motivated by the following observation: in a finite-dimensional inner product space, orthogonality of $\mathbf{x}$ and $\mathbf{y}$ implies that $\frac{\|\mathbf{x}\|+\|\mathbf{y}\|}{\|\mathbf{x}+\mathbf{y}\|} \leq \sqrt{2}$. In an arbitrary normed vector space, the upper bound $\sqrt{2}$ is replaced with the *rectangular constant*, defined as follows using Birkhoff-James orthogonality.

Definition 2.D.3 ([128, Definition 2]; original from [121, Definition 2])**.** *The **rectangular constant** $\mu(\mathcal{X})$ of a real normed vector space $\mathcal{X}$ is defined as*

$$\mu(\mathcal{X}) = \sup_{\mathbf{x} \perp \mathbf{y}} \frac{\|\mathbf{x}\| + \|\mathbf{y}\|}{\|\mathbf{x} + \mathbf{y}\|}.$$

It is known that $\sqrt{2} \leq \mu(\mathcal{X}) \leq 3$ [121, Section II], and these bounds are tight: $\mu(\mathcal{X}) = \sqrt{2}$ for any Hilbert space [121, Example 1, Section III], and $\mu(\mathcal{X}) = 3$ for "nonuniformly nonsquare" spaces such as ℓ^1 and ℓ^∞ ([122]). Moreover, $\mu(\ell^p) < 3$ for all $p \in (1, \infty)$. In fact, tighter bounds are known for the ℓ^p spaces: we review these in the following theorem.

Theorem 2.D.4 ([122, Theorems 5.2, 5.4, 5.5])**.** *For $1 < p \leq 2$,*

$$\mu(\ell^p) \leq \min\left\{ \left(1 + \left(2^{1/(p-1)} - 1\right)^{p-1}\right)^{1/p}, \sqrt{\frac{p}{p-1}} \right\}.$$

For $p \geq 2$,

$$\mu(\ell^p) \leq \left(1 + \left(2^{p-1} - 1\right)^{1/(p-1)}\right)^{(p-1)/p}.$$

Together, these constitute an upper bound on $\mu(\ell^p)$ that attains a (tight) minimum of $\sqrt{2}$ at $p = 2$, and that continuously increases toward 3 as $p \to \infty$ and $p \to 1$.

We now reiterate the definition of the radial retraction and its Lipschitz constant given in Section 9.

Definition 2.D.5 ([123])**.** *On a normed vector space $\mathcal{X} = (X, \|\cdot\|)$, the radial retraction $\rho(\cdot\,; r) : X \to B(\mathbf{0}, r)$ is the metric projection onto the closed ball of radius $r \geq 0$:*

$$\rho(\mathbf{x}; r) = \begin{cases} \mathbf{x} & \textit{if } \|\mathbf{x}\| \leq r \\ r\frac{\mathbf{x}}{\|\mathbf{x}\|} & \textit{if } \|\mathbf{x}\| > r. \end{cases}$$

We define $k(\mathcal{X})$ to be the Lipschitz constant of $\rho(\cdot\,; 1)$, i.e., the smallest real number satisfying

$$\|\rho(\mathbf{x}; 1) - \rho(\mathbf{y}; 1)\| \leq k(\mathcal{X})\|\mathbf{x} - \mathbf{y}\|$$

for all $\mathbf{x}, \mathbf{y} \in X$.

It holds that $k(\mathcal{X})$ is bounded between 1 and 2 in any normed vector space $\mathcal{X}$ ([124]). Moreover, $k(\mathcal{X})$ is identically the Lipschitz constant of $\rho(\cdot\,; r)$ for any $r > 0$ ([123]). To see that this is the case, observe that $\rho(\mathbf{x}; r) = r \cdot \rho(\frac{\mathbf{x}}{r}; 1)$; it then follows that

$$\|\rho(\mathbf{x}; r) - \rho(\mathbf{y}; r)\| = r\left\|\rho\left(\frac{\mathbf{x}}{r}; 1\right) - \rho\left(\frac{\mathbf{y}}{r}; 1\right)\right\| \leq k(\mathcal{X})\|\mathbf{x} - \mathbf{y}\|.$$

Thus $k(\mathcal{X})$ is an upper bound on the Lipschitz constant of $\rho(\mathbf{x}; r)$ for general $r > 0$. Similar reasoning shows that $\rho(\mathbf{x}; r)$ can have no smaller Lipschitz constant than $k(\mathcal{X})$; so $k(\mathcal{X})$ is the Lipschitz constant for all $\rho(\mathbf{x}; r)$, $r > 0$.

We conclude this section with a result relating $k(\mathcal{X})$ with $\mu(\mathcal{X})$.

Proposition 2.D.6. *On a real normed vector space $\mathcal{X} = (X, \|\cdot\|)$, it holds that $k(\mathcal{X}) \leq \mu(\mathcal{X})$.*

Proof. [124, Theorem 1] characterizes $k(\mathcal{X})$ as follows:

$$k(\mathcal{X}) = \sup_{\mathbf{x} \perp \mathbf{y}, \mathbf{y} \neq \mathbf{0}, \lambda \in \mathbb{R}} \frac{\|\mathbf{y}\|}{\|\mathbf{y} - \lambda\mathbf{x}\|}. \tag{2.21}$$

Since, by Lemma 2.D.2, Birkhoff-James orthogonality is homogeneous, it is straightforward to see that (2.21) can be equivalently expressed as

$$k(\mathcal{X}) = \sup_{\mathbf{x} \perp \mathbf{y}, \mathbf{y} \neq \mathbf{0}} \frac{\|\mathbf{y}\|}{\|\mathbf{x} + \mathbf{y}\|}.$$

Then it is clear that

$$k(\mathcal{X}) = \sup_{\mathbf{x} \perp \mathbf{y}, \mathbf{y} \neq \mathbf{0}} \frac{\|\mathbf{y}\|}{\|\mathbf{x} + \mathbf{y}\|} \leq \sup_{\mathbf{x} \perp \mathbf{y}} \frac{\|\mathbf{x}\| + \|\mathbf{y}\|}{\|\mathbf{x} + \mathbf{y}\|} = \mu(\mathcal{X}).$$

□

2.E Proof of Theorem 2.4.4

Geometric lemmas

Before presenting the analysis of Algorithm 4, we take a brief foray into the geometric theory of normed vector spaces, presenting and proving some lemmas that will be helpful in proving the bicompetitive bound given in Theorem 2.4.4. The results in this section depend heavily on the definitions and results introduced in Section 2.D.

The first lemma characterizes (a modified form of) the radial retraction as a metric projection onto the boundary of a closed ball.

Lemma 2.E.1. *Let* $(X, \|\cdot\|)$ *be a normed vector space, and consider arbitrary* $r \geq 0$, $\mathbf{x} \in X$, *and* $\mathbf{y} \in X \setminus \{\mathbf{x}\}$. *Define* $\hat{\mathbf{y}} = \mathbf{x} + r\frac{\mathbf{y}-\mathbf{x}}{\|\mathbf{y}-\mathbf{x}\|}$. *Then* $\|\mathbf{y} - \hat{\mathbf{y}}\| \leq \|\mathbf{y} - \mathbf{w}\|$ *for all* $\mathbf{w} \in \partial B(\mathbf{x}, r)$.

Proof. It suffices to consider the case when $\mathbf{x} = \mathbf{0}$. If $r = 0$, $\partial B(\mathbf{0}, r) = \{\mathbf{0}\}$, so the result is clear. Otherwise, fix arbitrary $\mathbf{w} \in \partial B(\mathbf{0}, r)$ and observe

$$\begin{aligned} \|\mathbf{y} - \mathbf{w}\| &\geq |\|\mathbf{y}\| - \|\mathbf{w}\|| && \text{by the triangle inequality} \\ &= |\|\mathbf{y}\| - r| \\ &= |\|\mathbf{y}\| - \|\hat{\mathbf{y}}\|| \\ &= \|\mathbf{y} - \hat{\mathbf{y}}\| \end{aligned}$$

where the last step follows from collinearity of $\mathbf{y}, \hat{\mathbf{y}}$, and $\mathbf{0}$. □

The second lemma generalizes the following geometric fact in the Euclidean plane to an arbitrary normed vector spaces: given a triangle **abc** in $(\mathbb{R}^2, \|\cdot\|_{\ell^2})$, and points $\mathbf{x} \in [\mathbf{a}, \mathbf{b}], \mathbf{y} \in [\mathbf{a}, \mathbf{c}]$ with $\|\mathbf{x} - \mathbf{b}\|_{\ell^2} = \|\mathbf{y} - \mathbf{c}\|_{\ell^2}$, it holds that $\|\mathbf{x} - \mathbf{y}\|_{\ell^2} \leq \|\mathbf{b} - \mathbf{c}\|_{\ell^2}$. In the general setting, this becomes a statement about the distance between the radial retractions of a single point onto two balls of the same radius with different centers.

Lemma 2.E.2. *Let* $(X, \|\cdot\|)$ *be a normed vector space, and fix arbitrary* $\mathbf{a}, \mathbf{b}, \mathbf{c} \in X$ *and* $r \geq 0$. *Define* $\mathbf{x} = \mathbf{b} + \rho(\mathbf{a} - \mathbf{b}; r)$ *and* $\mathbf{y} = \mathbf{c} + \rho(\mathbf{a} - \mathbf{c}; r)$. *Then* $\|\mathbf{x} - \mathbf{y}\| \leq \|\mathbf{b} - \mathbf{c}\|$.

Proof. We may assume without loss of generality that $\|\mathbf{a} - \mathbf{b}\| \geq \|\mathbf{a} - \mathbf{c}\|$. If $\mathbf{b} = \mathbf{c}$, $\|\mathbf{x} - \mathbf{y}\| = 0 = \|\mathbf{b} - \mathbf{c}\|$. Thus we restrict to the case where $\mathbf{b} \neq \mathbf{c}$ and distinguish cases based on the value of r. We may further restrict to the case where $\|\mathbf{a} - \mathbf{c}\| > 0$, as the case $\mathbf{a} = \mathbf{c}$ is trivial.

If $r = 0$, then $\mathbf{x} = \mathbf{b}$ and $\mathbf{y} = \mathbf{c}$. Thus $\|\mathbf{x} - \mathbf{y}\| = \|\mathbf{b} - \mathbf{c}\|$. On the other hand, if $r \geq \|\mathbf{a} - \mathbf{b}\|$, then $r \geq \|\mathbf{a} - \mathbf{c}\|$ as well, so $\mathbf{x} = \mathbf{y} = \mathbf{a}$, and certainly $\|\mathbf{x} - \mathbf{y}\| = 0 \leq \|\mathbf{b} - \mathbf{c}\|$.

Next, suppose $\|\mathbf{a}-\mathbf{c}\| \le r < \|\mathbf{a}-\mathbf{b}\|$. Then $\mathbf{y}=\mathbf{a}$, so $\|\mathbf{x}-\mathbf{y}\| = \|\mathbf{x}-\mathbf{a}\|$. Moreover, $\|\mathbf{x}-\mathbf{b}\| = r \ge \|\mathbf{a}-\mathbf{c}\|$. Then by the triangle inequality,

$$\begin{aligned}\|\mathbf{b}-\mathbf{c}\| &\ge |\|\mathbf{a}-\mathbf{b}\| - \|\mathbf{a}-\mathbf{c}\|| \\ &= |\|\mathbf{a}-\mathbf{x}\| + \|\mathbf{x}-\mathbf{b}\| - \|\mathbf{a}-\mathbf{c}\|| \\ &\ge \|\mathbf{a}-\mathbf{x}\| \\ &= \|\mathbf{x}-\mathbf{y}\|.\end{aligned}$$

Finally, suppose $0 < r < \|\mathbf{a}-\mathbf{c}\|$, and define $\lambda = 1-\frac{r}{\|\mathbf{a}-\mathbf{c}\|}$. Since $\mathbf{y}+\mathbf{b}-\mathbf{c} = \mathbf{b}+r\frac{\mathbf{a}-\mathbf{c}}{\|\mathbf{a}-\mathbf{c}\|}$, we know that $\mathbf{y}+\mathbf{b}-\mathbf{c} \in \partial B(\mathbf{b}, r)$. Observe moreover that

$$\mathbf{z} := \mathbf{y} + \lambda(\mathbf{b}-\mathbf{c}) = \mathbf{b} + r\frac{\mathbf{a}-\mathbf{b}}{\|\mathbf{a}-\mathbf{c}\|} \in [\mathbf{a}, \mathbf{b}]$$

and $\mathbf{z} \neq \mathbf{b}$ by assumption that $r > 0$. Thus $\frac{\mathbf{z}-\mathbf{b}}{\|\mathbf{z}-\mathbf{b}\|} = \frac{\mathbf{a}-\mathbf{b}}{\|\mathbf{a}-\mathbf{b}\|}$, so $\mathbf{x} = \mathbf{b} + r\frac{\mathbf{z}-\mathbf{b}}{\|\mathbf{z}-\mathbf{b}\|}$. Thus:

$$\begin{aligned}\|\mathbf{b}-\mathbf{c}\| &= \|(\mathbf{y}+\mathbf{b}-\mathbf{c})-\mathbf{y}\| \\ &= \|(\mathbf{y}+\mathbf{b}-\mathbf{c})-\mathbf{z}\| + \|\mathbf{z}-\mathbf{y}\| && \text{by collinearity of } \mathbf{y}, \mathbf{z}, \mathbf{y}+\mathbf{b}-\mathbf{c} \\ &\ge \|\mathbf{x}-\mathbf{z}\| + \|\mathbf{z}-\mathbf{y}\| && \text{applying Lemma 2.E.1} \qquad (2.22) \\ &\ge \|\mathbf{x}-\mathbf{y}\| && \text{by triangle inequality,}\end{aligned}$$

where, in (2.22), $\mathbf{x}, \mathbf{y}, \hat{\mathbf{y}}, \mathbf{w}$, and r in Lemma 2.E.1 are instantiated during its invocation with this proof's $\mathbf{b}, \mathbf{z}, \mathbf{x}, (\mathbf{y}+\mathbf{b}-\mathbf{c})$, and r, respectively. □

The next geometric lemma provides a bound on the total distance traveled first between two points on a sphere, and then from the second point to a scaled version thereof, in terms of the rectangular constant and the distance between the initial and final points.

Lemma 2.E.3. *Let $(X, \|\cdot\|)$ be a normed vector space, and let $t > 1$, $r > 0$, and $\mathbf{x}, \mathbf{y} \in \partial B(\mathbf{0}, r)$. Then*

$$\|\mathbf{y}-\mathbf{x}\| + (t-1)\|\mathbf{y}\| \le \mu(\mathcal{X})\|t\mathbf{y}-\mathbf{x}\|.$$

Proof. By a corollary of the Hahn-Banach theorem [129, Chapter 3, Corollary 7], there exists a *support functional* $f \in X^*$ at $\mathbf{y}$, i.e., some bounded linear functional $f : X \to \mathbb{R}$, with $\|f\|_{X^*} = 1$, $f(\mathbf{y}) = \|\mathbf{y}\| = r$, and the property that the hyperplane $H(r) := \{\mathbf{z} \in X : f(\mathbf{z}) = r\}$ contains no points in $\mathrm{int}(B(\mathbf{0}, r))$. Note that we can equivalently write $H(r)$ in affine subspace form $H(r) = \mathbf{y} + \ker(f) = \{\mathbf{z} \in X : \mathbf{z} =$

$\mathbf{y} + \mathbf{h}, \mathbf{h} \in \ker(f)\}$, by linearity of f. The fact that $H(r)$ contains no points in the interior of $B(\mathbf{0}, r)$ means that $\mathbf{y} \perp \mathbf{h}$ for all $\mathbf{h} \in \ker(f)$.

Define $s = f(\mathbf{x})$, and note that since $\|\mathbf{x}\| = r$ and $\|f\|_{X^*} = 1$, we must have $s \leq r$. Then define $\mathbf{z} = \frac{s}{r}\mathbf{y}$, and observe $H(s) = \mathbf{x} + \ker(f) = \mathbf{z} + \ker(f)$. Thus $\mathbf{x} = \mathbf{z} + \mathbf{h}$ for some specific $\mathbf{h} \in \ker(f)$. By homogeneity of Birkhoff-James orthogonality (Lemma 2.D.2), it follows that $(t - \frac{s}{r})\mathbf{y} \perp -\mathbf{h}$. As such,

$$\begin{aligned}
\|\mathbf{y} - \mathbf{x}\| + (t-1)\|\mathbf{y}\| &\leq \|\mathbf{y} - \mathbf{z}\| + \|\mathbf{x} - \mathbf{z}\| + (t-1)\|\mathbf{y}\| \\
&= \left(1 - \frac{s}{r}\right)\|\mathbf{y}\| + \|\mathbf{h}\| + (t-1)\|\mathbf{y}\| \\
&= \|-\mathbf{h}\| + \left(t - \frac{s}{r}\right)\|\mathbf{y}\| \\
&\leq \mu(\mathcal{X})\left\|-\mathbf{h} + \left(t - \frac{s}{r}\right)\mathbf{y}\right\| \\
&= \mu(\mathcal{X})\|t\mathbf{y} - \mathbf{x}\|.
\end{aligned}$$

□

Finally, we present a lemma building upon Lemma 2.E.3 that will be indispensable for the consistency analysis of Algorithm 4.

Lemma 2.E.4. *Let $(X, \|\cdot\|)$ be a normed vector space, and fix arbitrary $r \geq 0$, $\mathbf{w}, \mathbf{y} \in X$, and $\mathbf{x} \in X \setminus \mathrm{int}(B(\mathbf{w}, r))$. Define $\hat{\mathbf{x}} = \mathbf{w} + \rho(\mathbf{x} - \mathbf{w}; r)$ and $\hat{\mathbf{y}} = \mathbf{w} + \rho(\mathbf{y} - \mathbf{w}; r)$. Then*

$$\|\hat{\mathbf{y}} - \hat{\mathbf{x}}\| + \|\mathbf{y} - \hat{\mathbf{y}}\| \leq \mu(\mathcal{X})\|\mathbf{y} - \mathbf{x}\| + \|\mathbf{x} - \hat{\mathbf{x}}\|.$$

Proof. If $r = 0$, then $B(\mathbf{w}, r) = \{\mathbf{w}\}$, so $\hat{\mathbf{x}} = \hat{\mathbf{y}} = \mathbf{w}$, and the result follows from the triangle inequality, as $\mu(\mathcal{X}) \geq \sqrt{2}$. Thus we restrict to the case that $r > 0$.

It suffices to consider the case where $\mathbf{w} = \mathbf{0}$. Then $\hat{\mathbf{x}} = \rho(\mathbf{x}; r)$ and $\hat{\mathbf{y}} = \rho(\mathbf{y}; r)$. We distinguish two cases.

First, suppose $\mathbf{y} \in B(\mathbf{0}, r)$. Then $\hat{\mathbf{y}} = \mathbf{y}$, and by the triangle inequality,

$$\|\hat{\mathbf{y}} - \hat{\mathbf{x}}\| + \|\mathbf{y} - \hat{\mathbf{y}}\| = \|\mathbf{y} - \hat{\mathbf{x}}\| \leq \|\mathbf{y} - \mathbf{x}\| + \|\mathbf{x} - \hat{\mathbf{x}}\| \leq \mu(\mathcal{X})\|\mathbf{y} - \mathbf{x}\| + \|\mathbf{x} - \hat{\mathbf{x}}\|.$$

Second, suppose $\mathbf{y} \in X \setminus B(\mathbf{0}, r)$. Then $\hat{\mathbf{y}} = r\frac{\mathbf{y}}{\|\mathbf{y}\|}$. We distinguish two further subcases.

(i) Suppose that $\|\mathbf{y} - \hat{\mathbf{y}}\| \leq \|\mathbf{x} - \hat{\mathbf{x}}\|$. Then

$$\begin{aligned}\|\hat{\mathbf{y}} - \hat{\mathbf{x}}\| + \|\mathbf{y} - \hat{\mathbf{y}}\| &\leq \|\hat{\mathbf{y}} - \hat{\mathbf{x}}\| + \|\mathbf{x} - \hat{\mathbf{x}}\| \\ &\leq k(\mathcal{X})\|\mathbf{y} - \mathbf{x}\| + \|\mathbf{x} - \hat{\mathbf{x}}\| \\ &\leq \mu(\mathcal{X})\|\mathbf{y} - \mathbf{x}\| + \|\mathbf{x} - \hat{\mathbf{x}}\| \qquad \text{by Proposition 2.D.6.}\end{aligned}$$

(ii) On the other hand, suppose that $\|\mathbf{y} - \hat{\mathbf{y}}\| > \|\mathbf{x} - \hat{\mathbf{x}}\|$, or equivalently, $\|\mathbf{y}\| \geq \|\mathbf{x}\|$, and define $\mathbf{z} = \rho(\mathbf{y}; \|\mathbf{x}\|) = \|\mathbf{x}\|\frac{\mathbf{y}}{\|\mathbf{y}\|}$. By collinearity, $\|\mathbf{z} - \hat{\mathbf{y}}\| = \|\mathbf{x}\| - r = \|\mathbf{x} - \hat{\mathbf{x}}\|$. Furthermore, we have that

$$\|\hat{\mathbf{y}} - \hat{\mathbf{x}}\| \leq \frac{\|\mathbf{x}\|}{r}\|\hat{\mathbf{y}} - \hat{\mathbf{x}}\| = \|\mathbf{z} - \mathbf{x}\|.$$

It then follows that

$$\begin{aligned}\|\hat{\mathbf{y}} - \hat{\mathbf{x}}\| + \|\mathbf{y} - \hat{\mathbf{y}}\| &\leq \|\mathbf{z} - \mathbf{x}\| + \|\mathbf{y} - \mathbf{z}\| + \|\mathbf{z} - \hat{\mathbf{y}}\| \\ &= \|\mathbf{z} - \mathbf{x}\| + \left(\frac{\|\mathbf{y}\|}{\|\mathbf{x}\|} - 1\right)\|\mathbf{z}\| + \|\mathbf{x} - \hat{\mathbf{x}}\| \\ &\leq \mu(\mathcal{X})\|\mathbf{y} - \mathbf{x}\| + \|\mathbf{x} - \hat{\mathbf{x}}\|\end{aligned}$$

where the final inequality follows from Lemma 2.E.3, where $\mathbf{x}, \mathbf{z} \in \partial B(\mathbf{0}, \|\mathbf{x}\|)$ are (respectively) the points $\mathbf{x}, \mathbf{y}$ in that lemma's statement.

□

We have now presented all technical lemmas that will be employed in our proof of Theorem 2.4.4. Before moving on to this proof in the next section, we first provide several immediate corollaries of the preceding lemmas characterizing various steps of Algorithm 4.

Corollary 2.E.5. *In the specification of* Interp *(Algorithm 4), if x_t is determined by Line 8, then* $\|\mathbf{x}_t - \mathbf{z}_t\| \leq \|\mathbf{s}_t - \mathbf{s}_{t-1}\|$.

Proof. This follows immediately from Lemma 2.E.2 with the lemma's **a**, **b**, **c**, and r respectively chosen as $\tilde{\mathbf{x}}_t$, $\mathbf{s}_{t-1}$, $\mathbf{s}_t$ and $\|\mathbf{z}_t - \mathbf{s}_{t-1}\|$. □

Corollary 2.E.6. *In the specification of* Interp *(Algorithm 4),*

$$\|\mathbf{y}_t - \mathbf{x}_{t-1}\| \leq k(\mathcal{X})\|\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}\|.$$

Proof. This follows by definition of the Lipschitz constant $k(\mathcal{X})$ of the radial retraction, and the observation that $\mathbf{y}_t$ (respectively $\mathbf{x}_{t-1}$) is the radial retraction of $\tilde{\mathbf{x}}_t$ (respectively $\tilde{\mathbf{x}}_{t-1}$) onto the ball $B(\mathbf{s}_{t-1}, \|\mathbf{x}_{t-1} - \mathbf{s}_{t-1}\|)$. □

Corollary 2.E.7. *In the specification of* Interp *(Algorithm 4),*

$$\|\mathbf{y}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_t - \mathbf{y}_t\| \leq \mu(\mathcal{X})\|\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}\| + \|\tilde{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}\|.$$

Proof. This follows immediately from Lemma 2.E.4 with $\mathbf{x}, \mathbf{y}, \mathbf{w}$, and r in the lemma's statement chosen respectively as $\tilde{\mathbf{x}}_{t-1}$, $\tilde{\mathbf{x}}_t$, $\mathbf{s}_{t-1}$, and $\|\mathbf{x}_{t-1} - \mathbf{s}_{t-1}\|$, which in turn yields $\hat{\mathbf{y}} = \mathbf{y}_t$ and $\hat{\mathbf{x}} = \mathbf{x}_{t-1}$. □

Proof of bicompetitive bound

We prove the bicompetitive bound of Theorem 2.4.4 in two parts: we will first show the competitive ratio with respect to Adv, and will follow with the competitive ratio with respect to Rob. Both results proceed via potential function arguments: the first uses the potential function $\|\tilde{\mathbf{x}}_t - \mathbf{x}_t\|$, and the second uses the potential function $c\|\mathbf{x}_t - \mathbf{s}_t\|$ (with c to be defined later on). The robustness and consistency claim then follows immediately from the bicompetitive bound and the observation in Section 2.A.

Proof of competitiveness with respect to Adv. We define "phases" of the algorithm as follows: if $\mathbf{x}_t$ is determined by line 4 of the algorithm, then the advice is in the "Adv" phase. Otherwise, if $\mathbf{x}_t$ is determined by line 8, then the advice is in the "Rob" phase. We refer to the time indices in which the algorithm is in the "Rob" phase as $R_1, \ldots, R_k \in [T]$ (where $k \leq T$, and $R_1 < \cdots < R_k$ are in increasing order). If the algorithm is never in the "Rob" phase, then $\mathbf{x}_t = \tilde{\mathbf{x}}_t \ \forall t \in [T]$, and thus Interp is 1-competitive with respect to Adv. Thus we restrict to the case that there is at least one time index in which the algorithm is in the "Rob" phase. By design, for each $j \in [k]$, $\mathrm{C}_{\mathrm{Rob}}(1, R_j) \leq \delta \cdot \mathrm{C}_{\mathrm{Adv}}(1, R_j)$.

Now we break into two cases depending on the phase. First, suppose that Interp is in the "Adv" phase. This means that $\mathbf{x}_t = \tilde{\mathbf{x}}_t$. Then

$$\begin{aligned} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_t - \mathbf{x}_t\| &= f_t(\tilde{\mathbf{x}}_t) + \|\tilde{\mathbf{x}}_t - \mathbf{x}_{t-1}\| \\ &\leq f_t(\tilde{\mathbf{x}}_t) + \|\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}\| + \|\tilde{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}\| \end{aligned} \tag{2.23}$$

follows immediately from the triangle inequality.

Second, consider the case that the algorithm is in the "Rob" phase. This means that $\mathbf{x}_t$ is determined by line 8 of the algorithm; and there exists some $\lambda \in [0,1]$ for which $\mathbf{x}_t = \lambda \mathbf{s}_t + (1-\lambda)\tilde{\mathbf{x}}_t$. In this case, observe

$$\begin{aligned}
& f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_t - \mathbf{x}_t\| \\
\leq\ & \lambda f_t(\mathbf{s}_t) + (1-\lambda) f_t(\tilde{\mathbf{x}}_t) + 2\|\mathbf{x}_t - \mathbf{z}_t\| \\
& + 2\|\mathbf{z}_t - \mathbf{y}_t\| + \|\mathbf{y}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_t - \mathbf{y}_t\| && (2.24) \\
\leq\ & 2 \cdot \mathrm{C}_{\text{Rob}}(t,t) + 2\gamma \cdot \mathrm{C}_{\text{Adv}}(t,t) + f_t(\tilde{\mathbf{x}}_t) + \|\mathbf{y}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_t - \mathbf{y}_t\| && (2.25)
\end{aligned}$$

where (2.24) follows from convexity of f_t and the triangle inequality, and (2.25) follows from bounding $\|\mathbf{x}_t - \mathbf{z}_t\|$ via Corollary 2.E.5 and $\|\mathbf{z}_t - \mathbf{y}_t\|$ via line (7) of the algorithm. Invoking Corollary 2.E.7 gives the result

$$\begin{aligned}
& f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_t - \mathbf{x}_t\| \\
\leq\ & 2 \cdot \mathrm{C}_{\text{Rob}}(t,t) + 2\gamma \cdot \mathrm{C}_{\text{Adv}}(t,t) + f_t(\tilde{\mathbf{x}}_t) + \mu(\mathcal{X})\|\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}\| + \|\tilde{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}\| \\
\leq\ & 2 \cdot \mathrm{C}_{\text{Rob}}(t,t) + (\mu(\mathcal{X}) + 2\gamma)\mathrm{C}_{\text{Adv}}(t,t) + \|\tilde{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}\|. && (2.26)
\end{aligned}$$

Summing (2.23) and (2.26) over time and noting that the left-hand side $\|\tilde{\mathbf{x}}_t - \mathbf{x}_t\|$ and right-hand side $\|\tilde{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}\|$ telescope, we obtain

$$\begin{aligned}
& \mathrm{C}_{\text{Interp}}(1,T) \\
\leq\ & \sum_{t=1}^{T} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_T - \mathbf{x}_T\| \\
\leq\ & \sum_{t \in \{R_j\}_{j=1}^k} 2 \cdot \mathrm{C}_{\text{Rob}}(t,t) + (\mu(\mathcal{X}) + 2\gamma)\mathrm{C}_{\text{Adv}}(t,t) + \sum_{t \in [T] \setminus \{R_j\}_{j=1}^k} \mathrm{C}_{\text{Adv}}(t,t) \\
\leq\ & 2 \cdot \mathrm{C}_{\text{Rob}}(1,R_k) + (\mu(\mathcal{X}) + 2\gamma)\mathrm{C}_{\text{Adv}}(1,T) \\
\leq\ & 2\delta \cdot \mathrm{C}_{\text{Adv}}(1,R_k) + (\mu(\mathcal{X}) + 2\gamma)\mathrm{C}_{\text{Adv}}(1,T) \\
\leq\ & (\mu(\mathcal{X}) + \epsilon)\mathrm{C}_{\text{Adv}}(1,T)
\end{aligned}$$

where the second to last inequality follows from the assumption that the algorithm is in the "Rob" phase at time R_k, implying $\mathrm{C}_{\text{Rob}}(1,R_k) \leq \delta \cdot \mathrm{C}_{\text{Adv}}(1,R_k)$; and in the last inequality we use the assumption on the parameters that $2\gamma + 2\delta = \epsilon$. This gives the competitive bound with respect to Adv. Note that we can repeat the same argument with truncated time horizon to obtain that Interp is $(\mu(\mathcal{X}) + \epsilon)$-competitive with respect to Adv at every timestep. □

Proof of competitiveness with respect to Rob. Define the potential function $\phi_t = c\|\mathbf{x}_t - \mathbf{s}_t\|$, with $c > 0$ to be determined later.

Let $t' \in \{0, \ldots, T\}$ be the last time interval in which the algorithm's decision is determined by line 4 of the algorithm, or equivalently, the greatest t such that $\mathrm{C}_{\textsc{Rob}}(1,t) \geq \delta \cdot \mathrm{C}_{\textsc{Adv}}(1,t)$. Applying the competitive bound of Interp with respect to Adv to the subhorizon $t = 1, \ldots, t'$, we have $\mathrm{C}_{\textsc{Interp}}(1,t') \leq (\mu(\mathcal{X}) + \epsilon)\mathrm{C}_{\textsc{Adv}}(1,t')$. By the triangle inequality, and since all algorithms begin at the same starting point $\mathbf{x}_0$, we have $\|\tilde{\mathbf{x}}_{t'} - \mathbf{s}_{t'}\| \leq \mathrm{C}_{\textsc{Adv}}(1,t') + \mathrm{C}_{\textsc{Rob}}(1,t')$. Putting these together, we have

$$\begin{aligned} \mathrm{C}_{\textsc{Interp}}(1,t') + \phi_{t'} &= \mathrm{C}_{\textsc{Interp}}(1,t') + c\|\tilde{\mathbf{x}}_{t'} - \mathbf{s}_{t'}\| \\ &\leq (\mu(\mathcal{X}) + \epsilon)\mathrm{C}_{\textsc{Adv}}(1,t') + c(\mathrm{C}_{\textsc{Adv}}(1,t') + \mathrm{C}_{\textsc{Rob}}(1,t')) \\ &\leq \left(\frac{\mu(\mathcal{X}) + \epsilon + c}{\delta} + c\right)\mathrm{C}_{\textsc{Rob}}(1,t'). \end{aligned} \tag{2.27}$$

Now consider arbitrary $t \in \{t'+1, \ldots, T\}$. We distinguish two cases. First, suppose $\mathbf{x}_t = \mathbf{s}_t$. Then

$$\begin{aligned} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \phi_t - \phi_{t-1} &= f_t(\mathbf{s}_t) + \|\mathbf{s}_t - \mathbf{x}_{t-1}\| + c\|\mathbf{s}_t - \mathbf{s}_t\| - c\|\mathbf{s}_{t-1} - \mathbf{x}_{t-1}\| \\ &\leq f_t(\mathbf{s}_t) + \|\mathbf{s}_t - \mathbf{s}_{t-1}\| + \|\mathbf{s}_{t-1} - \mathbf{x}_{t-1}\| - c\|\mathbf{s}_{t-1} - \mathbf{x}_{t-1}\| \\ &\leq \mathrm{C}_{\textsc{Rob}}(t,t) \end{aligned} \tag{2.28}$$

where the final inequality holds so long as $c \geq 1$.

On the other hand, suppose $\mathbf{x}_t \neq \mathbf{s}_t$. Observe that

$$\begin{aligned} \|\mathbf{x}_t - \mathbf{s}_t\| &\leq \|\mathbf{z}_t - \mathbf{s}_{t-1}\| && \text{by line 8 of the algorithm} \\ &= \|\mathbf{y}_t - \mathbf{s}_{t-1}\| - \gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t) && \text{by line 7 of the algorithm and } \mathbf{x}_t \neq \mathbf{s}_t \\ &\leq \|\mathbf{x}_{t-1} - \mathbf{s}_{t-1}\| - \gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t) && \text{by line 6 of the algorithm.} \end{aligned} \tag{2.29}$$

Then noting that $\mathbf{x}_t = \lambda \mathbf{s}_t + (1-\lambda)\tilde{\mathbf{x}}_t$ for some $\lambda \in [0,1]$, we have

$$\begin{aligned} & f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \phi_t - \phi_{t-1} \\ \leq\ & \lambda f_t(\mathbf{s}_t) + (1-\lambda) f_t(\tilde{\mathbf{x}}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| - c\gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t) && (2.30) \\ \leq\ & f_t(\mathbf{s}_t) + f_t(\tilde{\mathbf{x}}_t) + \|\mathbf{x}_t - \mathbf{z}_t\| + \|\mathbf{z}_t - \mathbf{y}_t\| + \|\mathbf{y}_t - \mathbf{x}_{t-1}\| - c\gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t) && (2.31) \\ \leq\ & \mathrm{C}_{\textsc{Rob}}(t,t) + f_t(\tilde{\mathbf{x}}_t) + \gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t) + \|\mathbf{y}_t - \mathbf{x}_{t-1}\| - c\gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t) && (2.32) \\ \leq\ & \mathrm{C}_{\textsc{Rob}}(t,t) + (k(\mathcal{X}) + \gamma - c\gamma)\mathrm{C}_{\textsc{Adv}}(t,t) && (2.33) \\ \leq\ & \mathrm{C}_{\textsc{Rob}}(t,t) && (2.34) \end{aligned}$$

where (2.30) follows from convexity and (2.29), (2.31) follows from the triangle inequality, (2.32) follows from bounding $\|x_t - z_t\|$ via Corollary 2.E.5 and $\|z_t - y_t\|$

via algorithm line (7), (2.33) follows from the observation in Corollary 2.E.6, and (2.34) holds so long as $c \geq 1 + \frac{k(\mathcal{X})}{\gamma}$.

Thus we set $c = 1 + \frac{k(\mathcal{X})}{\gamma}$; summing (2.28) and (2.34) over times $t' + 1, \ldots, T$ and adding to (2.27), we obtain

$$\begin{aligned} \mathrm{C}_{\textsc{Interp}}(1,T) &\leq \left(1 + \frac{k(\mathcal{X})}{\gamma} + \frac{\mu(\mathcal{X}) + \epsilon + 1 + \frac{k(\mathcal{X})}{\gamma}}{\delta}\right)\mathrm{C}_{\textsc{Rob}}(1,t') + \mathrm{C}_{\textsc{Rob}}(t'+1,T) \\ &\leq \left(1 + \frac{k(\mathcal{X})}{\gamma} + \frac{\mu(\mathcal{X}) + \epsilon + 1 + \frac{k(\mathcal{X})}{\gamma}}{\delta}\right)\mathrm{C}_{\textsc{Rob}}(1,T). \end{aligned}$$

□

Parameter optimization

We conclude with a brief comment on the optimal selection of parameters γ, δ for Interp. If we minimize the competitive bound of Interp with respect to Rob over parameters $\gamma, \delta > 0$ satisfying $2\gamma + 2\delta = \epsilon$, then we obtain the following $O(\frac{1}{\epsilon^2})$ bound on the competitive ratio with respect to Rob (with arguments of $\mu(\mathcal{X}), k(\mathcal{X})$ suppressed):

$$3 + \frac{2(\epsilon + k(\mathcal{X})(4+\epsilon) + \epsilon\mu(\mathcal{X})) + 4\sqrt{k(\mathcal{X})(2+\epsilon)(2k(\mathcal{X}) + \epsilon(1+\epsilon+\mu(\mathcal{X})))}}{\epsilon^2}$$

which is obtained by setting

$$\gamma = \frac{\sqrt{k(\mathcal{X})(2+\epsilon)(2k(\mathcal{X}) + \epsilon(1+\epsilon+\mu(\mathcal{X})))} - k(\mathcal{X})(2+\epsilon)}{2(1 - k(\mathcal{X}) + \epsilon + \mu(\mathcal{X}))}$$

and

$$\delta = \frac{\epsilon}{2} - \gamma.$$

With parameters chosen optimally thus, Interp is $(\mu(\mathcal{X}) + \epsilon, O(\epsilon^{-2}))$-bicompetitive. Moreover, even if $\mu(\mathcal{X})$ and $k(\mathcal{X})$ are not known exactly, simply setting $\gamma = \delta = \frac{\epsilon}{4}$ gives an (up to a constant factor) identical $(\mu(\mathcal{X}) + \epsilon, O(\epsilon^{-2}))$-bicompetitiveness.

2.F Proof of Theorem 2.4.5

We prove Theorem 2.4.5 in two parts: we first prove the competitive ratio of BdInterp with respect to Adv, and then we prove the competitive ratio with respect to Rob. The robustness and consistency claim then follows immediately from the bicompetitive bound and the observation in Section 2.A.

Proof of competitiveness with respect to Adv. We define "phases" of the algorithm as follows: if $\mathbf{x}_t$ is determined by line 4 of the algorithm, then the advice is in the "Adv" phase. Otherwise, if $\mathbf{x}_t$ is determined by line 8, then the advice is in the "Rob" phase. We refer to the time indices in which the algorithm is in the "Rob" phase as $R_1, \ldots, R_k \in [T]$ (where $k \le T$, and $R_1 < \cdots < R_k$ are in increasing order). If the algorithm is never in the "Rob" phase, then $\mathbf{x}_t = \text{Adv}_t \ \forall t \in [T]$, and thus BdInterp is 1-competitive with respect to Adv. Thus we restrict to the case that there is at least one time index in which the algorithm is in the "Rob" phase. By design, for each $j \in [k]$, $\mathrm{C}_{\text{Rob}}(1, R_j) \le \delta \cdot \mathrm{C}_{\text{Adv}}(1, R_j)$.

Now we break into two cases depending on the phase. First, suppose the BdInterp is in the "Adv" phase. This means that $\mathbf{x}_t = \tilde{\mathbf{x}}_t$. Then

$$\begin{aligned} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_t - \mathbf{x}_t\| &= f_t(\tilde{\mathbf{x}}_t) + \|\tilde{\mathbf{x}}_t - \mathbf{x}_{t-1}\| \\ &\le f_t(\tilde{\mathbf{x}}_t) + \|\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}\| + \|\tilde{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}\| \end{aligned} \tag{2.35}$$

follows immediately from the triangle inequality.

Second, consider the case that the algorithm is in the "Rob" phase. This means that $\mathbf{x}_t$ is determined by line 8 of the algorithm; and there exists some $\lambda \in [0, 1]$ for which $\mathbf{x}_t = \lambda \mathbf{s}_t + (1 - \lambda)\tilde{\mathbf{x}}_t$. In this case, observe

$$\begin{aligned} & f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_t - \mathbf{x}_t\| \\ \le\ & \lambda f_t(\mathbf{s}_t) + (1-\lambda) f_t(\tilde{\mathbf{x}}_t) + 2\|\mathbf{x}_t - \mathbf{y}_t\| + \|\mathbf{y}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_t - \mathbf{y}_t\| && (2.36) \\ \le\ & f_t(\mathbf{s}_t) + f_t(\tilde{\mathbf{x}}_t) + 2\gamma \cdot \mathrm{C}_{\text{Adv}}(t, t) + \|\mathbf{y}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_t - \mathbf{y}_t\| && (2.37) \end{aligned}$$

where (2.36) follows from convexity of f_t and the triangle inequality, (2.37) follows via algorithm line 8. Observing that $\mathbf{x}_{t-1} = \nu \tilde{\mathbf{x}}_{t-1} + (1 - \nu)\mathbf{s}_{t-1}$, we can use the triangle inequality to obtain

$$\|\mathbf{y}_t - \mathbf{x}_{t-1}\| \le \nu \|\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}\| + (1 - \nu)\|\mathbf{s}_t - \mathbf{s}_{t-1}\|. \tag{2.38}$$

Moreover, observe

$$\begin{aligned} \|\tilde{\mathbf{x}}_t - \mathbf{y}_t\| &= (1 - \nu)\|\tilde{\mathbf{x}}_t - \mathbf{s}_t\| \\ &\le (1 - \nu)(\|\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}\| + \|\tilde{\mathbf{x}}_{t-1} - \mathbf{s}_{t-1}\| + \|\mathbf{s}_t - \mathbf{s}_{t-1}\|) \\ &= (1 - \nu)(\|\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}\| + \|\mathbf{s}_t - \mathbf{s}_{t-1}\|) + \|\tilde{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}\| \end{aligned} \tag{2.39}$$

where the final equality follows by definition of ν. Applying (2.38) and (2.39) to (2.37), we obtain

$$\begin{aligned} & f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_t - \mathbf{x}_t\| \\ \le\ & 2 \cdot \mathrm{C}_{\text{Rob}}(t, t) + (1 + 2\gamma)\mathrm{C}_{\text{Adv}}(t, t) + \|\tilde{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}\|. \end{aligned} \tag{2.40}$$

Summing (2.35) and (2.40) over time and noting that the left-hand side $\|\tilde{\mathbf{x}}_t - \mathbf{x}_t\|$ and right-hand side $\|\tilde{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}\|$ telescope, we obtain

$$
\begin{aligned}
& \mathrm{C}_{\textsc{BdInterp}}(1,T) \\
\leq\; & \sum_{t=1}^{T} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \|\tilde{\mathbf{x}}_T - \mathbf{x}_T\| \\
\leq\; & \sum_{t \in \{R_j\}_{j=1}^k} 2 \cdot \mathrm{C}_{\textsc{Rob}}(t,t) + (1+2\gamma)\mathrm{C}_{\textsc{Adv}}(t,t) + \sum_{t \in [T] \setminus \{R_j\}_{j=1}^k} \mathrm{C}_{\textsc{Adv}}(t,t) \\
\leq\; & 2 \cdot \mathrm{C}_{\textsc{Rob}}(1,R_k) + (1+2\gamma)\mathrm{C}_{\textsc{Adv}}(1,T) \\
\leq\; & 2\delta \cdot \mathrm{C}_{\textsc{Adv}}(1,R_k) + (1+2\gamma)\mathrm{C}_{\textsc{Adv}}(1,T) \\
\leq\; & (1+\epsilon)\mathrm{C}_{\textsc{Adv}}(1,T)
\end{aligned}
$$

where the second to last inequality follows from the assumption that the algorithm is in the "Rob" phase at time R_k, implying $\mathrm{C}_{\textsc{Rob}}(1,R_k) \leq \delta \cdot \mathrm{C}_{\textsc{Adv}}(1,R_k)$; and in the last inequality we use the assumption on the parameters that $2\gamma + 2\delta = \epsilon$. This gives the competitive bound with respect to Adv. Note that we can repeat the same argument with truncated time horizon to obtain that BdInterp is $(1+\epsilon)$-competitive with respect to Adv at every timestep. □

Proof of competitiveness with respect to Rob. Define the potential function $\phi_t = c\frac{\|\mathbf{x}_t - \mathbf{s}_t\|}{\|\tilde{\mathbf{x}}_t - \mathbf{s}_t\|}$, with $c > 0$ to be determined later (we set $\phi_t := 0$ in the case that $\tilde{\mathbf{x}}_t = \mathbf{s}_t$).

Let $t' \in \{0,\ldots,T\}$ be the last time interval in which the algorithm's decision is determined by line 4 of the algorithm, or equivalently, the greatest t such that $\mathrm{C}_{\textsc{Rob}}(1,t) \geq \delta \cdot \mathrm{C}_{\textsc{Adv}}(1,t)$. Applying the competitive bound of BdInterp with respect to Adv to the subhorizon $t = 1,\ldots,t'$, we have $\mathrm{C}_{\textsc{BdInterp}}(1,t') \leq (1+\epsilon)\mathrm{C}_{\textsc{Adv}}(1,t')$. Thereby we obtain

$$
\begin{aligned}
\mathrm{C}_{\textsc{BdInterp}}(1,t') + \phi_{t'} &\leq (1+\epsilon)\mathrm{C}_{\textsc{Adv}}(1,t') + c \\
&\leq \frac{1+\epsilon}{\delta}\mathrm{C}_{\textsc{Rob}}(1,t') + c. \qquad (2.41)
\end{aligned}
$$

where in the first inequality we have used the fact that $\|\mathbf{x}_t - \mathbf{s}_t\| \leq \|\tilde{\mathbf{x}}_t - \mathbf{s}_t\|$ for all t.

Now consider arbitrary $t \in \{t'+1, \dots, T\}$. We distinguish two cases. First, suppose $\mathbf{x}_t = \mathbf{s}_t$ and $\tilde{\mathbf{x}}_{t-1} \neq \mathbf{s}_{t-1}$. Then

$$\begin{aligned} f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \phi_t - \phi_{t-1} &= f_t(\mathbf{s}_t) + \|\mathbf{s}_t - \mathbf{x}_{t-1}\| - c\frac{\|\mathbf{x}_{t-1} - \mathbf{s}_{t-1}\|}{\|\tilde{\mathbf{x}}_{t-1} - \mathbf{s}_{t-1}\|} \\ &\leq f_t(\mathbf{s}_t) + \|\mathbf{s}_t - \mathbf{s}_{t-1}\| \\ &\quad + \|\mathbf{x}_{t-1} - \mathbf{s}_{t-1}\| - c\frac{\|\mathbf{x}_{t-1} - \mathbf{s}_{t-1}\|}{\|\tilde{\mathbf{x}}_{t-1} - \mathbf{s}_{t-1}\|} \\ &\leq \mathrm{C}_{\textsc{Rob}}(t,t) \end{aligned} \tag{2.42}$$

where the final inequality holds so long as $c \geq D$, by D-boundedness of Adv and Rob. Clearly (2.42) will also hold in the case that $\tilde{\mathbf{x}}_{t-1} = \mathbf{s}_{t-1}$, since this will imply $\mathbf{x}_{t-1} = \mathbf{s}_{t-1}$.

On the other hand, suppose $\mathbf{x}_t \neq \mathbf{s}_t$. Thus we can assume that $\tilde{\mathbf{x}}_t \neq \mathbf{s}_t$ and $\tilde{\mathbf{x}}_{t-1} \neq \mathbf{s}_{t-1}$. First, note that

$$\frac{\|\mathbf{x}_t - \mathbf{s}_t\|}{\|\tilde{\mathbf{x}}_t - \mathbf{s}_t\|} = \frac{\|\mathbf{y}_t - \mathbf{s}_t\| - \gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t)}{\|\tilde{\mathbf{x}}_t - \mathbf{s}_t\|} \tag{2.43}$$

$$\begin{aligned} &= \nu - \frac{\gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t)}{\|\tilde{\mathbf{x}}_t - \mathbf{s}_t\|} \\ &\leq \frac{\|\mathbf{x}_{t-1} - \mathbf{s}_{t-1}\|}{\|\tilde{\mathbf{x}}_{t-1} - \mathbf{s}_{t-1}\|} - \frac{\gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t)}{D} \end{aligned} \tag{2.44}$$

where (2.43) follows from line 8 of the algorithm and $\mathbf{x}_t \neq \mathbf{s}_t$, and (2.44) follows by definition of ν and the D-boundedness of Adv, Rob.

Then noting that by convexity, $\mathbf{x}_t = \lambda \mathbf{s}_t + (1-\lambda)\tilde{\mathbf{x}}_t$ for some $\lambda \in [0,1]$, we have

$$\begin{aligned} & f_t(\mathbf{x}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| + \phi_t - \phi_{t-1} \\ \leq\ & \lambda f_t(\mathbf{s}_t) + (1-\lambda) f_t(\tilde{\mathbf{x}}_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\| - c\frac{\gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t)}{D} && (2.45) \\ \leq\ & f_t(\mathbf{s}_t) + f_t(\tilde{\mathbf{x}}_t) + \|\mathbf{x}_t - \mathbf{y}_t\| + \|\mathbf{y}_t - \mathbf{x}_{t-1}\| - c\frac{\gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t)}{D} && (2.46) \\ \leq\ & f_t(\mathbf{s}_t) + f_t(\tilde{\mathbf{x}}_t) + \gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t) + \nu\|\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}\| \\ & + (1-\nu)\|\mathbf{s}_t - \mathbf{s}_{t-1}\| - c\frac{\gamma \cdot \mathrm{C}_{\textsc{Adv}}(t,t)}{D} && (2.47) \\ \leq\ & \mathrm{C}_{\textsc{Rob}}(t,t) + \left(1 + \gamma - \frac{c\gamma}{D}\right)\mathrm{C}_{\textsc{Adv}}(t,t) \\ \leq\ & \mathrm{C}_{\textsc{Rob}}(t,t) && (2.48) \end{aligned}$$

where (2.45) follows from convexity and (2.44), (2.46) follows from the triangle inequality, and (2.47) follows from (2.38) and line 8 of the algorithm. The final inequality (2.48) holds as long as $c \geq D + \frac{D}{\gamma}$.

Thus we set $c = D + \frac{D}{\gamma}$; summing (2.42) and (2.48) over times $t' + 1, \ldots, T$ and adding to (2.41), we obtain

$$\begin{aligned}
\mathrm{C}_{\textsc{Interp}}(1,T) &\leq \frac{1+\epsilon}{\delta}\mathrm{C}_{\textsc{Rob}}(1,t') + D + \frac{D}{\gamma} + \mathrm{C}_{\textsc{Rob}}(t'+1,T) \\
&\leq \frac{1+\epsilon}{\delta}\mathrm{C}_{\textsc{Rob}}(1,T) + D + \frac{D}{\gamma} \\
&\leq \left(D + \frac{D}{\gamma} + \frac{1+\epsilon}{\delta}\right)\mathrm{C}_{\textsc{Rob}}(1,T)
\end{aligned}$$

where in the final inequality we have used the assumption that $\mathrm{C}_{\textsc{Rob}} \geq 1$. □

Parameter optimization

To conclude, we briefly comment on the optimal selection of parameters γ, δ for BdInterp. Optimizing the competitive bound of BdInterp with respect to Rob over those $\gamma, \delta > 0$ satisfying $2\gamma + 2\delta = \epsilon$, we obtain the following $\mathcal{O}(\frac{D}{\epsilon})$-competitive bound with respect to Rob:

$$2 + D + \frac{2(1+D) + 4\sqrt{D(1+\epsilon)}}{\epsilon}$$

which is obtained by setting

$$\gamma = \frac{D\epsilon}{2(D + \sqrt{D(1+\epsilon)}}$$

and

$$\delta = \frac{\epsilon}{2} - \gamma.$$

With parameters chosen optimally thus, BdInterp is $(1+\epsilon, \mathcal{O}(D\epsilon^{-1}))$-bicompetitive. Moreover, even if D is not known exactly *a priori*, simply setting $\gamma = \delta = \frac{\epsilon}{4}$ gives an (up to a constant factor) identical $(1+\epsilon, \mathcal{O}(D\epsilon^{-1}))$-bicompetitiveness.

2.G Robustness and Consistency Corollaries of Theorems 2.4.4 and 2.4.5

In this section, we detail the upper bounds on robustness and consistency resulting from Theorems 2.4.4 and 2.4.5 on CFC and each of its subclasses defined in Section 2.A. Each of these corollaries follows immediately upon instantiating the robust algorithm Rob provided as input to Interp (Algorithm 4) or BdInterp (Algorithm 5) with a competitive algorithm whose competitive ratio is listed in Table 2.2. We begin with the corollaries of Theorem 2.4.4.

Corollary 2.G.1. *(i)* *INTERP (Algorithm 4) with ROB chosen as the functional Steiner point algorithm ([106]) is* $(\mu(\mathbb{R}^d, \|\cdot\|) + \epsilon)$*-consistent and* $O(\frac{d}{\epsilon^2})$*-robust for* CFC *and* CBC *on* $\mathbb{R}^d$ *with any norm.*

(ii) *INTERP (Algorithm 4) with ROB chosen as the low-dimensional chasing algorithm of [108] is* $(\sqrt{2} + \epsilon)$*-consistent and* $O(\frac{k}{\epsilon^2})$*-robust for* kCBC *on* $(\mathbb{R}^d, \|\cdot\|_{\ell^2})$.

(iii) *INTERP (Algorithm 4) with ROB chosen as the greedy algorithm ([103]) is* $(\mu(\mathcal{X}) + \epsilon)$*-consistent and* $O(\frac{1}{\alpha\epsilon^2})$*-robust for* αCFC *on any normed vector space* $\mathcal{X}$.

(iv) *INTERP (Algorithm 4) with ROB chosen as the greedy OBD algorithm ([107]) is* $(\sqrt{2} + \epsilon)$*-consistent and* $O(\frac{1}{\alpha^{1/2}\epsilon^2})$*-robust for* αCFC *on* $(\mathbb{R}^d, \|\cdot\|_{\ell^2})$.

(v) *INTERP (Algorithm 4) with ROB chosen as the Move towards Minimizer algorithm ([108]) is* $(\sqrt{2} + \epsilon)$*-consistent and* $O(\frac{2^{\gamma/2}\kappa}{\epsilon^2})$*-robust for* (κ, γ)CFC *on* $(\mathbb{R}^d, \|\cdot\|_{\ell^2})$.

In particular, each of the consistency bounds is $\sqrt{2} + \epsilon$ *in the case that the decision space is Hilbert.*

We now present the corollaries of Theorem 2.4.5.

Corollary 2.G.2. *In each of the following, suppose that* $(\textsc{Adv}, \textsc{Rob})$ *are* D*-bounded and* $\mathrm{C}_{\textsc{Rob}} \geq 1$.

(i) *BDINTERP (Algorithm 5) with ROB chosen as the functional Steiner point algorithm ([106]) is* $(1 + \epsilon)$*-consistent and* $O(\frac{dD}{\epsilon})$*-robust for* CFC *and* CBC *on* $\mathbb{R}^d$ *with any norm.*

(ii) *BDINTERP (Algorithm 5) with ROB chosen as the low-dimensional chasing algorithm of [108] is* $(1 + \epsilon)$*-consistent and* $O(\frac{kD}{\epsilon})$*-robust for* kCBC *on* $(\mathbb{R}^d, \|\cdot\|_{\ell^2})$.

(iii) *BDINTERP (Algorithm 5) with ROB chosen as the greedy algorithm ([103]) is* $(1 + \epsilon)$*-consistent and* $O(\frac{D}{\alpha\epsilon})$*-robust for* αCFC *on any normed vector space.*

(iv) *BDINTERP (Algorithm 5) with ROB chosen as the Greedy OBD algorithm ([107]) is* $(1 + \epsilon)$*-consistent and* $O(\frac{D}{\alpha^{1/2}\epsilon})$*-robust for* αCFC *on* $(\mathbb{R}^d, \|\cdot\|_{\ell^2})$.

(v) *BDINTERP (Algorithm 5) with ROB chosen as the Move towards Minimizer algorithm ([108]) is* $(1 + \epsilon)$*-consistent and* $O(\frac{2^{\gamma/2}\kappa D}{\epsilon})$*-robust for* (κ, γ)CFC *on* $(\mathbb{R}^d, \|\cdot\|_{\ell^2})$.

Chapter 3

OPTIMAL ROBUSTNESS-CONSISTENCY TRADEOFFS FOR LEARNING-AUGMENTED METRICAL TASK SYSTEMS

Moving beyond the convex setting of the previous chapter, we now examine the problem of designing *learning-augmented* algorithms for the more general *metrical task systems* (MTS) problem that exploit machine-learned advice while maintaining rigorous, worst-case guarantees on performance. We propose a randomized algorithm, DART, that achieves this dual objective, providing expected cost within a multiplicative factor $(1 + \epsilon)$ of the machine-learned advice (i.e., *consistency*) while ensuring expected cost within a multiplicative factor $2^{O(1/\epsilon)}$ of a baseline robust algorithm (i.e., *robustness*) for any $\epsilon > 0$. We show that this exponential tradeoff between consistency and robustness is unavoidable in general, but that in important subclasses of MTS, such as when the metric space has bounded diameter and in the k-server problem, our algorithm achieves improved, polynomial tradeoffs between consistency and robustness. We further show that, given an *a priori* bound D on the distance between the advice and robust decisions, a deterministic algorithm can obtain $(1 + \epsilon)$-consistency and cost within a multiplicative factor $O(\frac{1}{\epsilon})$ of the robust baseline algorithm, with an additive constant of $O(\frac{D}{\epsilon})$. We demonstrate the practical value of these algorithms in a case study on cogeneration power plant operation under high renewables penetration. In particular, our algorithms enable significant efficiency improvements while ensuring robustness to potentially poor machine learning performance.

This chapter is primarily based on the following paper:

[1] N. Christianson, J. Shen, and A. Wierman, "Optimal Robustness-Consistency Tradeoffs for Learning-Augmented Metrical Task Systems," in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, PMLR, Apr. 2023, pp. 9377–9399. [Online]. Available: https://proceedings.mlr.press/v206/christianson23a.html.

 In addition, Sections 3.6, 3.7, and 3.E, which present the deterministic algorithm, its performance analysis, and experimental results for the cogeneration power plant operation application, are adapted from the paper

[1] N. Christianson, C. Yeh, T. Li, M. Hosseini, M. T. Rad, A. Golmohammadi, and A. Wierman, "Robust Machine-Learned Algorithms for Efficient Grid Operation," *Environmental Data Science*, vol. 4, e24, Apr. 2025, ISSN: 2634-4602. DOI: 10.1017/eds.2024.28. [Online]. Available: https://www.cambridge.org/core/journals/environmental-data-science/article/robust-machinelearned-algorithms-for-efficient-grid-operation/0E29B936A8FEE565B169F70372B7F9DE.

3.1 Introduction

The *metrical task systems* (MTS) problem is a central problem in the theory of online algorithms, encompassing a wide range of problems broadly characterized as "online optimization with switching costs" such as *convex function chasing* (CFC) and *k-server*. In MTS, a decision-maker is faced with a metric space (X, d) and a sequence of adversarial cost functions $f_1, \ldots, f_T : X \to [0, +\infty]$ that are revealed *online*; after the function f_t is revealed, the decision-maker chooses a decision $x_t \in X$ and pays the *service* cost $f_t(x_t)$ as well as the *switching* or *movement* cost $d(x_t, x_{t-1})$, which penalizes changing decisions. The MTS problem has deep connections with online learning [116, 130, 131] and broad applicability to problems such as energy system operation [132, 133], datacenter operation [37, 38, 134], smoothed online regression and clustering [99, 135, 136], and logistics [137]. MTS algorithms are designed to minimize the *competitive ratio*, which quantifies the worst-case ratio in cost between an algorithm and the offline optimal sequence of decisions (Definition 3.2.1). The competitive ratio of MTS algorithms grows in the cardinality or dimension of the decision space; for instance, if $|X| = n$, any deterministic algorithm is $\Omega(n)$-competitive and any randomized algorithm is $\Omega(\log n)$-competitive [76, 138].

Due to the worst-case nature of the competitive ratio, traditional algorithms for MTS are conservative and may perform poorly in high-dimensional settings. In many real-world sequential decision-making tasks, however, significant data is available concerning typical problem instances, enabling data-driven, machine-learned (ML) algorithms to outperform traditional algorithms, which ignore such data. Despite this excellent practical performance, ML algorithms come with no *a priori* guarantees on worst-case behavior. As such, their performance may be jeopardized at deployment time if they are faced with distribution shift or unseen problem instances.

The tension between ML algorithms' excellent average-case performance and their lack of worst-case guarantees has motivated the development of *learning-augmented* algorithms for a wide range of online problems such as ski-rental, scheduling, and caching [69, 70, 109, 139]. These algorithms are designed to exploit the performance of untrusted or "black-box" *advice* (e.g., from an ML algorithm) while maintaining rigorous guarantees on worst-case performance. Specifically, learning-augmented algorithms are designed to give simultaneous guarantees of *consistency*—a competitive guarantee against the advice—along with *robustness*—a worst-case competitive ratio guarantee (Definition 3.2.2). Tunable guarantees are typically sought so that $(1+\epsilon)$-consistency can be obtained alongside bounded robustness for *any* $\epsilon > 0$, enabling better exploitation of good advice when ϵ is chosen to be small.

Antoniadis et al. [75] propose two algorithms that switch between an advice algorithm and a C-competitive algorithm for MTS or a special case thereof, giving guarantees of robustness and consistency for any MTS problem. In particular, their deterministic algorithm achieves 9-consistency and $9C$-robustness and their randomized algorithm achieves expected cost bounded by $(1+\epsilon)\cdot\min\{\mathrm{C}_{\mathrm{ADV}}, C\cdot\mathrm{C}_{\mathrm{OPT}}\}+O(\frac{D}{\epsilon})$, where $D = \mathrm{diam}(X)$ and $\mathrm{C}_{\mathrm{ADV}}, \mathrm{C}_{\mathrm{OPT}}$ are the advice and offline optimal costs, respectively. However, their deterministic algorithm cannot improve upon 9-consistency and the randomized algorithm is limited by the additive $O(\frac{D}{\epsilon})$ term, which precludes obtaining arbitrarily small consistency (e.g., when $\mathrm{C}_{\mathrm{ADV}}$ is small relative to the diameter) and causes the bound to degrade or fail as the diameter of the metric space grows. This diameter-dependence is of particular limitation to special cases of MTS such as CFC and k-server, where the natural setting is an unbounded metric space like $\mathbb{R}^n$. In addition, this randomized algorithm applies the classic multiplicative weights approach of [116], which is not clearly optimal in this setting.

Several subsequent works obtain robustness and consistency bounds independent of diameter in special cases. Rutten, Christianson, et al. [113] propose an algorithm achieving $(1+\epsilon)$-consistency and $2^{\tilde{O}(\frac{1}{\alpha\epsilon})}$-robustness under certain conditions on $\alpha, \epsilon > 0$ when service cost functions f_t are restricted to be *α-polyhedral* (Definition 3.2.4). In the case of convex function chasing (Section 3.2) on $(\mathbb{R}^n, \|\cdot\|_{\ell^2})$, Christianson et al. [140] propose a $(\sqrt{2}+\epsilon)$-consistent, $O(\frac{n}{\epsilon^2})$-robust algorithm, and Rutten, Christianson, et al. [113] give a $(1+\epsilon)$-consistent, $O(\frac{1}{\epsilon})$-robust algorithm for the one-dimensional case (see Chapter 2.4). Lindermayr et al. [141] give an algorithm for k-server (Section 3.2) on $\mathbb{R}$ that achieves $(1 + O(\epsilon))$-consistency and $O(\frac{1}{\epsilon^{k-1}})$-robustness.

These latter results indicate that in certain subclasses of MTS, it is possible to obtain robustness and consistency bounds that are independent of metric space diameter. However, these results only exist for a few subclasses of MTS, and do not always guarantee $(1 + \epsilon)$-consistency for arbitrarily small $\epsilon > 0$, thus limiting the exploitation of good advice. The following, important questions remain open: *Does there exist a general algorithm for* MTS *and its subclasses that achieves* $(1 + \epsilon)$-*consistency for any* $\epsilon > 0$ *and bounded robustness? And is it possible to obtain robustness bounded independently of the metric space diameter?*

Contributions

In this work, we answer the above questions in the affirmative. Specifically, we propose a randomized algorithm, Dart (Algorithm 6), that, given any advice and any C-competitive algorithm for MTS or a special case thereof, achieves $(1 + \epsilon)$-consistency and $2^{O(1/\epsilon)}C$-robustness (Theorem 3.3.1), with robustness independent of the diameter of the metric space.

This main result implies several robustness and consistency bounds for subclasses of MTS (Corollary 3.3.2), which we summarize in Table 3.1. In particular, we answer the question posed by [140] (Chapter 2) of whether $(1 + \epsilon)$-consistency and bounded robustness can be achieved for convex function chasing (CFC) on unbounded domains with ϵ arbitrarily close to 0. We further prove lower bounds on robustness and consistency for MTS and CFC, showing that our upper bounds are essentially tight: *any* $(1 + \epsilon)$-consistent algorithm must have robustness $2^{\Omega(1/\epsilon)}$ (Theorems 3.4.1, 3.4.2). Despite this exponential tradeoff for MTS and CFC in general settings, we show by a refined analysis that Dart actually achieves robustness $O(\frac{C}{\epsilon})$ when the space's diameter is bounded, with an additive term on the robustness matching the dependence on diameter of [75] (Theorem 3.5.1). Moreover, we find that Dart achieves $O(\frac{k}{\epsilon})$-robustness for the *k-server* problem, giving the best known robustness and consistency tradeoff in general metric spaces for this widely-studied special case of MTS (Theorem 3.5.3). We also consider the problem of k-chasing convex, α-polyhedral functions, a generalization of both k-server and CFC, and we find that Dart guarantees robustness $O(\frac{k}{\alpha\epsilon})$ in the one-dimensional setting (Theorem 3.5.4).

In addition to Dart, we propose a *deterministic* algorithm, DetRobustML, that achieves $(1+\epsilon)$-consistency and $O(\frac{C}{\epsilon})$-robustness when the diameter of the space is bounded, with an additive constant on the robustness matching that incurred by Dart in this setting (Theorem 3.6.1). While this algorithm requires prior knowledge of the diameter bound and does not ensure the diameter-independent robustness achieved by Dart, its determinism may be desirable to decision-makers who wish to avoid potential added risk resulting from the randomness used by Dart.

Finally, we present an experimental evaluation of Dart and DetRobustML in Section 3.7. These experiments, conducted on a realistic model of cogeneration power plant operation under increasing renewables penetration, show that Dart and DetRobustML can significantly reduce cost by exploiting machine-learned advice, while improving reliability and robustness when the advice performs poorly. Moreover, both algorithms outperform the state-of-the-art multiplicative weights algorithm of Antoniadis et al. [75].

Our algorithms—and Dart especially—are distinguished from prior learning-augmented algorithms for MTS in both their generality and specific MTS-oriented design. Prior algorithms were either devised for different online problems and simply applied off-the-shelf to MTS with advice (e.g., the cow path and multiplicative weights algorithms of Antoniadis et al. [75]), or heavily leveraged geometric and structural assumptions on the problem setting (e.g., convexity in [140] (Chapter 2), α-polyhedrality in [113], $X = \mathbb{R}$ in [141]). In contrast, Dart works for any MTS or special case and is designed principally to achieve $(1+\epsilon)$-consistency with respect to an advice algorithm. Specifically, it operates by updating probabilities assigned to the advice and to a chosen competitive algorithm based on the costs incurred by each algorithm *as well as* the distance between the two algorithms' decisions. The dependence on this latter quantity is important, as this enables obtaining consistency and robustness independent of diameter, and the randomized algorithm of [75] lacks such a dependence.

Proving the performance bounds for Dart requires several technical contributions. Dart's robustness is obtained by directly bounding the extent to which Dart can be led astray by bad advice; this approach requires proving a lower bound on a broad class of sums that includes as a special case the harmonic series (Supplemental Section 3.B). Moreover, the extremal case of this lower bound naturally leads to the robustness and consistency lower bound for MTS (Theorem 3.4.1). Furthermore, our lower bound on robustness and consistency for CFC (Theorem 3.4.2) makes novel

Table 3.1: Summary of prior robustness and consistency results for MTS and its special cases, with this chapter's contributions in bold. Recall $D = \mathrm{diam}(X)$.

Problem	Reference	Consistency	Robustness	Assumptions
MTS $\lvert X\rvert = n$	[142]	—	$O(\log^2 n)\mathrm{C}_{\mathrm{OPT}}$	
	[75]	$9 \cdot \mathrm{C}_{\mathrm{ADV}}$	$9 \cdot O(\log^2 n)\mathrm{C}_{\mathrm{OPT}}$	
	[75]	$(1+\epsilon)\mathrm{C}_{\mathrm{ADV}} + O(\frac{D}{\epsilon})$	$(1+\epsilon)O(\log^2 n)\mathrm{C}_{\mathrm{OPT}} + O(\frac{D}{\epsilon})$	$D < \infty$
	This chapter	$\boldsymbol{(1+\epsilon)\mathrm{C}_{\mathrm{ADV}}}$	$\boldsymbol{\min\{2^{O(1/\epsilon)} O(\log^2 n)\mathrm{C}_{\mathrm{OPT}}, O(\frac{\log^2 n}{\epsilon})\mathrm{C}_{\mathrm{OPT}} + O(\frac{D}{\epsilon})\}}$	
CFC $X \subseteq \mathbb{R}^n$	[106]	—	$n \cdot \mathrm{C}_{\mathrm{OPT}}$	
	[75]	$9 \cdot \mathrm{C}_{\mathrm{ADV}}$	$9n \cdot \mathrm{C}_{\mathrm{OPT}}$	
	[75]	$(1+\epsilon)\mathrm{C}_{\mathrm{ADV}} + O(\frac{D}{\epsilon})$	$(1+\epsilon)n \cdot \mathrm{C}_{\mathrm{OPT}} + O(\frac{D}{\epsilon})$	$D < \infty$
	[140]	$(\sqrt{2}+\epsilon)\mathrm{C}_{\mathrm{ADV}}$	$O\left(\frac{n}{\epsilon^2}\right)\mathrm{C}_{\mathrm{OPT}}$	Euclidean
	This chapter	$\boldsymbol{(1+\epsilon)\mathrm{C}_{\mathrm{ADV}}}$	$\boldsymbol{\min\{2^{O(1/\epsilon)} n \cdot \mathrm{C}_{\mathrm{OPT}}, O(\frac{n}{\epsilon})\mathrm{C}_{\mathrm{OPT}} + O(\frac{D}{\epsilon})\}}$	
k-server	[97]	—	$(2k-1)\mathrm{C}_{\mathrm{OPT}}$	
	[75]	$9 \cdot \mathrm{C}_{\mathrm{ADV}}$	$9(2k-1)\mathrm{C}_{\mathrm{OPT}}$	
	[75]	$(1+\epsilon)\mathrm{C}_{\mathrm{ADV}} + O(\frac{D}{\epsilon})$	$(1+\epsilon)(2k-1)\mathrm{C}_{\mathrm{OPT}} + O(\frac{D}{\epsilon})$	$D < \infty$
	[141]	$(1+O(\epsilon))\mathrm{C}_{\mathrm{ADV}}$	$O(\frac{1}{\epsilon^{k-1}})\mathrm{C}_{\mathrm{OPT}}$	$X = \mathbb{R}$
	This chapter	$\boldsymbol{(1+\epsilon)\mathrm{C}_{\mathrm{ADV}}}$	$\boldsymbol{O(\frac{k}{\epsilon})\mathrm{C}_{\mathrm{OPT}}}$	
k-chasing convex, α-polyhedral functions $X \subseteq \mathbb{R}^n$	[135] **This chapter**	**—**	$\boldsymbol{O(\frac{k}{\alpha})\mathrm{C}_{\mathrm{OPT}}}$	
	[75]	$9 \cdot \mathrm{C}_{\mathrm{ADV}}$	$9 \cdot O(\frac{k}{\alpha})\mathrm{C}_{\mathrm{OPT}}$	
	[75]	$(1+\epsilon)\mathrm{C}_{\mathrm{ADV}} + O(\frac{D}{\epsilon})$	$(1+\epsilon)O(\frac{k}{\alpha})\mathrm{C}_{\mathrm{OPT}} + O(\frac{D}{\epsilon})$	$D < \infty$
	This chapter	$\boldsymbol{(1+\epsilon)\mathrm{C}_{\mathrm{ADV}}}$	$\boldsymbol{\min\{2^{O(1/\epsilon)} O(\frac{k}{\alpha})\mathrm{C}_{\mathrm{OPT}}, O(\frac{k}{\alpha\epsilon})\mathrm{C}_{\mathrm{OPT}} + O(\frac{D}{\epsilon})\}}$	
	This chapter	$\boldsymbol{(1+\epsilon)\mathrm{C}_{\mathrm{ADV}}}$	$\boldsymbol{O(\frac{k}{\alpha\epsilon})\mathrm{C}_{\mathrm{OPT}}}$	$X = \mathbb{R}$

use of an observation (due to [142]) that MTS instances on trees are equivalent, in a certain sense, to CFC instances in a weighted ℓ^1 space. To our knowledge, no prior work has used this correspondence to translate performance bounds on MTS algorithms to results for CFC. Finally, the results we obtain in Section 3.5 all follow via *refined analyses* of the Dart algorithm, and in particular do not require modification of the algorithm or prior knowledge of, e.g., the diameter bound. Thus, Dart is a *unified* algorithm that matches or improves upon the best known results on robustness and consistency for MTS and many of its subclasses.

Notation

Let $\overline{\mathbb{R}}_+$ denote the nonnegative extended reals. We define $[n] := \{1, \ldots, n\}$ for $n \in \mathbb{N}$. Metric spaces (X, d) are assumed to be complete and separable. For a metric space X, $\mathrm{diam}(X) := \sup_{x,y \in X} d(x, y)$. $\Delta(X)$ denotes the set of (Borel) probability measures on the metric space X; when X is finite with cardinality n, we identify this with the simplex Δ_n. For $x \in X$, δ_x is the Dirac measure supported at x. In asymptotic notation involving the variable ϵ, the implied regime is $\epsilon \to 0$.

3.2 Preliminaries

This section introduces the metrical task systems problem and several subclasses that have received significant attention. We motivate these problems with applications to data science and multi-agent logistics, and introduce the notion of *learning-augmented* algorithms that enable breaking past pessimistic worst-case guarantees.

Metrical Task Systems

Let (X, d) be a metric space. In the *metrical task systems* (MTS) problem, at each time $t \in [T]$, a player beginning from some position $x_0 \in X$ observes an adversarially-chosen cost function $f_t : X \to \overline{\mathbb{R}}_+$ and must choose a state $x_t \in X$ to move to. The player then pays both the *service* cost $f_t(x_t)$ as well as the *movement* cost $d(x_t, x_{t-1})$. The time horizon T is unknown to the player *a priori*. An *instance* of MTS is characterized by a metric space (X, d), a starting position x_0, and the cost function sequence $f_1, \ldots, f_T$.

A *deterministic online algorithm* $\textsc{Alg}$ for MTS is a sequence of maps $\textsc{Alg}_t : (\overline{\mathbb{R}}_+^X)^t \to X$ which map the cost functions observed through time t to a decision in X for each $t \in [T]$. That is, upon observing the cost function f_t, $\textsc{Alg}_t(f_1, \ldots, f_t) \in X$ is the decision produced by $\textsc{Alg}$ at time t. When the instance is implicitly understood, we suppress the arguments and simply write $\textsc{Alg}_t$ for $\textsc{Alg}$'s decision at time t. $\textsc{Alg}$ thus incurs cost

$$\mathrm{C}_{\textsc{Alg}} := \sum_{t=1}^{T} f_t(\textsc{Alg}_t) + d(\textsc{Alg}_t, \textsc{Alg}_{t-1}).$$

We define the notation $\mathrm{C}_{\textsc{Alg}}(t, t') := \sum_{\tau=t}^{t'} f_t(\textsc{Alg}_\tau) + d(\textsc{Alg}_\tau, \textsc{Alg}_{\tau-1})$ to reflect $\textsc{Alg}$'s total cost incurred from time t through t'; if $t > t'$, then $\mathrm{C}_{\textsc{Alg}}(t, t') := 0$.

A *randomized* MTS algorithm produces its decisions randomly: $\textsc{Alg}_t \sim p_t \in \Delta(X)$. It suffices to describe a randomized algorithm by its marginal distribution over states at each time (see, e.g., [142]). That is, suppose $\textsc{Alg}_t$ is distributed according to p_t at each time $t \in [T]$; then the least-cost way for $\textsc{Alg}$ to move from p_{t-1}

to p_t is to couple the two distributions so as to minimize expected movement. Thus consecutive decisions should be distributed jointly according the optimal Wasserstein-1 transportation plan between p_{t-1} and p_t:

$$(\textsc{Alg}_t, \textsc{Alg}_{t-1}) \sim \gamma_t := \underset{\gamma \in \Pi(p_t, p_{t-1})}{\arg\min} \; \mathbb{E}[d(x_t, x_{t-1})],$$

where $(x_t, x_{t-1}) \sim \gamma$ and $\Pi(\mu, \nu)$ is the set of distributions over X^2 with marginals μ and ν. If $\textsc{Alg}$ couples consecutive decisions according to γ_t, then $\mathbb{E}[d(\textsc{Alg}_t, \textsc{Alg}_{t-1})] = \mathbb{W}_X^1(p_t, p_{t-1})$, the Wasserstein-1 distance between p_t and p_{t-1}.[1] Thus, $\mathbb{E}[\mathrm{C}_{\textsc{Alg}}] := \sum_{t=1}^T f_t(p_t) + \mathbb{W}_X^1(p_t, p_{t-1})$, where $f_t(p_t) := \mathbb{E}_{x \sim p_t}[f_t(x)]$.

The *offline optimal* algorithm $\textsc{Opt}$ for an MTS instance chooses the hindsight optimal sequence of decisions:

$$\mathrm{C}_{\textsc{Opt}} := \inf_{x_1, \ldots, x_T \in X} \sum_{t=1}^{T} f_t(x_t) + d(x_t, x_{t-1}).$$

Algorithms for MTS are typically judged by their *competitive ratio*, an adaptive measure of performance against $\textsc{Opt}$ or any other algorithm.

Definition 3.2.1. *A deterministic algorithm $\textsc{Alg}$ is c-**competitive with respect to** another algorithm $\textsc{Alg}'$ if, on any problem instance,*

$$\mathrm{C}_{\textsc{Alg}} \leq c \cdot \mathrm{C}_{\textsc{Alg}'} + b,$$

*where b is independent of the problem instance. If $\textsc{Alg}'$ is $\textsc{Opt}$, we simply say that $\textsc{Alg}$ is c-competitive, or has **competitive ratio** c. If $\textsc{Alg}$ or $\textsc{Alg}'$ are randomized, we replace costs with expected costs in the inequality:*

$$\mathbb{E}[\mathrm{C}_{\textsc{Alg}}] \leq c \cdot \mathbb{E}[\mathrm{C}_{\textsc{Alg}'}] + b.$$

When service cost functions are arbitrary, algorithms for MTS can only be competitive on metric spaces with finite cardinality $|X| = n \in \mathbb{N}$. In this case, the work function algorithm achieves the optimal deterministic competitive ratio of $2n - 1$ [76]. However, randomization can improve performance, with state-of-the-art algorithms achieving competitive ratio $O(\log^2 n)$ [142, 143], which is tight for certain metric spaces [138].

[1] Note that the optimal transportation plan γ_t exists by the assumption made throughout that X is complete and separable.

Consistency, Robustness, and Bicompetitiveness

The competitive ratio quantifies worst-case performance of an online algorithm; its focus on the worst case thus biases algorithm design toward more conservative algorithms. Moreover, as just noted, the competitive ratio of MTS algorithms degrades as $|X|$ grows. In practical applications, however, data on typical problem instances is available, and thus data-driven machine-learned algorithms may significantly outperform traditional competitive algorithms. Since these machine-learned algorithms generally lack worst-case performance guarantees, we seek to design algorithms that exploit the good performance of a machine-learned *advice* algorithm (hereafter, ADV) while maintaining worst-case competitiveness. This motivates the following definitions.

Definition 3.2.2. *Let* ADV *be an advice algorithm. An algorithm* ALG *is* c-***consistent*** *if it is* c*-competitive with respect to* ADV*.* ALG *is said to be* r-***robust*** *if it is* r*-competitive, regardless of the performance of* ADV*.*

Thus, if ADV is a machine-learned algorithm and ALG is c-consistent and r-robust, then ALG achieves performance within a multiplicative factor c of the machine-learned advice while maintaining a worst-case competitive ratio. In this work, we design algorithms with tunable guarantees of robustness and consistency, i.e., that can achieve $(1+\epsilon)$-consistency for any $\epsilon > 0$ while keeping bounded robustness. We approach this by designing *bicompetitive* algorithms, defined as follows.

Definition 3.2.3. *Let* $\textsc{Alg}, \textsc{Alg}', \textsc{Alg}''$ *be three algorithms.* ALG *is* (c, r)-***bicompetitive with respect to*** $(\textsc{Alg}', \textsc{Alg}'')$ *if* ALG *is both* c*-competitive with respect to* $\textsc{Alg}'$ *and* r*-competitive with respect to* $\textsc{Alg}''$*.*

It follows that if ALG is (c, r)-bicompetitive with respect to algorithms $(\textsc{Adv}, \textsc{Rob})$ and ROB is b-competitive, then ALG is c-consistent and rb-robust. Thus to design robust and consistent MTS algorithms, it suffices to design bicompetitive algorithms. We detail prior robustness and consistency results for MTS in Table 3.1.

Special Cases of MTS

We now briefly describe some important special cases of MTS that are of particular relevance for applications to data science and multi-agent planning. Select bounds on competitive ratio, robustness, and consistency from prior work are detailed in Table 3.1.

Convex function chasing. The problem of convex function chasing (CFC), also known as "smoothed" online convex optimization, is an MTS in which the metric space is a finite-dimensional normed vector space and cost functions f_t are restricted to be convex. The best known algorithm for CFC in an arbitrary n-dimensional normed vector space achieves competitive ratio n and improved performance of $O(\min\{n, \sqrt{n \log T}\})$ in the Euclidean setting [106]. On the other hand, any algorithm for CFC in $\mathbb{R}^n$ with the ℓ^p norm has competitive ratio $\Omega(\max\{\sqrt{n}, n^{1-1/p}\})$ [101]. It is straightforward to see by Jensen's inequality and convexity of norms that a c-competitive randomized algorithm for CFC can be derandomized by taking the expectation, yielding a c-competitive deterministic algorithm for CFC [120].

A number of special cases of CFC in which cost functions have additional structure have received attention in the literature. For example, the case where each f_t is the $\{0, +\infty\}$ indicator of a convex set $K_t \subseteq \mathbb{R}^n$ is known as *convex body chasing* and was first considered by Friedman and Linial [77]; the case of well-conditioned f_t was considered by Argue, Gupta, and Guruganesh [108]. The class of α-polyhedral functions has been widely studied as a special case in the CFC literature [102, 103, 144] and is defined as follows.

Definition 3.2.4 (cf. Definition 2.2.2)**.** *Fix $\alpha > 0$ and a normed vector space $(\mathbb{R}^n, \|\cdot\|)$. A function $g : \mathbb{R}^n \to \overline{\mathbb{R}}_+$ is α-**polyhedral** if it has a unique minimizer $\mathbf{v} \in \mathbb{R}^n$, and for all $\mathbf{x} \in \mathbb{R}^n$, $g(\mathbf{x}) \geq g(\mathbf{v}) + \alpha\|\mathbf{x} - \mathbf{v}\|$.*

A simply greedy algorithm obtains competitive ratio $\max\{1, \frac{2}{\alpha}\}$ for CFC with α-polyhedral service costs [103], but better results can be obtained in the Euclidean setting [107].

***k*-server.** In the k-server problem, we control k agents ("servers") residing in the metric space X, and at each time t, we receive a request $r_t \in X$ and must move one of the servers to r_t, paying the distance traveled by the server we moved to meet the request. It is straightforward to see this is an MTS on the metric space $\binom{X}{k}$ (i.e., unordered k-tuples of states in X) endowed with the minimal matching distance inherited from the metric on X. The service cost f_t enforces that one of the servers is located at r_t, so for $\mathbf{x}_t := \{x_t^{(1)}, \ldots, x_t^{(k)}\} \in \binom{X}{k}$, $f_t(\mathbf{x}_t) = \infty \cdot \mathbb{1}_{r_t \notin \mathbf{x}_t}$. The (deterministic) work function algorithm is $(2k-1)$-competitive for k-server on any metric space [97], and no deterministic algorithm can achieve competitive ratio better than k [145]. Significant work has been done establishing tighter bounds on deterministic algorithms in particular metric spaces as well as sublinear bounds for randomized algorithms; see [146] for a survey and [147, 148] for recent results. For brevity, we only invoke the $O(k)$-competitiveness of the work function algorithm in our work.

***k*-chasing convex functions.** The problem of k-chasing convex functions is a generalization of both k-server and CFC: the setting is taken to be a finite-dimensional vector space $(\mathbb{R}^n, \|\cdot\|)$, and we maintain a set of k servers $\mathbf{x}_t := \{\mathbf{x}_t^{(1)}, \ldots, \mathbf{x}_t^{(k)}\} \in \binom{\mathbb{R}^n}{k}$. At time t, an adversary serves a convex cost function $g_t : \mathbb{R}^n \to \overline{\mathbb{R}}_+$, and after moving our servers (by the triangle inequality, it suffices to just move one), we pay the service cost $\min_{i\in[k]} g_t(\mathbf{x}_t^{(i)})$ and the movement cost. Similar to k-server, this is an MTS on the metric space $\binom{\mathbb{R}^n}{k}$ endowed with the minimal matching distance inherited from the norm, with service costs f_t of the form $f_t(\mathbf{x}_t) := \min_{i\in[k]} g_t(\mathbf{x}_t^{(i)})$. This problem was introduced by [135], which found that under suitable structural assumptions on the functions g_t, competitive guarantees from existing k-server algorithms can be translated to k-chasing. However, they obtain randomized algorithms with guarantees dependent on adaptivity of the adversary. For the sake of clarity, in our work we consider k-chasing of convex, *α-polyhedral* functions (Definition 3.2.4), which enable translating deterministic algorithms for k-server into deterministic algorithms for k-chasing. In particular, following the proof of [135, Theorem 3.1], we have the following result, which is proved in Section 3.A.

Proposition 3.2.5. *Let $g_1, \ldots, g_T : \mathbb{R}^n \to \overline{\mathbb{R}}_+$ be an instance of k-chasing convex, α-polyhedral functions. If* Alg *is a deterministic, C-competitive algorithm for k-server, then applying* Alg *to the sequence of minimizers $\mathbf{v}_1, \ldots, \mathbf{v}_T$ of $g_1, \ldots, g_T$ achieves competitive ratio at most $C \max\{1, \frac{2}{\alpha}\}$ for the k-chasing instance.*

It follows that the algorithm feeding the minimizers $v_1, \ldots, v_T$ to the $(2k-1)$-competitive work function algorithm as requests is $(2k-1)\max\{1, \frac{2}{\alpha}\}$-competitive for k-chasing convex α-polyhedral functions.

Applications

Besides its deep connections to online learning [116, 130], MTS and its special cases have numerous applications to problems in operations and data science: many online decision-making problems that penalize switching between decisions can be modeled within this framework. For instance, MTS has been applied to contextual Bayesian optimization with switching costs [149] and energy system operation [132, 133], CFC has applications to smoothed online regression [99, 132] and k-

server and its generalizations have applications to multi-agent planning and logistics [137], dynamic clustering [135, 136], and beyond. To motivate our work designing learning-augmented algorithms for MTS, we briefly detail several applications of MTS and its special cases to decision-making problems in operations and data science.

Energy Resource Operation. Consider the problem of operating a self-scheduled dispatchable generation resource such as a combined cycle power plant or grid-scale energy storage. The goal of the resource operator is to choose dispatch decisions $\mathbf{x}_t \in X$ (such as generator setpoints, charge/discharge decisions, etc.) that maximize profit (equivalently, minimize negative profit) while satisfying operational constraints such as meeting a certain level of electricity demand. For many resources, the operator is faced with two kinds of costs at each timestep t: (1) the instantaneous cost/negative profit $f_t(\mathbf{x}_t)$, which varies with time and depends on such factors as electricity price, fuel price, operational constraints, and weather conditions that influence generation efficiency, and (2) a cost $d(\mathbf{x}_t, \mathbf{x}_{t-1})$ that penalizes switching decisions. The switching cost is an important consideration for multiple kinds of resources, such as conventional thermal generation—for which frequent ramping causes decreased efficiency and increased wear-and-tear [35]—and battery storage—for which large fluctuations in charge/discharge rate can cause degradation [150]. Accounting for this switching cost is thus of increasing importance on the modern grid, with increasing levels of variable renewable generation necessitating more frequent and significant ramp events.

Depending on the structure of the decision space X, the negative profit functions f_t, and the switching cost d, this problem can be cast as either an instance of MTS or some special case such as CFC. However, when the dimension of the decision space is large, traditional competitive algorithms for MTS/CFC may perform poorly for energy resource dispatch, since the competitive ratio grows as a function of dimension. While machine-learned algorithms trained on historical data may perform much better, it is crucial that they come with performance guarantees to ensure reliable and secure operation. This motivates the development of algorithms that can be *consistent* with respect to such ML advice while maintaining worst-case guarantees on *robustness*. We will discuss a more specific application to the setting of cogeneration power plant operation in Section 3.7.

Smoothed Online Clustering. Suppose a decision-maker seeks to cluster a stream of points arriving online and must pay both for the distance between a point and the cluster center it is assigned to, as well as for the movement of cluster centers as they are updated to accommodate new arrivals. When the clustering objective is the k-median objective, this problem is naturally modeled as an instance of k-chasing convex, α-polyhedral functions g_t of the form $g_t(\mathbf{x}) = c_t\|\mathbf{x} - \mathbf{v}_t\|_{\ell^1}$. The decision-maker's server positions $\mathbf{x}_t \in \binom{\mathbb{R}^n}{k}$ encode the k cluster centers, the minimizer $\mathbf{v}_t$ represents the position of the new arrival, and the service cost $f_t(\mathbf{x}_t) = \min_{i\in[k]} g_t(\mathbf{x}_t^{(i)})$ gives the cost of assigning the new arrival to the nearest cluster center. The weight c_t in the cost reflects the tradeoff between movement and service cost, i.e., the tradeoff between stability of the clustering and cost of adding new arrivals to the existing clusters.

When α is small and the number of clusters k is large, traditional competitive algorithms for k-chasing may perform poorly on this smoothed clustering objective, since the competitive ratio scales like $\frac{k}{\alpha}$ (Proposition 3.2.5). Machine-learned algorithms trained on "typical" examples of evolving datasets may perform better in practice, but come with no guarantees, motivating our work on robust and consistent algorithm design.

Smoothed Online Regression. Suppose a learner seeks to fit a sequence of regressors to an evolving dataset without changing the estimated parameters too much as the dataset evolves (e.g., to prevent rapid changes to predictions, which in some applications may impact user experience). This problem is naturally modeled as an instance of CFC where the service cost functions $f_t : \mathbb{R}^n \to \mathbb{R}_+$ encode the loss of the regression objective given the dataset at time t, and the switching cost penalizes changing regressors $\mathbf{x}_t \in \mathbb{R}^n$ at each time. A number of online regression tasks with convex losses can be modeled in this framework, such as Ridge/Lasso regression, logistic regression, and maximum likelihood estimation [99]. However, if the regressor is high-dimensional, then traditional competitive algorithms for CFC may perform very poorly on the smoothed regression objective, since the competitive ratio scales linearly in n. Machine-learned algorithms trained on "typical" examples of evolving datasets may perform better, but again they will not come with guarantees on performance, thus motivating our work on robust and consistent algorithm design.

Algorithm 6: $\textsc{Dart}(\textsc{Adv}, \textsc{Rob}; \epsilon)$

Input: Algorithms $\textsc{Adv}, \textsc{Rob}$; parameter $\epsilon > 0$
Output: Distributions $p_1, \ldots, p_T \in \Delta(X)$ chosen online

1 $\lambda_0 \leftarrow 0$
2 **for** $t = 1, 2, \ldots, T$ **do**
3   Observe f_t, $a_t := \textsc{Adv}_t$, and $r_t := \textsc{Rob}_t$
4   **if** $\mathrm{C}_{\textsc{Rob}}(1, t) \geq \frac{\epsilon}{4} \cdot \mathrm{C}_{\textsc{Adv}}(1, t)$ **then**
5     $\lambda_t \leftarrow 1$
6   **else**
7     $\lambda_t \leftarrow \max \left\{ \lambda_{t-1} - \frac{\frac{\epsilon}{2} \mathrm{C}_{\textsc{Adv}}(t,t) + (1-\lambda_{t-1}) f_t(a_t)}{2d(a_{t-1}, r_{t-1})}, 0 \right\}$
8   $p_t \leftarrow \lambda_t \delta_{a_t} + (1 - \lambda_t) \delta_{r_t}$
9 **end**

3.3 A Bicompetitive Algorithm for Metrical Task Systems

We now present a randomized algorithm, $\textsc{Dart}$ (**D**istance-**A**daptive **R**obust Weight **T**ransport, Algorithm 6), that achieves a bicompetitive guarantee $(1 + \epsilon, 2^{O(1/\epsilon)})$ in expectation with respect to *any* pair of (randomized) MTS algorithms $(\textsc{Adv}, \textsc{Rob})$.

The algorithm works as follows: it maintains a mixing weight $\lambda_t \in [0, 1]$ associated with the decision $a_t := \textsc{Adv}_t$ at each time t. This weight is adaptively updated at each timestep after observing the decisions made by $\textsc{Adv}$ and $\textsc{Rob}$, as well as their relative costs and the distance between the two algorithms' decisions (lines 4-7). $\textsc{Dart}$ then chooses its decision according to the distribution p_t (line 8), which takes value $\textsc{Adv}_t$ with probability λ_t and $\textsc{Rob}_t$ with probability $(1 - \lambda_t)$. For the weight update, there are two cases. First, if $\textsc{Rob}$ has incurred at least an $O(\epsilon)$ fraction of the cost that $\textsc{Adv}$ has (line 4), then the all weight is placed on $\textsc{Adv}$. If this is not the case, λ_t is decreased—that is, weight is shifted from $\textsc{Adv}$ toward $\textsc{Rob}$ in proportion to the ratio between $\textsc{Adv}$'s instantaneous cost and the distance between the two algorithms (line 7). The parameter $\epsilon > 0$ provided as input to $\textsc{Dart}$ governs how closely $\textsc{Dart}$ follows $\textsc{Adv}$, i.e., how much we choose to "trust" the advice. A choice of ϵ that is very small will cause λ_t to stay closer to 1, giving better consistency in exchange for possibly worse robustness. On the other hand, a larger choice of ϵ will cause the weight λ_t to decrease more rapidly toward 0 in line 7, leading $\textsc{Dart}$ to more closely follow $\textsc{Rob}$ and improving worst-case robustness. The specific form of the update rule for λ is designed to ensure that, in the case of line 7, the weight λ_t decreases slowly enough for $\textsc{Dart}$ to maintain performance close to that of $\textsc{Adv}$ (i.e., consistency), yet quickly enough that $\textsc{Dart}$ has bounded performance with respect to $\textsc{Rob}$. Note that, since $\textsc{Dart}$ is a randomized algorithm, we couple consecutive distributions p_{t-1}, p_t according to the optimal (Wasserstein-1) transportation plan, as discussed in Section 3.2.

The following theorem explicitly characterizes the performance of Dart.

Theorem 3.3.1. *Let* Adv, Rob *be any two (possibly randomized) algorithms for* MTS *or a special case thereof. For any chosen* $\epsilon > 0$*, Algorithm 6 (*Dart*) achieves bicompetitiveness* $\left(1+\epsilon, 2^{O(1/\epsilon)}\right)$ *in expectation against* (Adv, Rob)*.*

Our proof, which is presented in Section 3.B, consists of two parts. We first prove competitiveness with respect to Adv via amortized analysis, using the potential function $\mathbb{E}_{x_t \sim p_t}[d(x_t, a_t)]$. We then prove competitiveness with respect to Rob by means of a novel sum argument, upper bounding λ_t in terms of the cost incurred by Adv. This bound explicitly characterizes how much cost Dart can be forced to incur by a "bad" advice algorithm Adv before transferring all of its weight to Rob.

As immediate corollaries to Theorem 3.3.1, we obtain the following upper bounds on robustness and consistency for MTS, CFC, k-server, and k-chasing, which are proved in Section 3.B.

Corollary 3.3.2. *Choose any* $\epsilon > 0$*.*

i. *There is a* $(1+\epsilon)$*-consistent,* $2^{O(1/\epsilon)}O(\log^2(n))$*-robust randomized algorithm for* MTS *on any* n*-point metric space.*
ii. *There is a* $(1+\epsilon)$*-consistent,* $2^{O(1/\epsilon)}n$*-robust deterministic algorithm for* CFC *on any* n*-dimensional normed vector space.*
iii. *There is a* $(1+\epsilon)$*-consistent,* $2^{O(1/\epsilon)}(2k-1)$*-robust randomized algorithm for* k*-server on any metric space.*
iv. *There is a* $(1+\epsilon)$*-consistent,* $2^{O(1/\epsilon)}O(\frac{k}{\alpha})$*-robust randomized algorithm for* k*-chasing convex,* α*-polyhedral functions on any normed vector space.*

We wish to emphasize that the bicompetitive bound in Theorem 3.3.1 is the *first* bicompetitive bound for general MTS that is both independent of metric space diameter and provides bounded competitiveness with respect to Rob for arbitrarily small $\epsilon > 0$. This latter property is of particular significance to practical application since this enables Dart to achieve performance arbitrarily close to that of a black-box ML algorithm for MTS while maintaining a worst-case competitive guarantee, enabling better exploitation of the good ML performance.

In addition, the bound's independence from diameter enables obtaining robustness guarantees on unbounded spaces: Corollary 3.1.1.ii resolves the question of [140] of whether $(1+\epsilon)$-consistency and bounded robustness can be achieved for CFC on unbounded domains with arbitrary $\epsilon > 0$, and Corollaries 3.1.1.iii and iv answer for the first time the analogous question for k-server and k-chasing of convex, α-polyhedral functions. Although most practical problems have finite (but potentially very large) diameter, the robustness bounds given by Dart still improve on the diameter-dependent results of [75] and [140] (Theorem 2.4.5 in Chapter 2) when $\operatorname{diam}(X) = 2^{\omega(1/\epsilon)}$. Moreover, as we will discuss in Section 3.5, Dart achieves even better robustness when the diameter is bounded, matching the dependence of these other results and giving further-improved bounds for the k-server problem.

3.4 Fundamental Limits on Robustness and Consistency

Though the tradeoff between robustness and consistency given by Dart is exponential, it turns out that this is the best that we can hope for from any robust and consistent MTS algorithm. In the following theorem, which is proved in Section 3.C, we present a lower bound on the robustness of *any* $(1+\epsilon)$-consistent randomized MTS algorithm, showing that it must be exponential in $1/\epsilon$.

Theorem 3.4.1. *Let $\epsilon \in (0,1]$. There is an* MTS *instance on a finite metric space (X,d) with $|X| = O(\frac{1}{\epsilon})$ and an adversarial advice algorithm* Adv *such that* any *randomized algorithm achieving $(1+\epsilon)$-consistency with respect to* Adv *is $2^{\Omega(1/\epsilon)}$-robust.*

Since the metric space X in the preceding theorem has cardinality $O(\frac{1}{\epsilon})$, Dart achieves robustness $2^{O(1/\epsilon)}$, by Corollary 3.3.2.i. Thus, Dart yields the *optimal* robustness-consistency tradeoff for general metrical task systems, up to constant factors in the exponent. Moreover, the metric space realizing the lower bound in Theorem 3.4.1 is not a pathological example: it is simply a finite subset of $\mathbb{R}$ with the usual (Euclidean) metric. Further note that this lower bound does not contradict the diameter-dependent upper bound of [75]: the metric space has diameter exponential in $1/\epsilon$, and hence the randomized algorithm of [75] also obtains exponential robustness in this setting.

This exponential lower bound on the tradeoff between robustness and consistency for MTS raises the question of whether improved tradeoffs can be obtained for special cases of MTS where there is added structure. In particular, could the convexity inherent in CFC yield an improved dependence on ϵ in the robustness? In the following theorem, we answer this question in the negative, showing that in certain normed vector spaces the robustness-consistency tradeoff remains exponential.

Theorem 3.4.2. *Let* $\epsilon \in (0, 1]$. *There is a* CFC *instance in* $\mathbb{R}^{O(1/\epsilon)}$ *endowed with a weighted* ℓ^1 *norm, along with an adversarial advice algorithm* Adv, *such that* any *algorithm that is* $(1 + \epsilon)$*-consistent with respect to* Adv *has robustness* $2^{\Omega(1/\epsilon)}$.

We present a proof in Section 3.C; it follows via a reduction to the MTS instance realizing the lower bound of Theorem 3.4.1, using the fact that MTS instances on a tree metric can be isometrically converted into CFC instances in a weighted ℓ^1 space (à la [142]). To the best of our knowledge, our use of this correspondence to obtain lower bounds on the performance of algorithms for CFC is novel.

As Corollary 3.3.2.ii gives a $(1 + \epsilon)$-consistent, $2^{O(1/\epsilon)}$-robust algorithm for CFC in a normed vector space of dimension $O(\frac{1}{\epsilon})$, Dart thus achieves the *optimal* tradeoff between robustness and consistency for CFC in general normed vector spaces, up to constant factors in the exponent. Note that this leaves open the question of whether subexponential robustness can be achieved for CFC under other norms such as the Euclidean norm.

3.5 Breaking the Exponential Robustness Barrier

In Sections 3.3 and 3.4, we saw that Dart achieves $(1+\epsilon, 2^{O(1/\epsilon)})$-bicompetitiveness, and that the resultant tradeoff between robustness and consistency is optimal in general for MTS and CFC. However, prior work has obtained subexponential robustness bounds in certain special cases of MTS, including for CFC and k-server on the real line [113, 141]. In addition, for spaces with diameter bounded by some finite constant D, $(1+\epsilon)$-consistency and $O(\frac{1}{\epsilon})$-robustness can be obtained for CFC in n dimensions with an additive term $O(\frac{D}{\epsilon})$ on the robustness [140], and a similar bound holds for MTS more generally [75].

Given that Theorem 3.3.1 suggests an exponential bicompetitive tradeoff for Dart, it is worth asking whether Dart can perform better on such "easier" problem instances. It turns out that this is the case: we can prove that, in several special cases, Dart achieves $(1 + \epsilon)$-consistency together with robustness that depends only linearly on $\frac{1}{\epsilon}$. Notably, none of these improved bounds require modification of Dart: they simply follow by a refined analysis. We consider three cases in turn.

Bounded Diameter

When the metric space has bounded diameter D, and more generally when the algorithms $\textsc{Adv}$ and $\textsc{Rob}$ are never farther apart than a distance D, $\textsc{Dart}$ achieves bicompetitiveness $(1+\epsilon, O(\frac{1}{\epsilon}))$ with respect to $(\textsc{Adv}, \textsc{Rob})$, with just an additive term of $O(\frac{D}{\epsilon})$ on its competitiveness with respect to $\textsc{Rob}$. This matches the dependence on diameter obtained in prior work [75, 140]. Note that unlike the randomized algorithm of [75], this result does not require advance knowledge of the diameter bound D; it simply results from a specialized analysis in the case that the algorithms $\textsc{Adv}$ and $\textsc{Rob}$ are never further apart than a distance D. We present the formal performance bound in the following theorem.

Theorem 3.5.1. *Let $\textsc{Adv}, \textsc{Rob}$ be any two (possibly randomized) algorithms for* MTS *or a special case thereof. For any chosen $\epsilon > 0$, if $d(\textsc{Adv}_t, \textsc{Rob}_t) \leq D$ for all $t \in [T]$, Algorithm 6 ($\textsc{Dart}$) achieves cost bounded as*

$$\mathrm{C}_{\textsc{Dart}} \leq \min\left\{(1+\epsilon)\mathrm{C}_{\textsc{Adv}}, O\left(\frac{1}{\epsilon}\right)\mathrm{C}_{\textsc{Rob}} + O\left(\frac{D}{\epsilon}\right)\right\}.$$

That is, $\textsc{Dart}$ is $(1+\epsilon, O(1/\epsilon))$-bicompetitive against $(\textsc{Adv}, \textsc{Rob})$, with an additive constant $O(D/\epsilon)$ on its competitive guarantee against $\textsc{Rob}$.

This result is proved in Section 3.D. It is worth emphasizing that this bicompetitive guarantee holds *in addition* to the exponential tradeoff given by Theorem 3.3.1. Thus $\textsc{Dart}$ is $(1+\epsilon)$-competitive with respect to $\textsc{Adv}$ and has cost bounded by $\mathrm{C}_{\textsc{Rob}}$ as

$$\mathrm{C}_{\textsc{Dart}} \leq \min\left\{O\left(\frac{1}{\epsilon}\right)\mathrm{C}_{\textsc{Rob}} + O\left(\frac{D}{\epsilon}\right), 2^{O(1/\epsilon)}\mathrm{C}_{\textsc{Rob}}\right\}.$$

As such, $\textsc{Dart}$ achieves the "best of both worlds" in terms of robustness, regardless of whether D is small or large. This simultaneous bound further extends to all the robustness bounds in Corollary 3.3.2.

***k*-server**

We next consider the k-server problem. In doing so, we restrict $\textsc{Adv}$ and $\textsc{Rob}$ to be *lazy* algorithms for k-server, i.e., algorithms that at any timestep move at most one server, moving none if the current server positions already satisfy the request. This assumption is without loss of generality [151, §10.2.3]. We also make the assumption that all servers begin at the same location; relaxing this assumption only changes the results by a constant additive term.

We first state a lemma relating the distance between any two lazy k-server algorithms to the offline optimal cost; the lemma is proved in Section 3.D.

Lemma 3.5.2. *Let $s_1, \ldots, s_T \in X$ be the request sequence for a k-server instance on the metric space (X, d), and let* Adv *and* Rob *be any two (possibly randomized) k-server algorithms. Further suppose that* Adv *and* Rob *are both lazy, and that all their servers start at the same point $x_0 \in X$. Let $\mathbf{a}_1, \ldots, \mathbf{a}_T \in \binom{X}{k}$ and $\mathbf{r}_1, \ldots, \mathbf{r}_T \in \binom{X}{k}$ be the sequences of server positions of* Adv *and* Rob*, respectively, for the problem instance. Then for any time $t \in [T]$,*

$$d_{\text{mm}}(\mathbf{a}_t, \mathbf{r}_t) \le k \cdot \mathrm{C}_{\textsc{Opt}}(1, t),$$

where d_{mm} is the minimal matching distance inherited from the metric d.

Given any metric space (X, d), Lemma 3.5.2 allows us to bound the diameter of the subset of X that can be occupied by a lazy algorithm for a k-server instance by k times the offline optimal cost on that instance. Substituting this bound into Theorem 3.5.1 and using the fact that the work function algorithm is $(2k - 1)$-competitive, we obtain the following result.

Theorem 3.5.3. *Consider k-server on an arbitrary metric space with all servers starting at some $x_0 \in X$. Let* Adv *be a lazy advice algorithm, and let* Rob *be a lazy version of the work function algorithm. For any $\epsilon > 0$, Algorithm 6 (*Dart*) is $(1 + \epsilon)$-consistent and $O(\frac{k}{\epsilon})$-robust.*

This is the *first* result obtaining $(1 + \epsilon)$-consistency together with robustness linear in $\frac{1}{\epsilon}$ for k-server; in particular, applying the diameter bound from Lemma 3.5.2 to the multiplicative weights algorithm of [75] yields only a bound of $O(k)$ on both robustness and consistency, which is no better than ignoring advice.

k-chasing

Finally, we consider k-chasing of convex, α-polyhedral functions on $\mathbb{R}$. We assume that Rob is a k-server algorithm that operates on the sequence of minimizers $v_1, \ldots, v_T$, e.g., the work function algorithm applied to this sequence, which by Proposition 3.2.5 is $O(\frac{k}{\alpha})$-competitive. Moreover, we assume that Adv and Rob are both *lazy*, meaning that they move at most a single server, and they only move a server if it results in strictly lower service cost. Again, this is without loss of generality. A similar diameter bound to that for k-server yields the following result, which is proved in Section 3.D.

Theorem 3.5.4. *Let* $\textsc{Adv}$ *be a lazy advice algorithm for* k*-chasing convex,* α*-polyhedral functions on* $\mathbb{R}$*, and let* $\textsc{Rob}$ *be a lazy,* $O(\frac{k}{\alpha})$*-competitive algorithm for the problem with the property that, at each time* $t \in [T]$*,* $\textsc{Rob}$ *has a server at the minimizer* v_t *of the current cost function. Suppose* $\textsc{Adv}$ *and* $\textsc{Rob}$ *begin with all servers at the same position* $x_0 \in \mathbb{R}$*. Then* $\textsc{Dart}$ *achieves, for any* $\epsilon > 0$*,* $(1+\epsilon)$*-consistency and* $O(\frac{k}{\alpha\epsilon})$*-robustness.*

3.6 A Deterministic Algorithm under Bounded Diameter

In many real-world applications, it may be desirable to produce decisions deterministically, to avoid potential risk resulting from randomized algorithms like $\textsc{Dart}$, which obtains the optimal tradeoff between robustness and consistency amongst all randomized algorithms, but which might, with some small probability, perform poorly. Despite the fact that $\textsc{Dart}$ can be derandomized in the CFC setting by taking the expectation (Corollary 3.3.2ii), there is no obvious way to do so for the general MTS setting. Motivated by this consideration, in this section we propose a deterministic algorithm, $\textsc{DetRobust}$ML, that deterministically switches between the actions of the advice $\textsc{Adv}$ and a baseline $\textsc{Rob}$. This algorithm, which is specified in Algorithm 7, behaves as follows: it begins by following $\textsc{Adv}$'s decisions, but if $\textsc{Adv}$ surpasses a certain cost threshold and $\textsc{Rob}$ is performing relatively well, $\textsc{DetRobust}$ML will switch to following $\textsc{Rob}$'s decisions (line 8). However, if $\textsc{Rob}$ begins to perform worse relative to $\textsc{Adv}$, then $\textsc{DetRobust}$ML will switch back to following $\textsc{Adv}$ (line 15). The specific thresholds for switching are determined by the parameters $\epsilon, \delta > 0$, which reflect the decision-maker's confidence in $\textsc{Adv}$; they also depend on D, the maximum distance between $\textsc{Adv}$ and $\textsc{Rob}$ at any time during the problem instance. When ϵ and δ are small (close to 0), $\textsc{DetRobust}$ML will spend more time following the decisions of $\textsc{Adv}$, and the threshold for switching to $\textsc{Rob}$ will be more stringent. On the other hand, when ϵ and δ are large, $\textsc{DetRobust}$ML will spend more of its time following the decisions of $\textsc{Rob}$. Tuning ϵ and δ thus allows for trading off between robustness and consistency.

In the following theorem, which is proved in Section 3.E, we present an analytic performance bound on $\textsc{DetRobust}$ML.

Theorem 3.6.1. *Let* D *be an upper bound on the distance betweeen* $\textsc{Adv}$ *and* $\textsc{Rob}$ *for all time, i.e.,* $d(\textsc{Adv}_t, \textsc{Rob}_t) \le D$ *for all* $t \in [T]$*. Then for* $\epsilon, \delta > 0$*, the algorithm* $\textsc{DetRobustML}(\epsilon, \delta, D)$ *(Algorithm 7) achieves cost bounded as*

$$\mathrm{C}_{\textsc{DetRobustML}} \le \min\left\{(1+\epsilon+\delta)\mathrm{C}_{\textsc{Adv}}, \left(1+\frac{1+\epsilon}{\delta}\right)\mathrm{C}_{\textsc{Rob}} + \left(1+\frac{2}{\epsilon}\right)D\right\}.$$

Algorithm 7: The algorithm DetRobustML(ϵ, δ, D). Note that all algorithms are assumed to begin in the same initial state x_0, so $a_0 = r_0 = x_0$ (where $a_0 = \text{Adv}_0$ and $r_0 = \text{Rob}_0$).

Input: Algorithms Adv, Rob; hyperparameters $\epsilon, \delta > 0$, space diameter D
Output: Decisions $x_1, \ldots, x_T$ chosen online

1 $s \leftarrow 1$
2 $x_1 \leftarrow a_1 := \text{Adv}_1$
3 **for** $t = 2, 3, \ldots, T$ **do**
4   Observe f_t, $a_t := \text{Adv}_t$, and $r_t := \text{Rob}_t$
5   **if** $x_{t-1} = a_{t-1}$ **then** // Case where the algorithm coincides with Adv_{t-1}
6   **if** $C_{\text{Adv}}(s,t) \geq \frac{2D}{\epsilon}$ and $C_{\text{Rob}}(1,t) < \delta \cdot C_{\text{Adv}}(1,t)$ **then**
7       $s \leftarrow t + 1$
8       $x_t \leftarrow r_t$
9   **else**
10       $x_t \leftarrow a_t$
11   **else** // Case where the algorithm coincides with Rob_{t-1}
12   **if** $C_{\text{Rob}}(1,t) < \delta \cdot C_{\text{Adv}}(1,t)$ **then**
13       $x_t \leftarrow r_t$
14   **else**
15       $x_t \leftarrow a_t$
16 **end**

In particular, when $\delta = \epsilon$, DetRobustML is $(1 + 2\epsilon, O(1/\epsilon))$-bicompetitive against (Adv, Rob), with an additive constant of $O(\frac{D}{\epsilon})$ on its competitive guarantee against Rob.

Two remarks are in order. First, note that the performance bound achieved by DetRobustML—including both the bicompetitive bound and the distance-dependent additive term—asymptotically matches the one obtained by Dart in the distance-bounded setting (Theorem 3.5.1). There is one primary difference between the results: DetRobustML requires *prior knowledge* of the distance bound D, which is used in the cost threshold in line 6; in contrast, Dart does not require knowledge of this distance bound. As such, Dart's performance bound will be significantly less conservative than that of DetRobustML in settings where the decision space has a large diameter but Adv and Rob remain close to each other, or when D is large but the cost incurred by Rob is small. Moreover, it is worth noting that Dart maintains its distance-independent bound (Theorem 3.3.1) even if D is unbounded, whereas DetRobustML *requires* a finite D to obtain a nontrivial performance bound with respect to Rob.

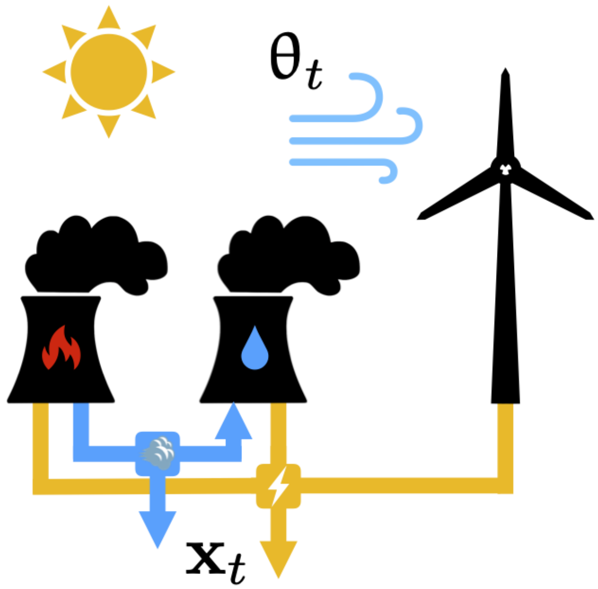


Figure 3.1: A schematic of the cogeneration power plant model. The plant operator chooses how much electricity (yellow arrow) and steam (blue arrow) the three gas turbines (left cooling tower) produce, as well as how much steam is directed to the steam turbine (right cooling tower) to produce additional electricity. At each time t, ambient conditions (e.g., temperature) together with electricity and steam demand are represented by the vector θ_t, and electricity and steam dispatch decisions are represented by the vector $\mathbf{x}_t$.

Second, recall that in Theorem 2.3.3 of the previous chapter, we proved a lower bound showing that any deterministic switching algorithm that achieves finite robustness must have consistency at least 3. At first glance, this might seem to conflict with the above bound for DetRobustML, which is a switching algorithm. However, this is not the case, as the lower bound construction relies on allowing Adv and Rob to be arbitrarily far apart, so D has no *a priori* bound. As such, while there is a fundamental limit on the robustness and consistency achievable for switching algorithms—and thus, in the MTS setting, deterministic algorithms—in general, DetRobustML demonstrates that this is not the case for the bounded diameter setting.

3.7 An Application to Cogeneration Power Plant Operation

In this section, we evaluate the performance of our algorithms, Dart and DetRobustML, on a realistic model of combined cycle cogeneration power plant operation under increasing penetrations of variable renewable energy generation. We use an adapted form of the CogenEnv environment from SustainGym, a reinforcement learning benchmarks suite [59]; specifically, our experimental setup builds on the code available at https://zenodo.org/records/13623809 to include experiments involving Dart and the multiplicative weights algorithm of [75].

Model

Consider the problem of operating a combined cycle cogeneration power plant to meet both electricity and steam demand in the presence of exogenous variable renewable generation; see Figure 3.1 for a schematic. Specifically, we consider a plant with three gas turbines and a single steam turbine; we index the gas and steam turbines as $\{1, 2, 3\}$ and $\{4\}$, respectively. At each time $t \in [T]$, representing every 15 minutes over the course of a 24 hour period, the plant operator observes an electricity demand (net of renewables) $d_t \in \mathbb{R}_+$ and a steam demand $q_t \in \mathbb{R}_+$. In response, they choose energy dispatch setpoints $p_t^{(i)} \in [\underline{p}_i, \bar{p}_i]$ and steam dispatch setpoints $s_t^{(i)} \in [\underline{s}_i, \bar{s}_i]$ for all the turbines $i = 1, \dots, 4$, where $\underline{p}_i, \bar{p}_i$ are the lower and upper bounds, respectively, for the energy dispatch, and $\underline{s}_i, \bar{s}_i$ are the lower and upper bounds, respectively, for the steam dispatch. Note that while the gas turbines $i = 1, 2, 3$ produce steam, and thus have positive steam dispatches, the steam turbine $i = 4$ consumes steam, so $s_t^{(4)} < 0$. The plant operator's goal is to minimize the total cost of its dispatch decisions (its hitting cost) and its cost for ramping electricity generation (its switching cost) while producing sufficient electricity and steam to meet demand. Formally, the plant operator faces the following constrained minimization problem:

$$\min_{\substack{\mathbf{p}_1,\dots,\mathbf{p}_T \in [\underline{\mathbf{p}}, \bar{\mathbf{p}}] \\ \mathbf{s}_1,\dots,\mathbf{s}_T \in [\underline{\mathbf{s}}, \bar{\mathbf{s}}]}} \quad \sum_{t=1}^{T} f(\mathbf{p}_t, \mathbf{s}_t; \theta_t) + \alpha \|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1 \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{i=1}^{4} p_t^{(i)} = d_t \qquad \text{for all } t = 1, \dots, T \tag{3.1b}$$

$$\sum_{i=1}^{4} s_t^{(i)} = q_t \qquad \text{for all } t = 1, \dots, T, \tag{3.1c}$$

where $\mathbf{p}_t = (p_t^{(i)})_{i=1}^4$, $\mathbf{s}_t = (s_t^{(i)})_{i=1}^4$, f is a per-round fuel cost function that depends on ambient conditions such as temperature, pressure, and humidity, θ_t is a vector containing these ambient conditions for time t, and α is a parameter determining the magnitude of the switching cost—i.e., the extent to which ramping energy generation is penalized. Importantly, the demands d_t, q_t and the cost function's parameters θ_t are not all known in advance, so the plant operator cannot solve Problem 3.1 all at once. Instead, the plant operator only knows the current timestep's problem parameters θ_t, d_t, and q_t exactly, though they may have (possibly noisy) access to predictions of these parameters in a short lookahead window.

In our experiments, we utilize the cogeneration power plant fuel cost function and associated data on ambient conditions, electricity demand, and steam demand from the CogenEnv environment in SustainGym [59]. This environment models the fuel cost of the cogeneration power plant in a black-box fashion via neural networks, due to the complexity of the physical model of the system; thus, the fuel cost is nonconvex in general. We use a ramp constant of $\alpha = 2$, and in our experiments evaluating the impact of increased renewable energy generation, we utilize wind data obtained from the Wind Integration National Dataset Toolkit [152].

This cogeneration dispatch problem can be easily seen as an instance of the more general MTS problem, where the hitting cost is the fuel cost $f(\cdot;\theta_t)$ (with added penalties for the constraints (3.1b) and (3.1c)) and the switching cost is the ramp cost.[2] In the rest of this section, we will absorb the electricity and steam demands d_t, q_t into the vector θ_t, and we will write the objective of Problem (3.1) abstractly as

$$\sum_{t=1}^{T} f(\mathbf{x}_t, \theta_t) + d(\mathbf{x}_t, \mathbf{x}_{t-1}),$$

where $\mathbf{x}_t$ is the vector of all decisions and d is the ramp cost.

Because this problem is *online*, the decision-maker only has access to the parameters $\theta_1, \ldots, \theta_t$ that have been revealed through time t when making the decision $\mathbf{x}_t$. However, we will assume that the decision-maker has access to (possibly inaccurate) forecasts $\hat{\theta}_{t+1|t}, \ldots, \hat{\theta}_{t+w|t}$ of parameters within a lookahead window of length $w \in \mathbb{N}$, which can help them anticipate and reduce future ramp costs. Such forecasts could be obtained using standard ML methods for predicting near-term weather or energy demand.

Advice and Robust Algorithms

In our experiments, we seek to evaluate the performance of DART and DETROBUSTML when combining two different strategies ADV and ROB for the cogeneration dispatch problem. In the following, we briefly describe the strategies we employ to this end.

[2]Note that while in (3.1) only ramping energy decisions yields a ramp cost, in the general MTS setting, the switching cost must be a metric, so changing any decision must result in a switching cost. However, we can model this by simply making the ramp cost for the steam decisions arbitrarily small.

Our robust baseline algorithm, Rob, will be a myopic, greedy algorithm that simply chooses the decision $\mathbf{x}_t$ that minimizes $f(\cdot;\theta_t)$ at each time t. This algorithm, which we call Greedy, resembles the single-stage dispatch algorithm widely used by power system operators, and it also has worst-case cost guarantees under mild assumptions on the structure of the cost function f such as α-polyhedrality [103]. Its behavior is characterized formally as follows:

$$\textsc{Greedy} : \theta_t \mapsto \arg\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x};\theta_t) =: \mathbf{x}_t.$$

That is, Greedy can be viewed as a function that, when provided with the current parameter vector θ_t, returns the minimizer of $f(\cdot;\theta_t)$ as a dispatch decision. Since we assume that $f(\cdot;\theta_t)$ penalizes all constraints from the original problem (3.1), Greedy will satisfy these constraints. Note that, since f is nonconvex (due to its being parametrized as a neural network), we may not be able to solve for the minimizer of $f(\cdot;\theta_t)$ exactly; in practice, we will employ sequential least-squares programming in SciPy to solve for Greedy's decision [153].

For our advice algorithm, Adv, we train a neural network model to approximate the behavior of model predictive control (MPC) in an unsupervised fashion. That is, at time t, given the known current parameter vector θ_t and forecasts $\hat{\theta}_{t+1|t},\ldots,\hat{\theta}_{t+w|t}$ of parameters over the next $w = 6$ timesteps, we want our model, which we call ML, to choose decisions $\mathbf{x}_t, \mathbf{x}_{t+1},\ldots,\mathbf{x}_{t+w}$ that minimize the lookahead objective

$$f(\mathbf{x}_t;\theta_t) + \sum_{\tau=t+1}^{t+w} f(\mathbf{x}_\tau;\hat{\theta}_{t+\tau|t}) + d(\mathbf{x}_\tau,\mathbf{x}_{\tau-1}). \tag{3.2}$$

At each time t, ML will output recommend decisions $\mathbf{x}_t,\ldots,\mathbf{x}_{t+w}$, but only the decision $\mathbf{x}_t$ will be binding; that is, we will implement *only* the first decision $\mathbf{x}_t$, and then at time $t+1$, we will repeat this process with a new window of predictions to choose $\mathbf{x}_{t+1}$. To train ML, we construct a dataset $\mathcal{D} = \{\mathbf{\Theta}_i\}_{i=1}^N$ of 100 days' worth of $(w+1)$-length windows of cost function parameters (i.e., each $\mathbf{\Theta}_i = (\theta_t, \hat{\theta}_{t+1|t},\ldots,\hat{\theta}_{t+w|t})$ for some t). For the purposes of training, we construct this dataset assuming perfect predictions, i.e., $\hat{\theta}_{t+i|t} = \theta_{t+i}$. We then train ML to take as input a parameter window $\mathbf{\Theta}_i$ and output decisions $\mathbf{x}_t,\ldots,\mathbf{x}_{t+w}$ using (3.2) as the loss function; because the fuel cost f is itself modeled as a neural network, this loss is differentiable, and thus standard stochastic gradient descent methods can be applied to train our model. To ensure that ML produces decisions that respect the plant capacity limits and supply-demand balance constraints for electricity (3.1b) and steam (3.1c), we employ the method of [56], including a gauge map on the output layer of the neural network to enforce constraint satisfaction.

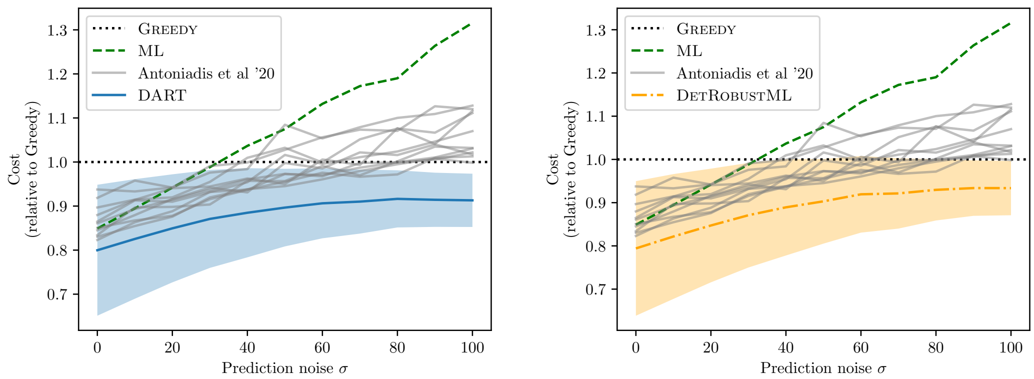
Note that, despite the fact that this unsupervised training approach may yield low empirical error on the training set and the gauge map ensures constraint satisfaction, this does not guarantee that the ML algorithm will perform well on out-of-sample instances or under distribution shift on the parameter forecasts. This motivates combining ML with Greedy via the learning-augmented algorithms we designed earlier in this chapter, to ensure that even if ML is not performing well, we still have worst-case performance guarantees with respect to our Greedy baseline.

Finally, we remark that, while the objective (3.2) resembles that of a standard MPC problem, MPC with even a moderate lookahead window w is computationally prohibitive to run in this setting due to the nonconvexity of the fuel cost functions $f(\cdot, \theta_t)$ and the coupling across time through the switching costs. For a more detailed discussion of these considerations and a comparison of the time complexity of MPC against our ML approach, see the full paper [133].

Experimental Results

In this section, we evaluate the performance of Dart and DetRobustML on the cogeneration power plant operation problem when provided input algorithms Adv = ML and Rob = Greedy as described in the previous section. We begin by exploring the impact of poor machine learning performance due to a distribution shift in prediction noise. That is, we compare the setting of perfect predictions ($\hat{\theta}_{t+1|t} = \theta_{t+1}, \ldots, \hat{\theta}_{t+w|t} = \theta_{t+w}$) to the case of predictions with added i.i.d. Gaussian noise and increasing standard deviation σ (i.e., $\hat{\theta}_{t+1|t} = \theta_{t+1} + \mathbf{z}_{t+1|t}, \ldots, \hat{\theta}_{t+w|t} = \theta_{t+w} + \mathbf{z}_{t+w|t}$, with $\mathbf{z}_{\tau|t} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$). Note that these noisy predictions will not impact the performance of Greedy, which does not use lookahead; it will only impact the performance of ML and the algorithms that use it.

We evaluate the performance of ML, Greedy, Dart (with $\epsilon = 4.4$), and DetRobustML (with $\epsilon = \delta = 1.05$) under noise magnitude σ ranging between 0 and 100. Algorithm hyperparameters were selected to yield the best tradeoff between performance under the low noise ($\sigma = 0$) and high noise ($\sigma = 100$) regimes. We also evaluate the performance of the randomized multiplicative weights algorithm MIN^{rand} of Antoniadis et al. [75] for a range of hyperparameters ϵ between



(a) (b)

Figure 3.2: Cost (normalized by Greedy's) of Dart (a) and DetRobustML (b) compared against ML, Greedy, and the randomized algorithm of Antoniadis et al. [75] (evaluated across several hyperparameter choices) under increasing noise σ on the lookahead predictions. Curves indicate mean normalized cost; shaded regions (provided for Dart and DetRobustML) cover ± one standard deviation.

0 and 1, the range over which their theoretical performance guarantee is valid.[3] We display the resulting costs, normalized by those of Greedy, in Figure 3.2. Remarkably, despite the fact that ML's performance degrades as σ increases, we find that both Dart and DetRobustML gracefully transition between the good performance of ML for small σ to beating the performance of Greedy in the large σ regime. In particular, both Dart and DetRobustML *uniformly improve upon both Greedy and ML on average*, regardless of prediction noise, even though the quality of predictions is unknown to the algorithm *a priori*. In general, the performance of Dart appears to essentially match that of DetRobustML, suggesting that in this specific application, there is not a significant benefit to the improved theoretical guarantees Dart obtains. Despite the fact that it is randomized, though, Dart appears to obtain slightly better performance and less variance in cost than DetRobustML in the large σ regime. In contrast, while the algorithm of Antoniadis et al. takes advantage of good ML performance when σ is small, its performance degrades as σ increases. Thus, Dart and DetRobustML, which gracefully trade off between good ML performance and good robust performance under high σ, uniformly outperform the algorithm of Antoniadis et al.

We next examine the performance of the algorithms Dart and DetRobustML (with unchanged parameters) under increasing penetration of wind energy between 0 MW

[3] Recall that the randomized algorithm of Antoniadis et al. [75] obtains expected cost bounded by $(1+\epsilon)\cdot\min\{C_{\mathrm{ML}}, C_{\mathrm{Greedy}}\} + O(\frac{D}{\epsilon})$.

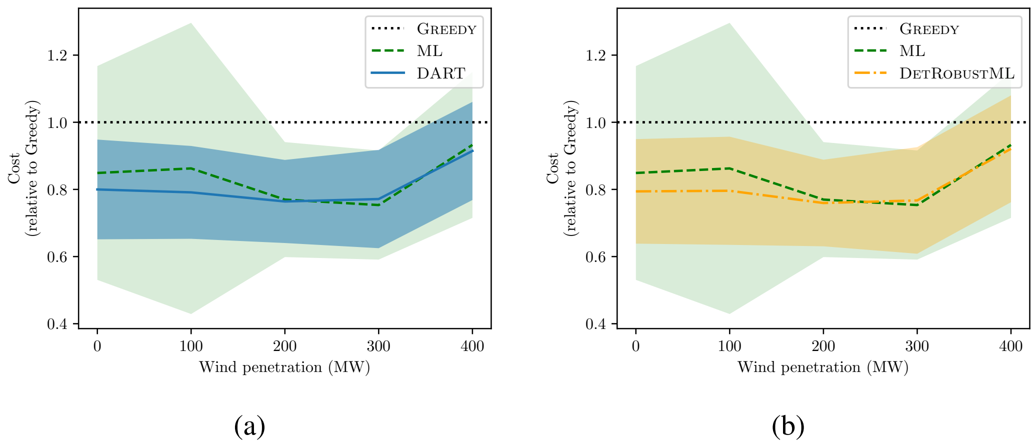


Figure 3.3: Cost (normalized by GREEDY's) of DART (a) and DETROBUSTML (b) compared against ML and GREEDY under increasing wind penetration. Curves indicate mean normalized cost; shaded regions (provided for ML, DART, and DETROBUSTML) cover ± one standard deviation.

and 400 MW (reflecting roughly a $2/3$ fraction of peak demand), displaying the results in Figure 3.3. We find that the efficiency improvement of ML over GREEDY widens for moderate wind penetration (up to 300 MW), highlighting the value of using machine-learned algorithms with lookahead to increase efficiency when variable renewable generation necessitates more frequent ramping. Moreover, in this regime, both DART and DETROBUSTML, though close in average performance to ML, offer significantly lower variance in cost compared to ML. Thus, these learning-augmented strategies enable not just improved performance through machine learning, but also more reliable performance in cases where machine learning fails.

Notably, it appears that ML's performance suffers and gets closer to that of GREEDY when wind penetration nears 400 MW, possibly reflecting the fundamental challenge of adequately anticipating and mitigating future ramp needs in a high-renewables environment, even when equipped with lookahead. Nonetheless, ML still delivers a modest improvement in cost even in this high-renewables regime, which is matched by DART and DETROBUSTML, which continue to provide the best tradeoff between performance and reliability.

3.8 Concluding Remarks

In this chapter, we examine the problem of designing learning-augmented algorithms for MTS and its special cases. Our main algorithm, DART, achieves $(1+\epsilon)$-consistency and robustness exponential in $\frac{1}{\epsilon}$ for MTS and its special cases, which we show is tight for both MTS and for CFC with a certain weighted ℓ^1 norm. We further show that DART achieves improved performance, matching known results, when the diameter of the problem instance is bounded, and improves upon the best known bounds on robustness and consistency for k-server on any metric space and for k-chasing on the line. We also propose a deterministic algorithm, DETROBUSTML, that, given an *a priori* bound on the diameter of the decision space, obtains performance comparable to that of DART. We evaluate our algorithms on a realistic model of cogeneration power plant operation, where they exhibit an ability to bridge the good performance of machine-learned algorithms with the reliability of standard algorithms.

Several interesting avenues remain open for study. Specifically, (i) can subexponential robustness be achieved for CFC and k-chasing with "nicer" norms, e.g., in the Euclidean setting, (ii) can matching lower bounds be obtained on robustness and consistency for the k-server problem, and (iii) is it possible to design learning-augmented algorithms for online optimization problems with more general, constrained structure by extending the ideas developed in this work? This last question will be the focus of Chapter 5.

Appendix

In these appendix sections, we present proofs of the theoretical results in the main body of the chapter.

3.A Proof of Proposition 3.2.5

Let $g_1, \dots, g_T$ be the sequence of α-polyhedral cost functions for an instance of k-chasing, and let $\mathbf{v}_1, \dots, \mathbf{v}_T \in \mathbb{R}^n$ be their minimizers. Let OPT_s be the offline optimal algorithm for the k-server instance on $\mathbb{R}^n$ with requests $\mathbf{v}_1, \dots, \mathbf{v}_T$, and let OPT_c be the offline optimal algorithm for the k-chasing instance on $\mathbb{R}^n$ with function requests $g_1, \dots, g_T$. We denote by $\mathrm{C}^s_{\text{OPT}_s}$ the cost of OPT_s as a k-server algorithm (i.e., ignoring the service costs), and by $\mathrm{C}^c_{\text{OPT}_s}$ the cost of OPT_s as a k-chasing algorithm (including the hitting costs); we use similar notation for ALG. Note that the cost of a k-server algorithm applied to the minimizers of the k-chasing instance will simply be the cost of the k-server algorithm (i.e., the total movement cost incurred by the servers) plus the sum of minimizer costs $\sum_{t=1}^T g_t(\mathbf{v}_t)$, since the minimizer will be occupied by a server at each time. Thus,

$$\mathrm{C}^c_{\text{ALG}} = \mathrm{C}^s_{\text{ALG}} + \sum_{t=1}^{T} g_t(\mathbf{v}_t),$$

and the same holds for $\textsc{Opt}_s$.

Let $\mathbf{o}_1, \ldots, \mathbf{o}_T \in \binom{\mathbb{R}^n}{k}$ be the sequence of server positions of the algorithm $\textsc{Opt}_c$, and let $i_t \in [k]$ denote the server of $\textsc{Opt}_c$ that realizes the binding service cost at time t, i.e., $i_t \coloneqq \arg\min_{i \in [k]} g_t(\mathbf{o}_t^{(i)})$. Thus $\mathrm{C}_{\textsc{Opt}_c} = \sum_{t=1}^{T} g_t(\mathbf{o}_t^{(i_t)}) + d(\mathbf{o}_t, \mathbf{o}_{t-1})$, where d is the minimal matching distance inherited from the norm $\|\cdot\|$. Define the offline algorithm $\textsc{Opt}'_c$ that acts like $\textsc{Opt}_c$, except that at time t it moves the server i_t from $\mathbf{o}_t^{(i_t)}$ to the minimizer $\mathbf{v}_t$, and moves it back to $\mathbf{o}_t^{(i_t)}$ at time $t+1$ before any other server is moved. Since the costs g_t are α-polyhedral,

$$
\begin{aligned}
\mathrm{C}_{\textsc{Opt}'_c} &= \sum_{t=1}^{T} g_t(\mathbf{v}_t) + 2\|\mathbf{o}_t^{(i_t)} - \mathbf{v}_t\| + d(\mathbf{o}_t, \mathbf{o}_{t-1}) \\
&\le \sum_{t=1}^{T} \max\left\{1, \frac{2}{\alpha}\right\} \left(g_t(\mathbf{v}_t) + \alpha\|\mathbf{o}_t^{(i_t)} - \mathbf{v}_t\|\right) + d(\mathbf{o}_t, \mathbf{o}_{t-1}) \\
&\le \max\left\{1, \frac{2}{\alpha}\right\} \sum_{t=1}^{T} g_t(\mathbf{o}_t^{(i_t)}) + d(\mathbf{o}_t, \mathbf{o}_{t-1}) \\
&= \max\left\{1, \frac{2}{\alpha}\right\} \mathrm{C}_{\textsc{Opt}_c}. \qquad (3.3)
\end{aligned}
$$

Moreover, since $\textsc{Opt}'_c$ always has a server at $\mathbf{v}_t$ at time t, it is a feasible k-server algorithm for the request sequence $\mathbf{v}_1, \ldots, \mathbf{v}_T$, and its cost as a k-server algorithm is simply its total cost as a k-chasing algorithm, minus the the sum of minimizer service costs. Thus we have

$$
\begin{aligned}
\mathrm{C}^c_{\textsc{Alg}} &= \mathrm{C}^s_{\textsc{Alg}} + \sum_{t=1}^{T} g_t(\mathbf{v}_t) \\
&\le C \cdot \mathrm{C}^s_{\textsc{Opt}_s} + \sum_{t=1}^{T} g_t(\mathbf{v}_t) \qquad (3.4) \\
&\le C \cdot \mathrm{C}_{\textsc{Opt}'_c} \qquad (3.5) \\
&\le C \cdot \max\left\{1, \frac{2}{\alpha}\right\} \mathrm{C}_{\textsc{Opt}_c}, \qquad (3.6)
\end{aligned}
$$

where (3.4) follows by C-competitiveness of $\textsc{Alg}$ for k-server, (3.5) follows from the fact that $\textsc{Opt}_s$ is optimal and $\textsc{Opt}'_c$ is feasible for the k-server instance $\mathbf{v}_1, \ldots, \mathbf{v}_T$, and (3.6) follows (3.3). Thus $\textsc{Alg}$ is $C \cdot \max\left\{1, \frac{2}{\alpha}\right\}$-competitive for the k-chasing instance. $\square$

3.B Proofs for Section 3.3

Proof of Theorem 3.3.1

It suffices to prove the bicompetitive bound in the case that Adv and Rob are deterministic algorithms. That is, we prove the following bound on Dart's expected cost:

$$\mathbb{E}[\mathrm{C}_{\textsc{Dart}}] \leq \min\{(1+\epsilon)\mathrm{C}_{\textsc{Adv}}, 2^{O(1/\epsilon)}\mathrm{C}_{\textsc{Rob}}\}, \tag{3.7}$$

where the expectation is over the randomness of Dart. The result in its full generality, i.e., when Adv and Rob can be randomized algorithms, follows by the observation that (3.7) establishes the same bound on the expected cost of Dart *conditioned* on a particular pair of realized trajectories $(a_1, \dots, a_T)$, $(r_1, \dots, r_T)$ of Adv and Rob:

$$\mathbb{E}[\mathrm{C}_{\textsc{Dart}} | a_1, \dots, a_T; r_1, \dots, r_T] \leq \min\{(1+\epsilon)\mathrm{C}_{\textsc{Adv}}, 2^{O(1/\epsilon)}\mathrm{C}_{\textsc{Rob}}\},$$

where the expectation is now over the randomness of Dart, Adv, and Rob. With this inequality established, the desired result follows immediately by taking the expectation over the behavior of Adv and Rob on both sides and applying the law of total expectation.

In the following, we thus assume that Adv and Rob are deterministic, with decision trajectories $a_1, \dots, a_T$ and $r_1, \dots, r_T$, respectively. All expectations are over the decisions $x_1, \dots, x_T$ made by Dart, which are each distributed marginally according to $x_t \sim p_t$, with consecutive distributions jointly distributed according to the optimal transportation plan γ_t between p_{t-1} and p_t.

We begin by proving competitiveness with respect to Adv, i.e., *consistency* of Dart. The argument takes the form of a potential function argument, with potential function $\phi_t = \mathbb{E}[d(x_t, a_t)] = (1-\lambda_t)d(r_t, a_t)$. For an arbitrary time t, there are two cases.

(1) Suppose the algorithm follows the case in line 4; then $\lambda_t = 1$, so $x_t = a_t$. Then

$$\begin{aligned}
&\mathbb{E}[f_t(x_t) + d(x_t, x_{t-1}) + \phi_t - \phi_{t-1}] \\
&= f_t(a_t) + \lambda_{t-1} d(a_t, a_{t-1}) + (1-\lambda_{t-1}) d(a_t, r_{t-1}) \\
&\qquad + (1-\lambda_t) d(r_t, a_t) - (1-\lambda_{t-1}) d(r_{t-1}, a_{t-1}) \\
&\leq f_t(a_t) + d(a_t, a_{t-1})
\end{aligned} \tag{3.8}$$

where (3.8) follows from the triangle inequality applied to $d(a_t, r_{t-1})$.

(2) Suppose the algorithm follows the case in line 6. First, note that since the coupling between x_{t-1} and x_t is done via the optimal transport plan between p_{t-1} and p_t, we can upper bound $\mathbb{E}[d(x_t, x_{t-1})]$ by the expected movement cost under *any* transport plan between p_{t-1} and p_t. In particular, we can use the transport plan in which we first send a probability mass of $\min\left\{\frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})}, \lambda_{t-1}\right\}$ from a_{t-1} to r_{t-1}, resulting in a mass of λ_t at a_{t-1} and of $(1-\lambda_t)$ at r_{t-1}, followed by sending the entire mass at a_{t-1} to a_t and the entire mass at r_{t-1} to r_t. Upper bounding $\mathbb{E}[d(x_t, x_{t-1})]$ with this transportation plan, we find:

$$\begin{aligned}\mathbb{E}[d(x_t, x_{t-1})] &\leq (1-\lambda_t)d(r_t, r_{t-1}) + \lambda_t d(a_t, a_{t-1}) \\ &\quad + \min\left\{\frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1}, r_{t-1})}, \lambda_{t-1}\right\} d(a_{t-1}, r_{t-1}) \\ &\leq (1-\lambda_t)d(r_t, r_{t-1}) + \lambda_t d(a_t, a_{t-1}) \\ &\quad + \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2}. \end{aligned} \tag{3.9}$$

Second, note that

$$\begin{aligned}(1-\lambda_t)d(r_t, a_t) &\leq (1-\lambda_t)(d(r_t, r_{t-1}) + d(a_t, a_{t-1})) \\ &\quad + \left(1 - \lambda_{t-1} + \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1}, r_{t-1})}\right) d(a_{t-1}, r_{t-1}) \end{aligned} \tag{3.10}$$

$$\begin{aligned}&\leq (1-\lambda_t)(d(r_t, r_{t-1}) + d(a_t, a_{t-1})) \\ &\quad + (1-\lambda_{t-1})d(a_{t-1}, r_{t-1}) + \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2} \end{aligned} \tag{3.11}$$

where (3.10) follows from the triangle inequality and line 7 of the algorithm. Then, by (3.9) and (3.11), and noting that $\lambda_t \leq \lambda_{t-1}$ in this case, we have

$$\begin{aligned}&\mathbb{E}[f_t(x_t) + d(x_t, x_{t-1}) + \phi_t - \phi_{t-1}] \\ &= \lambda_t f_t(a_t) + (1-\lambda_t)f_t(r_t) + \mathbb{E}[d(x_t, x_{t-1})] \\ &\quad + (1-\lambda_t)d(r_t, a_t) - (1-\lambda_{t-1})d(r_{t-1}, a_{t-1}) \\ &\leq \left(1 + \frac{\epsilon}{2}\right)(f_t(a_t) + d(a_t, a_{t-1})) + 2(f_t(r_t) + d(r_t, r_{t-1})).\end{aligned}$$

Summing cases 1 and 2 over time and using the fact that case 2 only occurs in timesteps t where $\mathrm{C}_{\mathrm{ROB}}(1,t) < \frac{\epsilon}{4}\mathrm{C}_{\mathrm{ADV}}(1,t)$ we obtain

$$\mathbb{E}[\mathrm{C}_{\mathrm{DART}}] \leq (1+\epsilon)\mathrm{C}_{\mathrm{ADV}}.$$

We now turn to proving the competitive bound with respect to $\textsc{Rob}$, i.e., *robustness*. Let $\tau \in \{0, \ldots, T\}$ be the last time index that $\mathrm{C}_{\textsc{Rob}}(1,\tau) \geq \frac{\epsilon}{4} \cdot \mathrm{C}_{\textsc{Adv}}(1,\tau)$. Clearly if $\tau = 0$, then $\lambda_t = 0$ for all $t \in [T]$, so $\textsc{Dart}$ follows $\textsc{Rob}$ exactly and we are finished. Thus we restrict to the case that $\tau \geq 1$, i.e., $\lambda_t > 0$ for *some* time $t \in [T]$. By the consistency result just presented, we have

$$\begin{aligned}
\mathbb{E}[\mathrm{C}_{\textsc{Dart}}] &= \mathbb{E}[\mathrm{C}_{\textsc{Dart}}(1,\tau)] + \mathbb{E}[\mathrm{C}_{\textsc{Dart}}(\tau+1,T)] \\
&\leq (1+\epsilon)\mathrm{C}_{\textsc{Adv}} + \mathbb{E}[\mathrm{C}_{\textsc{Dart}}(\tau+1,T)] \\
&\leq \frac{4(1+\epsilon)}{\epsilon}\mathrm{C}_{\textsc{Rob}}(1,\tau) + \mathbb{E}[\mathrm{C}_{\textsc{Dart}}(\tau+1,T)]. \qquad (3.12)
\end{aligned}$$

Thus we are faced with the task of bounding $\mathbb{E}[\mathrm{C}_{\textsc{Dart}}(\tau+1,T)]$ in terms of $\mathrm{C}_{\textsc{Rob}}$. Let $\sigma \geq \tau$ be the last time index at which $\lambda_\sigma > 0$ (it is possible that $\sigma = T$, i.e., that the weights λ_t remain strictly positive from time τ through the end of the instance). Note that, if $\sigma < T-1$, then at time $\sigma+1$ the algorithm will move to coinciding with $\textsc{Rob}$, and from time $\sigma+2$ onward the algorithm (and its costs) will exactly coincide with $\textsc{Rob}$. Then the cost of $\textsc{Dart}$ during this phase is

$$\begin{aligned}
&\mathbb{E}[\mathrm{C}_{\textsc{Dart}}(\tau+1,T)] \\
&= \mathbb{E}[\mathrm{C}_{\textsc{Dart}}(\tau+1,\sigma+1) + \mathrm{C}_{\textsc{Dart}}(\sigma+2,T)] \\
&= \mathbb{E}[\mathrm{C}_{\textsc{Dart}}(\tau+1,\sigma)] + f_{\sigma+1}(r_{\sigma+1}) + \mathbb{E}[d(r_{\sigma+1},x_\sigma)] + \mathrm{C}_{\textsc{Rob}}(\sigma+2,T) \\
&\leq \sum_{t=\tau+1}^{\sigma} \mathbb{E}\left[f_t(x_t) + d(x_t,x_{t-1})\right] + \lambda_\sigma d(r_\sigma,a_\sigma) + \mathrm{C}_{\textsc{Rob}}(\sigma+1,T) \\
&\leq \sum_{t=\tau+1}^{\sigma} \lambda_t f_t(a_t) + (1-\lambda_t)f_t(r_t) + (1-\lambda_t)d(r_t,r_{t-1}) + \lambda_t d(a_t,a_{t-1}) \\
&\quad + \frac{\frac{\epsilon}{2}\mathrm{C}_{\textsc{Adv}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2} + \lambda_\sigma d(r_\sigma,a_\sigma) + \mathrm{C}_{\textsc{Rob}}(\sigma+1,T) \qquad (3.13) \\
&\leq \sum_{t=\tau+1}^{\sigma} (1-\lambda_t)\mathrm{C}_{\textsc{Rob}}(t,t) + \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\textsc{Adv}}(t,t) \\
&\quad + \lambda_\sigma d(r_\sigma,a_\sigma) + \mathrm{C}_{\textsc{Rob}}(\sigma+1,T) \qquad (3.14) \\
&\leq \mathrm{C}_{\textsc{Rob}}(\tau+1,T) + \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\textsc{Adv}}(\tau+1,\sigma) + \lambda_\sigma d(r_\sigma,a_\sigma) \qquad (3.15) \\
&\leq \mathrm{C}_{\textsc{Rob}}(\tau+1,T) + \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\textsc{Adv}}(\tau+1,\sigma) + \mathrm{C}_{\textsc{Rob}}(1,\sigma) + \mathrm{C}_{\textsc{Adv}}(1,\sigma) \qquad (3.16) \\
&\leq \mathrm{C}_{\textsc{Rob}}(\tau+1,T) + \mathrm{C}_{\textsc{Rob}}(1,\sigma) + \mathrm{C}_{\textsc{Adv}}(1,\tau) + \left(2+\frac{\epsilon}{4}\right)\mathrm{C}_{\textsc{Adv}}(\tau+1,\sigma) \\
&\leq \mathrm{C}_{\textsc{Rob}}(\tau+1,T) + \mathrm{C}_{\textsc{Rob}}(1,\sigma) + \frac{4}{\epsilon}\mathrm{C}_{\textsc{Rob}}(1,\tau) + \left(2+\frac{\epsilon}{4}\right)\mathrm{C}_{\textsc{Adv}}(\tau+1,\sigma) \qquad (3.17)
\end{aligned}$$

where (3.13) follows (3.9) and the fact that $\lambda_\sigma > 0$, so $\lambda_t = \lambda_{t-1} - \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})}$ exactly for each $t = \tau+1,\ldots,\sigma$, (3.14) follows from $\lambda_t \le \lambda_{t-1}$ for $t = \tau+1,\ldots,\sigma$, (3.16) follows from the triangle inequality applied to $d(r_\sigma, a_\sigma)$, and (3.17) follows by the assumption that $\mathrm{C}_{\mathrm{ROB}}(1,\tau) \ge \frac{\epsilon}{4}\cdot\mathrm{C}_{\mathrm{ADV}}(1,\tau)$.

All that remains is to upper bound $(2+\frac{\epsilon}{4})\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)$ under the assumption that $\lambda_\sigma > 0$. By assumption, $\lambda_\tau = 1$, hence

$$\lambda_\sigma = 1 - \sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})}. \tag{3.18}$$

This begs the question: *given that $\lambda_\sigma > 0$, how large can $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)$ be?* To help answer this question, we prove the following lemma.

Lemma 3.B.1. *Let $(y_i)_{i=0}^T$ be a sequence of nonnegative reals with $y_0 > 0$. Then*

$$\sum_{t=1}^{T} \frac{y_t}{\sum_{i=0}^{t-1} y_i} \ge \log\left(\frac{\sum_{i=0}^{T} y_i}{y_0}\right). \tag{3.19}$$

This lemma can be seen as a generalization of the classical observation that the Tth harmonic number H_T is lower bounded by $\log(T+1)$; indeed, this result can be recovered from Lemma 3.B.1 by setting $y_i = 1$ for all i. The proof of the lemma goes as follows.

Proof. Define a piecewise constant function $y(t) : [0,T] \to \mathbb{R}_+$ as follows:

$$y(t) = \begin{cases} y_1 & \text{for } t \in [0,1) \\ y_2 & \text{for } t \in [1,2) \\ \vdots & \\ y_T & \text{for } t \in [T-1,T] \end{cases}$$

and further define a function $Y(t) : [0,T] \to \mathbb{R}_+$ as its integral:

$$Y(t) = y_0 + \int_0^t y(s)\,\mathrm{d}x.$$

Note that for $t \in [T]$, $Y(t) = \sum_{i=0}^{t} y_i$. Moreover, by the fundamental theorem of calculus, Y'(t) = y(t).

Since $Y(t)$ is increasing, observe that for arbitrary $t \in [T]$,

$$\int_{t-1}^{t} \frac{y(s)}{Y(s)}\,\mathrm{d}s \le \int_{t-1}^{t} \frac{y_t}{Y(t-1)}\,\mathrm{d}s = \frac{y_t}{Y(t-1)} = \frac{y_t}{\sum_{i=0}^{t-1} y_i}.$$

Thus, we may lower bound the sum on the left-hand side of (3.19) as follows:

$$\begin{aligned}\sum_{t=1}^{T} \frac{y_t}{\sum_{i=0}^{t-1} y_i} &\geq \sum_{t=1}^{T} \int_{t-1}^{t} \frac{y(s)}{Y(s)}\, \mathrm{d}s \\ &= \int_0^T \frac{y(s)}{Y(s)}\, \mathrm{d}s \\ &= \int_0^T \frac{Y'(s)}{Y(s)}\, \mathrm{d}s \\ &= [\log(Y(s))]_{s=0}^T \\ &= \log(Y(T)) - \log(Y(0)),\end{aligned}$$

establishing the desired bound. □

With the lemma proved, let us return to (3.18) and the question of how large $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)$ can be given that λ_σ remains strictly positive. By (3.18), this is equivalent to the question of how large $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)$ can be given that the sum $\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})}$ is strictly less than 1. To answer this question, it suffices to prove a lower bound on the sum in terms of $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)$. If we can show that

$$\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})} \geq g(\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)) \tag{3.20}$$

for some strictly increasing function $g:\mathbb{R}_+ \to \mathbb{R}_+$, then $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma) \geq g^{-1}(1)$ would imply that $\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})} \geq 1$. Thus the assumption that $\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})} < 1$ will in turn imply an upper bound of $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma) < g^{-1}(1)$ on the cost, as desired.

Let us thus construct a lower bound in the form of (3.20). Before moving on, we note two inequalities: first,

$$d(a_\tau,r_\tau) \leq \mathrm{C}_{\mathrm{ADV}}(1,\tau) + \mathrm{C}_{\mathrm{ROB}}(1,\tau) \leq \left(1+\frac{4}{\epsilon}\right)\mathrm{C}_{\mathrm{ROB}}(1,\tau) \tag{3.21}$$

by the assumption that $\mathrm{C}_{\mathrm{ROB}}(1,\tau) \geq \frac{\epsilon}{4}\cdot \mathrm{C}_{\mathrm{ADV}}(1,\tau)$. Second, for $t \in \{\tau+1,\dots,\sigma\}$,

$$\begin{aligned}d(a_t,r_t) &\leq \mathrm{C}_{\mathrm{ADV}}(1,t) + \mathrm{C}_{\mathrm{ROB}}(1,t) \\ &\leq \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(1,t) &(3.22)\\ &\leq \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(1,\tau) + \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(\tau+1,t) \\ &\leq \left(1+\frac{4}{\epsilon}\right)\mathrm{C}_{\mathrm{ROB}}(1,\tau) + \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(\tau+1,t), &(3.23)\end{aligned}$$

where (3.22) and (3.23) both follow from the assumption that τ is the last time index in which $\mathrm{C}_{\textsc{Rob}}(1,\tau) \geq \frac{\epsilon}{4} \cdot \mathrm{C}_{\textsc{Adv}}(1,\tau)$. Applying the bounds (3.21) and (3.23), we obtain

$$\begin{aligned}
&\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\textsc{Adv}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})} \\
&\geq \frac{\epsilon}{4} \sum_{t=\tau+1}^{\sigma} \frac{\mathrm{C}_{\textsc{Adv}}(t,t)}{d(a_{t-1},r_{t-1})} \\
&\geq \frac{\epsilon}{4(1+\frac{\epsilon}{4})} \sum_{t=\tau+1}^{\sigma} \frac{(1+\frac{\epsilon}{4})\mathrm{C}_{\textsc{Adv}}(t,t)}{\left(1+\frac{4}{\epsilon}\right)\mathrm{C}_{\textsc{Rob}}(1,\tau) + (1+\frac{\epsilon}{4})\mathrm{C}_{\textsc{Adv}}(\tau+1,t-1)}.
\end{aligned} \tag{3.24}$$

(Recall that $\mathrm{C}_{\textsc{Alg}}(t,t')$ is defined to be 0 when $t' < t$). Applying Lemma 3.B.1 to (3.24) with $y_0 = \left(1+\frac{4}{\epsilon}\right)\mathrm{C}_{\textsc{Rob}}(1,\tau)$ and $y_i = (1+\frac{\epsilon}{4})\mathrm{C}_{\textsc{Adv}}(\tau+i,\tau+i)$ for $i = 1,\ldots,\sigma-\tau$, we obtain

$$\begin{aligned}
&\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\textsc{Adv}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})} \\
&\geq \frac{\epsilon}{4(1+\frac{\epsilon}{4})} \log\left(\frac{\left(1+\frac{4}{\epsilon}\right)\mathrm{C}_{\textsc{Rob}}(1,\tau) + (1+\frac{\epsilon}{4})\mathrm{C}_{\textsc{Adv}}(\tau+1,\sigma)}{\left(1+\frac{4}{\epsilon}\right)\mathrm{C}_{\textsc{Rob}}(1,\tau)}\right) \\
&= \frac{\epsilon}{4+\epsilon} \log\left(1 + \frac{\epsilon}{4}\frac{\mathrm{C}_{\textsc{Adv}}(\tau+1,\sigma)}{\mathrm{C}_{\textsc{Rob}}(1,\tau)}\right).
\end{aligned} \tag{3.25}$$

Thus the lower bound (3.20) holds with $g : \mathbb{R}_+ \to \mathbb{R}_+$ defined as $g(y) = \frac{\epsilon}{4+\epsilon}\log\left(1+\frac{\epsilon}{4}\frac{y}{\mathrm{C}_{\textsc{Rob}}(1,\tau)}\right)$. Since

$$g^{-1}(1) = \frac{4\mathrm{C}_{\textsc{Rob}}(1,\tau)}{\epsilon}\left[\exp\left(\frac{4+\epsilon}{\epsilon}\right) - 1\right],$$

by the argument following (3.20), we obtain the upper bound $\mathrm{C}_{\textsc{Adv}}(\tau+1,\sigma) < \frac{4\mathrm{C}_{\textsc{Rob}}(1,\tau)}{\epsilon}\left[\exp\left(\frac{4+\epsilon}{\epsilon}\right)-1\right]$ on Adv's cost from time $\tau+1$ through σ. Substituting this bound into (3.17), and that bound subsequently into (3.12), we conclude that

$$\begin{aligned}
\mathbb{E}[\mathrm{C}_{\textsc{Dart}}] &\leq \frac{4(1+\epsilon)}{\epsilon}\mathrm{C}_{\textsc{Rob}}(1,\tau) + \mathbb{E}[\mathrm{C}_{\textsc{Dart}}(\tau+1,T)] \\
&\leq \frac{4(1+\epsilon)}{\epsilon}\mathrm{C}_{\textsc{Rob}}(1,\tau) + \mathrm{C}_{\textsc{Rob}}(\tau+1,T) + \mathrm{C}_{\textsc{Rob}}(1,\sigma) \\
&\quad + \frac{4}{\epsilon}\mathrm{C}_{\textsc{Rob}}(1,\tau) + \left(2+\frac{\epsilon}{4}\right)\mathrm{C}_{\textsc{Adv}}(\tau+1,\sigma) \\
&\leq \left(5+\frac{8}{\epsilon}\right)\mathrm{C}_{\textsc{Rob}} + \left(2+\frac{\epsilon}{4}\right)\mathrm{C}_{\textsc{Adv}}(\tau+1,\sigma) \\
&\leq \left(5+\frac{8}{\epsilon}\right)\mathrm{C}_{\textsc{Rob}} + \left(2+\frac{\epsilon}{4}\right)\frac{4\mathrm{C}_{\textsc{Rob}}(1,\tau)}{\epsilon}\left[\exp\left(\frac{4+\epsilon}{\epsilon}\right)-1\right] \\
&= 2^{O(1/\epsilon)}\mathrm{C}_{\textsc{Rob}}.
\end{aligned}$$

This concludes the proof. □

Proof of Corollary 3.3.2

These results follow immediately from Theorem 3.3.1, the definition of bicompetitiveness (Definition 3.2.3), and the observation that an algorithm that is (c,r)-bicompetitive with respect to $(\textsc{Adv},\textsc{Rob})$, where $\textsc{Rob}$ is b-competitive, achieves c-consistency with respect to $\textsc{Adv}$ together with rb robustness. Thus (i) follows from the existence of an $O(\log^2 n)$-competitive algorithm for MTS on any n-point metric space [142]; (ii) follows from the existence of an n-competitive algorithm for CFC on any n-dimensional normed vector space [106], as well as the fact that CFC algorithms can be derandomized by taking the expectation; (iii) follows from the fact that the work function algorithm is $(2k-1)$-competitive for k-server [97]; and (iv) follows from Proposition 3.2.5, i.e., the fact that the work function algorithm is $(2k-1)\max\left\{1,\frac{2}{\alpha}\right\}$-competitive for k-chasing α-polyhedral convex functions.

3.C Proofs for Section 3.4

Proof of Theorem 3.4.1

We proceed under the assumption that $\frac{2}{\epsilon}\in\mathbb{N}$; if this is not the case, then the same result holds up to some small constant factor. We define the metric space (X,d) as follows: $X=\{0\}\cup\{2^i : i=0,\dots,\frac{2}{\epsilon}\}$, and d is just the usual (Euclidean) metric on $\mathbb{R}$: for $x,y\in X$, $d(x,y)=|x-y|$. All algorithms start at $x_0=1$.

The MTS instance realizing the lower bound is constructed as follows: at each time $t=1,\dots,T:=\frac{2}{\epsilon}$, the adversary delivers the service cost function

$$f_t(x) = \infty\cdot\mathbb{1}_{x\notin\{0,2^t\}},$$

forcing any competitive algorithm to assign zero probability mass to any point other than 0 and 2^t. The advice chooses decisions $\textsc{Adv}_t = 2^t$ at each time, i.e., it deterministically chooses the rightmost point with zero service cost. Let $\textsc{Alg}$ be an arbitrary randomized algorithm for MTS that is $(1+\epsilon)$-consistent with respect to $\textsc{Adv}$.

Suppose p_t is the probability that $\textsc{Alg}$ assigns to the state $\textsc{Adv}_t = 2^t$ at time t; $1 - p_t$ is thus the probability assigned to the state 0. If $p_t \leq p_{t-1}$, then the expected movement cost of $\textsc{Alg}$ at time t is

$$\mathbb{W}_X^1(p_t, p_{t-1}) = p_t 2^{t-1} + (p_{t-1} - p_t)2^{t-1} = p_{t-1}2^{t-1}.$$

On the other hand, if $p_t > p_{t-1}$, then the expected movement cost of $\textsc{Alg}$ at time t is

$$\mathbb{W}_X^1(p_t, p_{t-1}) = p_{t-1}2^{t-1} + (p_t - p_{t-1})2^t \geq p_t 2^{t-1} > p_{t-1}2^{t-1}.$$

Combining the above equality and inequality, the total cost of $\textsc{Alg}$ from time 1 through t is bounded below as

$$\mathbb{E}[\mathrm{C}_{\textsc{Alg}}(1,t)] \geq \sum_{\tau=1}^{t} p_{\tau-1}2^{\tau-1} \tag{3.26}$$

for any $t \in [T]$, with $p_0 = 1$ by convention.

Since $\textsc{Alg}$ is $(1+\epsilon)$-consistent with respect to $\textsc{Adv}$, it must be the case that, for each $t \in [T]$,

$$\mathbb{E}[\mathrm{C}_{\textsc{Alg}}(1,t)] + (1 - p_t)2^t \leq (1+\epsilon)\mathrm{C}_{\textsc{Adv}}(1,t). \tag{3.27}$$

If this were not the case, then the adversary could simply send $f_{t+1}(x) = \infty \cdot \mathbb{1}_{x \neq 2^t}$ as the final service cost and end the instance, and $\textsc{Alg}$ would violate the assumed consistency. Note that $\mathrm{C}_{\textsc{Adv}}(1,t) = \sum_{\tau=1}^{t} 2^{\tau-1} = 2^t - 1$. By the inequalities (3.26) and (3.27), it must hold that

$$\sum_{\tau=1}^{t} p_{\tau-1}2^{\tau-1} + (1-p_t)2^t \leq (1+\epsilon)(2^t - 1) \tag{3.28}$$

for all $t \in [T]$. For $t = 1$, this tells us that $1 + 2(1 - p_1) \leq 1 + \epsilon$, so $p_1 \geq 1 - \frac{\epsilon}{2}$. It is straightforward to see via induction that in general, $p_t \geq 1 - t\frac{\epsilon}{2}$. Thus, from (3.26), we obtain

$$\begin{aligned}\mathbb{E}[\mathrm{C}_{\textsc{Alg}}] &\geq \sum_{t=1}^{T} p_{t-1} 2^{t-1} \\ &\geq \sum_{t=1}^{T} \left(1 - (t-1)\frac{\epsilon}{2}\right) 2^{t-1} \\ &= \frac{\epsilon}{2} 2^{\frac{2}{\epsilon}+1} - (1+\epsilon)\end{aligned}$$

where the final equality follows from $T = \frac{2}{\epsilon}$. Thus we have obtained that $\mathbb{E}[\mathrm{C}_{\textsc{Alg}}] = 2^{\Omega(1/\epsilon)}$. Since the offline optimal algorithm for this instance simply moves to 0 and stays there, incurring total cost 1, Alg is thus $2^{\Omega(1/\epsilon)}$-robust. □

Proof of Theorem 3.4.2

The proof proceeds via a reduction to the lower bound presented in the previous proof (Section 3.C). Specifically, we show that the space of probability distributions over the metric space (X, d) from the previous proof endowed with the Wasserstein-1 distance $\mathbb{W}_X^1$ is bijectively isometric to a convex subset K of a vector space endowed with a weighted ℓ^1 norm. This fact, along with a similar correspondence between service costs, will imply that any trajectory of decisions produced by a randomized MTS algorithm on a given problem instance is in one-to-one correspondence with a trajectory of decisions produced by a deterministic CFC algorithm on a corresponding instance, and that moreover, the two trajectories incur identical cost (both movement and service). Note that this correspondence was essentially observed for tree metrics in [142]; our construction is slightly different, so we provide further detail for the sake of completeness.

Let $n = \frac{2}{\epsilon} + 2$, and let $X = \{0\} \cup \{2^i : i = 0, \ldots, \frac{2}{\epsilon}\}$ be as in the previous section. Let the simplex $\Delta_n \subset \mathbb{R}^n$ represent the set of probability distributions over X, with ith coordinate corresponding to the probability assigned to the ith state of X (in increasing order, e.g., 0 is the 1st state). We define the convex body

$$K = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq 0, x_1 = 1, x_i \geq x_{i+1} \text{ for } i = 1, \ldots, n-1\}.$$

Let us define a linear map from Δ_n to K: the map $\mathbf{\Phi} : \mathbb{R}^n \to \mathbb{R}^n$ represented by the upper triangular matrix with all ones on and above the diagonal, and all zeros below the diagonal. In other words,

$$(\mathbf{\Phi p})_i = \sum_{j \geq i} p_j$$

for each $i \in [n]$. It is straightforward to observe that $\mathbf{\Phi}(\Delta_n) \subseteq K$, by the property that any $\mathbf{p} \in \Delta_n$ satisfies $\mathbf{p} \geq 0$ and $\mathbf{1}^\top \mathbf{p} = 1$. To see that $\mathbf{\Phi}^{-1}(K) \subseteq \Delta_n$, first note that $\mathbf{\Phi}^{-1}$ is just the matrix with 1s occupying its diagonal, and -1s just above the diagonal, i.e.,

$$\mathbf{\Phi}^{-1} = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix}.$$

But then, for $\mathbf{x} \in K$,

$$\mathbf{\Phi}^{-1}\mathbf{x} = \begin{bmatrix} x_1 - x_2 \\ \vdots \\ x_{n-1} - x_n \\ x_n \end{bmatrix}.$$

And thus by definition of K, we have $\mathbf{\Phi}^{-1}\mathbf{x} \geq 0$ and $\mathbf{1}^\top \mathbf{\Phi}^{-1}\mathbf{x} = x_1 = 1$. Thus $\mathbf{\Phi}^{-1}(K) \subseteq \Delta_n$, so $\mathbf{\Phi}$ is a bijection between Δ_n and K.

Now, define a vector of weights $\mathbf{w} \in \mathbb{R}^n$ with $w_1 = w_2 = 1$, and $w_i = 2^{i-3}$ for $i = 3, \ldots, n$. We define a correspondingly weighted ℓ^1 norm as follows: for $\mathbf{x} \in \mathbb{R}^n$,

$$\|\mathbf{x}\|_{\ell^1(\mathbf{w})} := \sum_{i=1}^{n} w_i |x_i|.$$

On the other hand, we also consider the Wasserstein-1 distance

$$\mathbb{W}_X^1(\mathbf{p}, \mathbf{p}') = \min_{\gamma \in \Pi(\mathbf{p}, \mathbf{p}')} \mathbb{E}_{(x,x')\sim\gamma}[d(x, x')]$$

between two distributions $\mathbf{p}, \mathbf{p}' \in \Delta_n$ over states of X. Since X is a subset of $\mathbb{R}$ and d is the standard metric on $\mathbb{R}$, the Wasserstein-1 distance can be computed in closed form [154]: defining $F_{\mathbf{p}} : \mathbb{R} \to [0, 1]$ as the cumulative distribution function of $\mathbf{p}$ over $\mathbb{R}$, we have

$$\mathbb{W}_X^1(\mathbf{p}, \mathbf{p}') = \int_{\mathbb{R}} |F_{\mathbf{p}}(t) - F_{\mathbf{p}'}(t)| \, \mathrm{d}t = |p_1 - p_1'| + \sum_{i=2}^{n} 2^{i-2} \sum_{j=1}^{i} |p_j - p_j'|. \tag{3.29}$$

We now show that $\mathbf{\Phi}$ preserves the Wasserstein-1 distance: for any $\mathbf{p}, \mathbf{p}' \in \Delta_n$, we have

$$\|\mathbf{\Phi p} - \mathbf{\Phi p}'\|_{\ell^1(\mathbf{w})} = \sum_{i=1}^{n} w_i \left| \sum_{j=i}^{n} p_j - p_j' \right|. \tag{3.30}$$

Applying the equalities $\mathbf{1}^\top \mathbf{p} = \mathbf{1}^\top \mathbf{p}' = 1$ and $\sum_{j=i}^n p_j = 1 - \sum_{j=1}^{i-1} p_j$ (and similarly for $\mathbf{p}'$), equality of (3.29) and (3.30) follows immediately. Thus $\mathbf{\Phi}$ is a bijective isometry between $(\Delta_n, \mathbb{W}_X^1)$ and $(K, \|\cdot\|_{\ell^1(w)})$.

We now go about showing that on the MTS instance realizing the lower bound proved in the previous section (Section 3.C), there is a corresponding CFC instance with the property that any sequence of decisions $\mathbf{p}_1, \dots, \mathbf{p}_T \in \Delta_n$ for the MTS instance maps under $\mathbf{\Phi}$ to a sequence of decisions $\mathbf{x}_1, \dots, \mathbf{x}_T$ for the CFC instance, and that moreover, these sequences have identical costs for their respective instances. Note that this correspondence will hold more generally beyond the particular instance we consider.

Define for each $t \in [T]$ the vector $\mathbf{c}_t \in \overline{\mathbb{R}}_+^n$ whose 1st and $(t+2)$th entry is 0, with all other entries $+\infty$; these are the vector representations of the service cost functions for the lower bound from the previous section. Then let us define a CFC instance with cost functions $f_t : \mathbb{R}^n \to \overline{\mathbb{R}}_+$ defined as

$$f_t(\mathbf{x}) = \mathbf{c}_t^\top \mathbf{\Phi}^{-1} \mathbf{x} + \infty \cdot \mathbb{1}_{\mathbf{x} \notin K}.$$

The costs f_t are certainly convex, since $\mathbf{c}_t^\top \mathbf{\Phi}^{-1} \mathbf{x}$ is linear and K is a convex set. Moreover, because of the indicator term, the only decisions yielding finite cost are those residing in K. Observe that for any $\mathbf{p} \in \Delta_n$, $f_t(\mathbf{\Phi p}) = \mathbf{c}_t^\top \mathbf{p}$. Thus, it is straightforward to observe by the construction of the cost functions and the fact that $\mathbf{\Phi}$ is a bijective isometry between $(\Delta_n, \mathbb{W}_X^1)$ and $(K, \|\cdot\|_{\ell^1(w)})$ that the CFC instance defined by $f_1, \dots, f_T$ on $\mathbb{R}^n$ is equivalent to the MTS instance defined by $\mathbf{c}_1, \dots, \mathbf{c}_T$ on X, in the sense that sequences of decisions for the latter are in one-to-one correspondence via $\mathbf{\Phi}$ with (finite-cost) sequences of decisions for the former, and this correspondence preserves total cost (both moving and service). Thus any performance bound on algorithms for the MTS instance translates to an identical performance bound on algorithms for the CFC instance, giving the desired result. □

3.D Proofs for Section 3.5

Proof of Theorem 3.5.1

The proof is identical to that of Theorem 3.3.1 presented in Section 3.B, save for the function g realizing the lower bound 3.20. By assumption, $d(a_t, r_t) \leq D$ for all

$t \in [T]$, hence

$$\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{Adv}}(t,t) + (1-\lambda_{t-1}) f_t(a_t)}{2d(a_{t-1}, r_{t-1})} \geq \sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{Adv}}(t,t)}{2D}$$
$$= \frac{\epsilon}{4D}\mathrm{C}_{\mathrm{Adv}}(\tau+1,\sigma).$$

Thus, per the argument following (3.20), $\mathrm{C}_{\mathrm{Adv}}(\tau+1,\sigma) < \frac{4D}{\epsilon}$. Substituting this bound into (3.15), we obtain

$$\begin{aligned}\mathbb{E}[\mathrm{C}_{\mathrm{Dart}}(\tau+1,T)] &\leq \mathrm{C}_{\mathrm{Rob}}(\tau+1,T) + \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{Adv}}(\tau+1,\sigma) + \lambda_\sigma d(r_\sigma, a_\sigma) \\ &\leq \mathrm{C}_{\mathrm{Rob}}(\tau+1,T) + \left(1+\frac{\epsilon}{4}\right)\frac{4D}{\epsilon} + D\end{aligned}$$

and substituting this bound subsequently into (3.12), we conclude

$$\begin{aligned}\mathbb{E}[\mathrm{C}_{\mathrm{Dart}}] &\leq \frac{4(1+\epsilon)}{\epsilon}\mathrm{C}_{\mathrm{Rob}}(1,\tau) + \mathbb{E}[\mathrm{C}_{\mathrm{Dart}}(\tau+1,T)] \\ &\leq \frac{4(1+\epsilon)}{\epsilon}\mathrm{C}_{\mathrm{Rob}}(1,\tau) + \mathrm{C}_{\mathrm{Rob}}(\tau+1,T) + \left(1+\frac{\epsilon}{4}\right)\frac{4D}{\epsilon} + D \\ &\leq \left(4+\frac{4}{\epsilon}\right)\mathrm{C}_{\mathrm{Rob}} + \frac{4D}{\epsilon} + 2D.\end{aligned}$$

Thus the proof. □

Proof of Lemma 3.5.2

Recall that all the servers of both Adv and Rob begin at the state $x_0 \in X$ at time 0. Fix any $t \in [T]$. The algorithm Adv has servers at $a_t^{(1)}, \cdots, a_t^{(k)} \in X$ and Rob has servers at $r_t^{(1)}, \cdots, r_t^{(k)}$. Since Adv and Rob are both lazy, each of these $2k$ servers must either be at x_0, or at some previous request s_i for $i \in [t]$. Consider a pair of server positions $a_t^{(j)}$ and $r_t^{(j)}$ for some $j \in [k]$; if $a_t^{(j)} = r_t^{(j)}$, then $d(a_t^{(j)}, r_t^{(j)}) = 0$. On the other hand, if one of the servers is at x_0 and the other is at s_i for some $i \in [t]$, then $d(a_t^{(j)}, r_t^{(j)}) \leq d(x_0, s_i) \leq \mathrm{C}_{\mathrm{Opt}}(1,t)$, since the offline optimal will also have had to move a server from x_0 to meet the request s_i. Finally, if both $a_t^{(j)}$ and $r_t^{(j)}$ are at different requests s_i, s_j for $i \neq j \in [t]$, then

$$d(a_t^{(j)}, r_t^{(j)}) = d(s_i, s_j) \leq \mathrm{C}_{\mathrm{Opt}}(1,t).$$

To see that this holds, note that if Opt served the requests s_i and s_j with different servers, then by the triangle inequality $\mathrm{C}_{\mathrm{Opt}}(1,t) \geq d(x_0, s_i) + d(x_0, s_j) \geq d(s_i, s_j)$ since all the servers began at x_0. On the other hand, if Opt served s_i and s_j with the same server, then that server must have moved from s_i to s_j (or vice versa), hence $d(s_i, s_j) \leq \mathrm{C}_{\mathrm{Opt}}(1,t)$.

Since there are k such pairs of servers $a_t^{(j)}, r_t^{(j)}$, and since d_{mm} is the minimal matching distance, we obtain

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \leq \sum_{j=1}^{k} d(a_t^{(j)}, r_t^{(j)}) \leq k \cdot \mathrm{C}_{\mathrm{OPT}}(1, t).$$

$\square$

Proof of Theorem 3.5.4

Before proving the theorem, let us formally define *lazy* algorithms for k-chasing convex, α-polyhedral functions on $\mathbb{R}$.

Definition 3.D.1. *An algorithm* ALG *for k-chasing convex, α-polyhedral functions on $\mathbb{R}$ is* ***lazy*** *if, at each time t, the following conditions hold on its decision:*

i. ALG *moves at most a single server at time t, and the only server that it moves (if any) is the one that realizes the service cost.*

ii. ALG *only moves a server in order to obtain a strictly lower service cost.*

That is, if ALG *is a lazy algorithm for k-chasing convex, α-polyhedral functions on $\mathbb{R}$ and $\mathbf{x}_{t-1}, \mathbf{x}_t \in \binom{\mathbb{R}}{k}$ are* ALG*'s decisions at times $t-1$ and t on an instance $g_1, \dots, g_T : \mathbb{R} \to \overline{\mathbb{R}}_+$, then either $\mathbf{x}_{t-1} = \mathbf{x}_t$, or $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ differ by exactly one server, and moreover,*

$$\min_{i \in [k]} g_t(x_t^{(i)}) < \min_{i \in [k]} g_t(x_{t-1}^{(i)}).$$

With this definition formalized, it is straightforward to see by the triangle inequality that, similar to the k-server setting [151, §10.2.3], we can assume without loss of generality that ADV and ROB are lazy algorithms for k-chasing convex functions on $\mathbb{R}$. Next, we prove the following lemma.

Lemma 3.D.2. *Let $g_1, \dots, g_T : \mathbb{R} \to \overline{\mathbb{R}}_+$ be a sequence of α-polyhedral costs for an instance of k-chasing convex, α-polyhedral functions on $\mathbb{R}$ endowed with the usual (Euclidean) metric $d(x, y) = |x - y|$. Let* ADV *and* ROB *be two lazy algorithms for the problem that both start with all servers at the same point $x_0 \in \mathbb{R}$, and let $\mathbf{a}_1, \dots, \mathbf{a}_T \in \binom{\mathbb{R}}{k}$ and $\mathbf{r}_1, \dots, \mathbf{r}_T \in \binom{\mathbb{R}}{k}$ be the sequences of server positions of* ADV *and* ROB*, respectively, on the problem instance. Further suppose that* ROB *ends each timestep with a server at the minimizer $v_t = \arg\min_x g_t(x)$, so, $v_t \in \mathbf{r}_t$ at each time t. Then, for any time $t \in [T]$, we have*

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \leq \max\{k, k/\alpha\} \cdot \mathrm{C}_{\mathrm{OPT}}(1, t),$$

where d_{mm} is the minimal matching distance inherited from the metric d.

Proof. Suppose without loss of generality that all servers start at $x_0 = 0$. Note that since ADV is lazy (without loss of generality) and costs are α-polyhedral and convex, ADV will never move a server away from the minimizer v_t of the current cost function g_t.

Fix any time $t \in [T]$. Suppose without loss of generality that ADV and ROB have servers indexed in increasing order, i.e., $a_t^{(1)} \le \cdots \le a_t^{(k)}$ and $r_t^{(1)} \le \cdots \le r_t^{(k)}$. Define $\tau = \arg\min_{\tau \in [t]} v_\tau$ and $\sigma = \arg\max_{\sigma \in [t]} v_\sigma$. We break into two cases.

(1) Suppose $0 \in [v_\tau, v_\sigma]$. Since ADV begins with all servers at 0 and never moves away from a minimizer, all of its servers will lie in the interval $[v_\tau, v_\sigma]$. Similarly, since ROB begins with all servers at 0, is lazy, and always occupies the current minimizer with a server, all of its server positions will also lie in the interval $[v_\tau, v_\sigma]$. As a result, we have

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \le \sum_{i=1}^{k} d(a_t^{(i)}, r_t^{(i)}) \le k \cdot d(v_\tau, v_\sigma), \tag{3.31}$$

since the minimal matching of $\mathbf{a}_t$ and $\mathbf{r}_t$ will match servers in increasing order, and all servers lie in the interval $[v_\tau, v_\sigma]$.

Now we must simply bound $d(v_\tau, v_\sigma)$ in terms of $C_{\mathrm{OPT}}(1,t)$. Let let o_τ^* be OPT's closest server to v_τ at time τ, and let o_σ^* be OPT's closest server to v_σ at time σ. Since all servers begin at 0, we can thus lower bound $C_{\mathrm{OPT}}(1,t)$ by

$$\begin{aligned} C_{\mathrm{OPT}}(1,t) &\ge g_\tau(o_\tau^*) + d(o_\tau^*, 0) + g_\sigma(o_\sigma^*) + d(o_\sigma^*, 0) \\ &\ge \alpha \cdot d(o_\tau^*, v_\tau) + d(o_\tau^*, 0) + \alpha \cdot d(o_\sigma^*, v_\sigma) + d(o_\sigma^*, 0) \qquad (3.32) \\ &\ge \min\{1, \alpha\} \cdot d(v_\tau, v_\sigma) \qquad (3.33) \end{aligned}$$

where (3.32) follows by α-polyhedrality of the cost functions. Substituting (3.33) into (3.31) then gives

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \le \max\left\{k, \frac{k}{\alpha}\right\} C_{\mathrm{OPT}}(1,t),$$

as desired.

(2) Suppose 0 is outside of the interval $[v_\tau, v_\sigma]$; we may assume without loss of generality that $0 < v_\tau$. By similar reasoning as in the previous case, all of the servers of ADV and ROB, by laziness, are in the interval $[0, v_\sigma]$. Then we follow

a similar argument. Note that

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \leq \sum_{i=1}^{k} d(a_t^{(i)}, r_t^{(i)}) \leq k \cdot d(0, v_\sigma),$$

and, defining o_σ^* as Opt's closest server to v_σ at time σ,

$$\begin{aligned} \mathrm{C}_{\mathrm{Opt}}(1,t) &\geq g_\sigma(o_\sigma^*) + d(o_\sigma^*, 0) \\ &\geq \alpha \cdot d(o_\sigma^*, v_\sigma) + d(o_\sigma^*, 0) \\ &\geq \min\{1, \alpha\} \cdot d(0, v_\sigma). \end{aligned}$$

Thus we obtain

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \leq \max\left\{k, \frac{k}{\alpha}\right\} \mathrm{C}_{\mathrm{Opt}}(1,t),$$

completing the proof.

□

The result of Theorem 3.5.4 then follows immediately by substituting the diameter bound from Lemma 3.D.2 into Theorem 3.5.1 and instantiating Rob with the work function algorithm applied to the minimizer sequence $v_1, \ldots, v_T$, which we know is $\mathcal{O}(\frac{k}{\alpha})$-competitive. □

3.E Proof of Theorem 3.6.1

We begin by showing $\mathrm{C}_{\mathrm{DetRobustML}} \leq (1 + \epsilon + \delta)\mathrm{C}_{\mathrm{Adv}}$. Note that DetRobustML (Algorithm 7) consists of *phases* in which DetRobustML first coincides with Adv, and then switches to following Rob, before switching back to Adv, and so on. We will assume that DetRobustML ends the instance coinciding with Adv, so $x_T = a_T$; the case in which DetRobustML ends at r_T is similar. Let t_i denote the timestep in which DetRobustML switches from Rob back to Adv for the ith time, with $t_0 := 1$ since DetRobustML always begins by following Adv. Similarly, let m_i denote the timestep in which DetRobustML switches from Adv to Rob for the ith time. Clearly we have $1 = t_0 < m_1 < t_1 < \cdots < m_k < t_k \leq T$, for some $k \in \mathbb{N}$. Even though DetRobustML ends at Adv, define $m_{k+1} := T + 1$ for notational simplicity. Then the cost of DetRobustML may be written as

$$\begin{aligned}
\mathrm{C}_{\textsc{DetRobustML}} &= \sum_{\tau=1}^{m_1-1} f_\tau(a_\tau) + d(a_\tau, a_{\tau-1}) \\
&\quad + \sum_{i=1}^{k} \Bigg(f_{m_i}(r_{m_i}) + d(r_{m_i}, a_{m_i-1}) + \sum_{\tau=m_i+1}^{t_i-1} f_\tau(r_\tau) + d(r_\tau, r_{\tau-1}) \\
&\qquad\qquad + f_{t_i}(a_{t_i}) + d(a_{t_i}, r_{t_i-1}) + \sum_{\tau=t_i+1}^{m_{i+1}-1} f_\tau(a_\tau) + d(a_\tau, a_{\tau-1}) \Bigg) \\
&\le \mathrm{C}_{\textsc{Adv}}(1, m_1 - 1) + \sum_{i=1}^{k} \Big(\mathrm{C}_{\textsc{Rob}}(m_i, t_i - 1) + d(r_{m_i-1}, a_{m_i-1}) \\
&\qquad\qquad + \mathrm{C}_{\textsc{Adv}}(t_i, m_{i+1} - 1) + d(a_{t_i-1}, r_{t_i-1}) \Big) \qquad (3.34) \\
&\le \mathrm{C}_{\textsc{Adv}}(1, m_1 - 1) + 2kD \\
&\quad + \sum_{i=1}^{k} \mathrm{C}_{\textsc{Rob}}(m_i, t_i - 1) + \mathrm{C}_{\textsc{Adv}}(t_i, m_{i+1} - 1) \qquad (3.35) \\
&\le (1+\epsilon)\mathrm{C}_{\textsc{Adv}} + \sum_{i=1}^{k} \mathrm{C}_{\textsc{Rob}}(m_i, t_i - 1) \qquad (3.36) \\
&\le (1+\epsilon+\delta)\mathrm{C}_{\textsc{Adv}} \qquad (3.37)
\end{aligned}$$

where (3.34) follows from the triangle equality on $d(r_{m_i}, a_{m_i-1})$ and $d(a_{t_i}, r_{t_i-1})$, and (3.35) follows by the diameter bound. The inequality (3.36) follows by line 6 of the algorithm, which states that the algorithm will switch from following Adv to following Rob at time t only if $\mathrm{C}_{\textsc{Adv}}(s,t) \ge \frac{2D}{\epsilon}$. Noting that at the start of any timestep t, s is exactly

$$s = \max_{i: m_i + 1 \le t} m_i + 1$$

(with $m_0 := 0$ for notational convenience), it follows that for each $i \in [k]$, $\mathrm{C}_{\textsc{Adv}}(m_{i-1}+1, m_i) \ge \frac{2D}{\epsilon}$. Thus

$$2kD \le \epsilon \sum_{i=1}^{k} \mathrm{C}_{\textsc{Adv}}(m_{i-1}+1, m_i) = \epsilon \cdot \mathrm{C}_{\textsc{Adv}}(1, m_k) \le \epsilon \cdot \mathrm{C}_{\textsc{Adv}}.$$

Finally, (3.37) follows from

$$\sum_{i=1}^{k} \mathrm{C}_{\textsc{Rob}}(m_i, t_i - 1) \le \mathrm{C}_{\textsc{Rob}}(1, t_k - 1) < \delta \cdot \mathrm{C}_{\textsc{Adv}}(1, t_k - 1) \le \delta \cdot \mathrm{C}_{\textsc{Adv}},$$

since by definition, $x_{t_k-1} = r_{t_k-1}$, which by line 12 of the algorithm means that $\mathrm{C}_{\textsc{Rob}}(1, t_k - 1) < \delta \cdot \mathrm{C}_{\textsc{Adv}}(1, t_k - 1)$. Thus we have proved the desired bound $\mathrm{C}_{\textsc{DetRobustML}} \le (1 + \epsilon + \delta)\mathrm{C}_{\textsc{Adv}}$.

We now turn to showing $\mathrm{C}_{\textsc{DetRobustML}} \le \left(1 + \frac{1+\epsilon}{\delta}\right)\mathrm{C}_{\textsc{Rob}} + \left(1 + \frac{2}{\epsilon}\right)D$. First suppose DetRobustML finishes the instance coinciding with Adv, so $x_T = a_T$. Let $\tau \in \{0, \dots, T-1\}$ denote the last time at which DetRobustML coincided with Rob, or that $x_\tau = r_\tau$. Thus the cost can be bounded as

$$
\begin{aligned}
\mathrm{C}_{\textsc{DetRobustML}} &= \mathrm{C}_{\textsc{DetRobustML}}(1, \tau+1) + \mathrm{C}_{\textsc{DetRobustML}}(\tau+2, T) \\
&\le (1 + \epsilon + \delta)\mathrm{C}_{\textsc{Adv}}(1, \tau+1) + \mathrm{C}_{\textsc{Adv}}(\tau+2, T) \qquad (3.38) \\
&\le \max\left\{\left(1 + \frac{1+\epsilon}{\delta}\right)\mathrm{C}_{\textsc{Rob}}(1, \tau+1) + \frac{2D}{\epsilon}, \left(1 + \frac{1+\epsilon}{\delta}\right)\mathrm{C}_{\textsc{Rob}}\right\} \qquad (3.39) \\
&\le \left(1 + \frac{1+\epsilon}{\delta}\right)\mathrm{C}_{\textsc{Rob}} + \frac{2D}{\epsilon} \qquad (3.40)
\end{aligned}
$$

where (3.38) follows via the previously proved inequality $\mathrm{C}_{\textsc{DetRobustML}} \le (1 + \epsilon + \delta)\mathrm{C}_{\textsc{Adv}}$, and (3.39) follows by the fact (according to line 14 of the algorithm) that DetRobustML switching from Rob to Adv at time $\tau + 1$ means that $\mathrm{C}_{\textsc{Rob}} \ge \delta \cdot \mathrm{C}_{\textsc{Adv}}(1, \tau+1)$, as well as from the following observation: since DetRobustML coincides with Adv between times $\tau + 1$ and T, line 6 of the algorithm tells us that either $\mathrm{C}_{\textsc{Adv}}(\tau+2, T) < \frac{2D}{\epsilon}$ or $\mathrm{C}_{\textsc{Rob}} \ge \delta \cdot \mathrm{C}_{\textsc{Adv}}$.

Finally, suppose DetRobustML finishes the instance coinciding with Rob, so $x_T = r_T$. Let $\sigma \in \{0, \dots, T-1\}$ denote the last time at which DetRobustML coincided with Adv, or that $x_\sigma = a_\sigma$. By the previous case's inequality (3.40), we have

$$
\begin{aligned}
\mathrm{C}_{\textsc{DetRobustML}} &= \mathrm{C}_{\textsc{DetRobustML}}(1, \sigma) + \mathrm{C}_{\textsc{DetRobustML}}(\sigma+1, T) \\
&\le \left(1 + \frac{1+\epsilon}{\delta}\right)\mathrm{C}_{\textsc{Rob}}(1, \sigma) + \frac{2D}{\epsilon} + f_{\sigma+1}(r_{\sigma+1}) \\
&\quad + d(r_{\sigma+1}, a_\sigma) + \mathrm{C}_{\textsc{Rob}}(\sigma+2, T) \\
&\le \left(1 + \frac{1+\epsilon}{\delta}\right)\mathrm{C}_{\textsc{Rob}}(1, \sigma) + \frac{2D}{\epsilon} + D + \mathrm{C}_{\textsc{Rob}}(\sigma+1, T) \\
&\le \left(1 + \frac{1+\epsilon}{\delta}\right)\mathrm{C}_{\textsc{Rob}} + \left(1 + \frac{2}{\epsilon}\right)D.
\end{aligned}
$$

□

Chapter 4

THE ONLINE PAUSE AND RESUME PROBLEM: OPTIMAL ALGORITHMS AND AN APPLICATION TO CARBON-AWARE LOAD SHIFTING

We now take a brief detour from the design of learning-augmented algorithms to investigate the design of competitive algorithms for online optimization problems with switching costs and long-term deadline constraints. We propose to study the online pause and resume problem, where a player attempts to find the k lowest (alternatively, highest) prices in a sequence of fixed length T, which is revealed sequentially. At each timestep, the player is presented with a price and decides whether to accept or reject it. The player incurs a *switching cost* whenever their decision changes in consecutive timesteps, i.e., whenever they pause or resume purchasing. This online problem is motivated by the goal of carbon-aware load shifting, where a workload may be paused during periods of high carbon intensity and resumed during periods of low carbon intensity, and a cost is incurred when saving or restoring its state. This problem has strong connections to existing problems studied in the literature on online optimization, though it introduces unique technical challenges that prevent the direct application of existing algorithms. Extending prior work on threshold-based algorithms, we introduce *double-threshold* algorithms for both the minimization and maximization variants of this problem. We further show that the competitive ratios achieved by these algorithms are the best achievable by any deterministic online algorithm. Finally, we empirically validate our proposed algorithm through case studies on the application of carbon-aware load shifting using real carbon trace data and existing baseline algorithms.

This chapter is primarily based on the following paper:

[1] A. Lechowicz, N. Christianson, J. Zuo, N. Bashir, M. Hajiesmaili, A. Wierman, and P. Shenoy, "The Online Pause and Resume Problem: Optimal Algorithms and An Application to Carbon-Aware Load Shifting," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 7, no. 3, 45:1–45:32, Dec. 2023. DOI: 10.1145/3626776. [Online]. Available: https://doi.org/10.1145/3626776.

4.1 Introduction

This chapter introduces and studies the *online pause and resume problem* (OPR), considering both minimization (OPR-min) and maximization (OPR-max) variants. In OPR-min, a player is presented with time-varying prices in a sequential manner and decides whether or not to purchase one unit of an item at the current price. The player must purchase k units of the item over a time horizon of T and they incur a *switching cost* whenever their decision changes in consecutive timesteps, i.e., whenever they pause or resume purchasing. The goal of the player is to minimize their total cost, which consists of the aggregate price of purchasing k units and the aggregate switching cost incurred over T slots. In OPR-max, the setting is exactly the same, but the goal of the player is to maximize their total profit, and any switching cost that they incur is subtracted. In both cases, the price values are revealed to the player one by one in an online manner, and the player has to make a decision without knowing the future values.

Our primary motivation for introducing OPR is the emerging importance of carbon-aware computing and, more specifically, carbon-aware temporal workload shifting, which has seen significant attention in recent years [33, 155–157]. In carbon-aware temporal workload shifting, an interruptible and deferrable workload may be paused during periods of high carbon intensity and resumed during periods of low carbon intensity. The workload must be running for k units of time to complete and must be completed before its deadline T. However, pausing and resuming the workload typically comes with overheads such as storing the state in memory and checkpointing. For example, an empirical study [158] shows that this overhead can nullify any savings in carbon emissions from temporal shifting if the job is interrupted frequently. Moreover, with the rise of big ML training workloads, such as the training, fine-tuning, and inference of large language models (LLM), datacenter workloads' memory footprints are frequently in the hundreds of GBs [159–161]. These emerging workloads will result in high checkpoint-and-restore overheads, which must be considered in carbon-aware scheduling. This motivates adding a *switching cost* in OPR, since a naïve algorithm that does not account for the interruptions' overhead may frequently checkpoint and, in some cases, increase carbon emissions beyond a carbon-agnostic execution.

The objective of temporal workload shifting is to minimize the total carbon footprint of running the workload, which includes both the original compute demand and the overhead due to pausing and resuming (a.k.a., the switching cost). We consider a worst-case performance objective based on competitive analysis, defined explicitly in Section 4.2, wherein we seek to find an effective algorithm that is robust to uncertain and nonstochastic fluctuations in price (or carbon intensity in the context of carbon-aware load shifting). We note that even though statistical modeling of grid carbon intensity has been explored [162], we focus on developing worst-case optimized algorithms as the intended application has nonlinearity and nonstationarity, which complicate the task of designing a single probabilistic model to solve this problem.

OPR also captures other interesting applications with highly variable time-varying costs where switching frequently is undesirable. A related example is the carbon-aware electric vehicle (EV) charging problem, which considers when to charge an EV with respect to the time-varying availability of carbon-free electricity, a charging deadline (e.g., set by the EV owner), and battery health design goals (i.e., a constant charging rate is better for battery longevity) [150]. When the charger is *non-adaptive* (i.e., the charging rate is either 0 or the maximum rate), the problem reduces exactly to OPR. Beyond these "carbon-aware" applications, there are additional examples that deal with pricing, such as managing grid-scale energy storage with respect to real-time prices in the wholesale electricity market, where a "smooth" charge or discharge rate is desired [163]. Another example is renting spot virtual machines from a cloud service provider in the setting where pricing is set according to supply-demand dynamics [164–166].

On the theory front, the OPR problem has strong connections to various existing problems in the literature on online optimization. We extensively review the prior literature in Section 4.7 and focus on the most relevant theoretical problems below. The OPR problem is a generalization of the k-search problem [79, 96], which belongs to the broader class of online conversion problems [167], a.k.a., time series search and one-way trading [78]. In the minimization variant of the k-search problem, an online decision-maker aims to buy k units of an item for the least cost over a sequence of time-varying cost values. At each step, a cost value is observed, and the decision is whether or not to buy one unit at the current observed cost without knowing the future values (see Section 4.2 for a deeper discussion of k-search). In contrast to k-search, the OPR problem introduces the additional component of managing the switching cost, which poses a significant additional challenge in algorithm design.

The existence of the switching cost in OPR connects it to the well-studied problem of smoothed online convex optimization (SOCO) [38], also known as convex function chasing (CFC) [77], and its generalizations including metrical task systems (MTS) [76]. In SOCO, a learner is faced with a sequence of cost functions f_t that are revealed online, and must choose an action x_t after observing f_t. Based on that decision, the learner incurs a hitting cost, $f_t(x_t)$ as well as a switching cost, $\|x_t - x_{t-1}\|$, which captures the cost associated with changing the decision between rounds. In contrast to SOCO, OPR includes the long-term constraint of satisfying the demand of k units over the horizon T, which poses a significant challenge not present in SOCO-like problems.

The coexistence of these differentiating factors, namely the *switching cost* and the *long-term deadline constraint*, make OPR uniquely challenging, and means that prior algorithms and analyses for related problems such as k-search and SOCO cannot be directly adapted.

Contributions. We introduce online algorithms for the minimization and maximization variants of OPR and show that our algorithms achieve the best possible competitive ratios. We also evaluate the empirical performance of the proposed algorithms on a case study of carbon-aware load shifting. The details of our contributions are outlined below.

Algorithmic idea: Double-threshold To tackle OPR, we focus our efforts on online threshold-based algorithms (`OTA`), the prominent design paradigm for classic problems such as k-search [79, 96], one-way trading [78, 167], and online knapsack problems [168–170]. In the k-min search problem, for example, a threshold-based algorithm specifies k threshold values and chooses to trade the i-th item only if the current price is less than or equal to the value suggested by the i-th threshold value.

Direct application of prior `OTA` algorithms to OPR results in undesirable behavior (such as frequently changing decisions) since their threshold function design is oblivious to the switching cost present in OPR. To address this challenge, we seek an algorithm that can simultaneously achieve the following behaviors: (1) when the player is in "trading mode," they should not impulsively switch away from trading in response to a price that is only slightly worse, since this will result in a switching penalty; and (2) the player should not switch to "trading mode" unless prices are sufficiently good to warrant the switching cost. These two ideas motivate an algorithm design that uses two distinct threshold functions, each of which captures one of the above two cases. We present our algorithms `DTPR-min` and `DTPR-max` for OPR-min and OPR-max, respectively, in Section 4.3, which build upon this high-level idea of a double-threshold.

Main results While `OTA` algorithms are intuitive and simple to describe, it is highly challenging to design threshold functions that lead the corresponding algorithms to be competitive against the offline optimum. The addition of switching cost in OPR further exacerbates the technical challenge of designing optimal threshold functions. The key result which enables our double-threshold approach is a technical observation (see Observation 4.3.3), which shows that the difference between the functions guiding the algorithm's decisions should be exactly 2β, where β represents the fixed switching cost incurred by changing the decision in OPR.

Identifying this relationship between the two threshold functions significantly facilitates the competitive analysis of both `DTPR-min` and `DTPR-max`, enabling our derivation of a closed form of each threshold. Using this idea, we characterize the competitive ratios of `DTPR-min` and `DTPR-max` as a function of problem parameters, including an explicit dependence on the magnitude of the switching cost β (see Theorems 4.4.1 and 4.4.2). Furthermore, we derive lower bounds for the competitive ratio of any deterministic online algorithm, showing that our proposed algorithms are optimal for this problem (formal statements in Theorems 4.4.5 and 4.4.6). The competitive ratios we derive for both `DTPR-min` and `DTPR-max` exactly recover the best prior competitive results for the k-search problem [79], which corresponds to the case of $\beta = 0$ in OPR, i.e., no switching cost. Formal statements and a more detailed discussion of our main results are presented in Section 4.4.

Case study. Finally, in Section 4.6, we illustrate the performance of our proposed algorithm by conducting an experimental case study simulating the carbon-aware load shifting problem. We utilize real-world *carbon traces* from Electricity Maps [171], which contain carbon intensity values for grid-sourced electricity across the world. Our experiments simulate different strategies for scheduling a deferrable and interruptible workload in the face of uncertain future carbon intensity values. We show that our algorithm's performance significantly improves upon existing baseline methods and adapted forms of algorithms for related problems such as k-min search.

4.2 Problem Formulation and Preliminaries

We begin by formally introducing the OPR problem and providing background on the online threshold-based algorithm design paradigm, which is used in the design of our proposed algorithms. Table 4.1 summarizes the core notation for OPR. Recall that this formulation is motivated by the setting of carbon-aware temporal workload shifting, as described in the introduction.

Problem Formulation

We present two variants of the online pause and resume problem (OPR).[1] In OPR-min (OPR-max) a player must buy (sell) $k \geq 1$ units of some asset (one unit at each timestep) with the goal of minimizing (maximizing) their total cost (profit) within a time horizon of length T. At each timestep $1 \leq t \leq T$, the player is presented with a price c_t, and must immediately decide whether to accept this price ($x_t = 1$) or reject it ($x_t = 0$). The player is required to complete this transaction for all k units by some point in time T. Both k and T are known in advance. Thus, the requirement of k transactions is a hard constraint, i.e., $\sum_{t=1}^{T} x_t = k$, and if at time $T - i$ the player still has i units remaining to buy/sell, they must accept the prices in the subsequent i slots to accomplish k transactions.

Additionally, in both variants of OPR, the player incurs a *fixed switching cost* $\beta > 0$ whenever they decide to change decisions between two adjacent timesteps (i.e., when $\|x_{t-1} - x_t\| = 1$). We assume that $x_0 = 0$ and $x_{T+1} = 0$, implying that any player must incur a minimum switching cost of 2β, once for switching "on" and once for switching "off." While the player incurs at least a switching cost of 2β, note that the total switching cost incurred by the player is bounded by the size of the asset k since the switching cost cannot be larger than $k2\beta$.

In summary, the offline version of OPR-min can be summarized as follows:

$$\begin{aligned} \min_{x_1,\ldots,x_T \in \{0,1\}} \quad & \underbrace{\left(\sum_{t=1}^{T} c_t x_t \right)}_{\text{Accepted prices}} + \underbrace{\left(\sum_{t=1}^{T+1} \beta ||x_t - x_{t-1}|| \right)}_{\text{Switching cost}} \\ \text{s.t.} \quad & \underbrace{\sum_{t=1}^{T} x_t = k,}_{\text{Deadline constraint}} \end{aligned} \tag{4.1}$$

[1] We use OPR whenever the context is applicable to both minimization (OPR-min) and maximization (OPR-max) variants of the problem, otherwise, we refer to the specific variant. The same policy applies to DTPR, our proposed algorithm for OPR.

Table 4.1: A summary of key notation in the OPR problem

Notation	Description
$k \in \mathbb{N}$	Number of units which must be bought (or sold)
T	Deadline constraint; the player must buy (or sell) k units before time T
$t \in [1, T]$	Current timestep
$x_t \in \{0, 1\}$	Decision at time t. $x_t = 1$ if price c_t is accepted, $x_t = 0$ if c_t is not accepted
β	Switching cost incurred when algorithm's decision $x_t \neq x_{t-1}$
U	Upper bound on any price that will be encountered
L	Lower bound on any price that will be encountered
$\theta = U/L$	Price fluctuation ratio
c_t	(*Online input*) Price revealed to the player at time t
$c_{\min}$ & $c_{\max}$	(*Online input*) The actual minimum and maximum prices in a sequence

while the offline version of OPR-max is

$$\begin{aligned} \max_{x_1,\dots,x_T \in \{0,1\}} \quad & \left(\sum_{t=1}^{T} c_t x_t \right) - \left(\sum_{t=1}^{T+1} \beta \| x_t - x_{t-1} \| \right) \\ \text{s.t.} \quad & \sum_{t=1}^{T} x_t = k. \end{aligned} \tag{4.2}$$

Of course, our focus is the online version of OPR, where the player must make irrevocable decisions at each timestep without the knowledge of future inputs. More specifically, in both variants of OPR the sequence of prices $\{c_t\}_{t \in [1,T]}$ is revealed sequentially—future prices are *unknown* to an online algorithm, and each decision x_t is irrevocable.

Competitive analysis Our goal is to design an online algorithm that maintains a small *competitive ratio* [76], i.e., performs nearly as well as the offline optimal solution. For an online algorithm ALG and an offline optimal solution OPT, the competitive ratio for a minimization problem is defined as: $\text{CR}(\texttt{ALG}) = \max_{\sigma \in \Omega} \texttt{ALG}(\sigma)/\texttt{OPT}(\sigma)$, where σ denotes a valid input sequence for the problem and Ω is the set of all feasible input instances. Further, $\texttt{OPT}(\sigma)$ is the optimal cost given this input, and $\texttt{ALG}(\sigma)$ is the cost of the solution obtained by running the online algorithm over this input. Conversely, for a problem with a maximization objective, the competitive ratio is defined as $\max_{\sigma \in \Omega} \texttt{OPT}(\sigma)/\texttt{ALG}(\sigma)$. With these definitions, the competitive ratio for both minimization and maximization problems is always greater than or equal to one, and the lower the better.

Note that competitive algorithm development, in its classic worst-case optimized design, cannot capture data-driven adaptation and stochasticity of data in decision-making. However, beyond the significance of the theoretical analysis in this framework, competitive algorithms could be of interest to practitioners since they are robust against adversarial or non-stationary behavior in the underlying environment. For example, in the context of carbon-aware load shifting, the carbon intensity values significantly change across the temporal and spatial domains following the makeup and behavior of an electric grid (e.g., different ISOs and generation mixes; see Figure 4.9 in the appendix); and online algorithms are robust to those drastic temporal and spatial variations. Competitive algorithms are extremely simple to implement, e.g., in OPR, all we need are two threshold functions to decide the pause and resume decisions. Furthermore, worst-case optimized algorithms can potentially be augmented with machine-learned predictions, as explored in, e.g., [69, 70, 75, 111, 140, 167], to achieve the best of both worlds of worst-case and average-case performance.

Assumptions and additional notations. We make no assumptions on the underlying distribution of the prices other than the assumption that the set of prices arriving online $\{c_t\}_{t\in[1,T]}$ has bounded support, i.e., $c_t \in [L, U]\ \forall t \in [1,\ T]$, where L and U are known to the player. We also define $\theta = U/L$ as the *price fluctuation*. These are standard assumptions in the literature for many online problems, including one-way trading, online search, and online knapsack; and without them the competitive ratio of any algorithm is unbounded. Most papers in this literature additionally assume that $U, L > 0$ (i.e., the lowest price is still positive), but our design can handle the special case where $L = 0$, and therefore do not adopt this assumption. We use $c_{\min}(\sigma) = \min_{t\in[1,T]} c_t$ and $c_{\max}(\sigma) = \max_{t\in[1,T]} c_t$ to denote the minimum and maximum encountered prices for any valid OPR sequence σ.

Background: Online Threshold-Based Algorithms (OTA)

Online threshold-based algorithms (OTA) are a family of algorithms for online optimization in which a carefully designed *threshold function* is used to specify the decisions made at each timestep. At a high level, the threshold function defines the “minimum acceptable quality” that an arriving input/price must satisfy in order to be accepted by the algorithm. The threshold is chosen specifically so that an agent greedily accepting prices meeting the threshold at each step will be ensured a competitive guarantee. This algorithmic framework has seen success in the online search and one-way trading problems [78, 79, 96, 167] as well as the related online knapsack problem [168–170]. In these works, the derived threshold functions are optimal in the sense that the competitive ratios of the resulting threshold-based algorithms match information-theoretic lower bounds of the corresponding online problems. As discussed in the introduction, the framework does not apply directly to the OPR setting, but we make use of ideas and techniques from this literature. We briefly detail the most relevant highlights from the prior results before discussing how these related problems generalize to OPR in the next section.

1-min/1-max search. In the online 1-min/1-max search problem, a player attempts to find the single lowest (respectively, highest) price in a sequence, which is revealed sequentially. The player's objective is to either minimize their cost or maximize their profit. When each price arrives, the player must decide immediately whether to accept the price, and the player is forced to accept exactly one price before the end of the sequence. For this problem, El-Yaniv et al. [78] presents a deterministic threshold-based algorithm. The algorithm assumes a finite price interval, i.e., the price is bounded by the interval $[L, U]$, where L and U are known. Then, it sets a constant threshold $\Phi = \sqrt{LU}$, and the algorithm simply selects the first price that is less than or equal to Φ (for the maximization version, it accepts the first price greater than or equal to Φ). This algorithm achieves a competitive ratio of $\sqrt{U/L} = \sqrt{\theta}$, which matches the lower bound; hence, it is optimal [78].

k-min/k-max search. The online k-min/k-max search problem extends the 1-min/1-max search problem—a player attempts to find the k lowest (conversely, highest) prices in a sequence of prices revealed sequentially. The player's objective is identical to the 1-min/1-max problem, and the player must accept at least k prices by the end of the sequence. Several works have developed a known optimal deterministic threshold-based algorithm for this problem, including [78, 79]. Leveraging the same assumption of a finite price interval $[L, U]$, the threshold function is a sequence of k thresholds $\{\Phi_i\}_{i \in [1,k]}$, which is also called the *reservation price policy*. At each step, the algorithm accepts the first price, which is less than or equal to Φ_i, where $i - 1$ is the number of prices that have been accepted thus far (for the maximization version, it accepts the first price which is $\geq \Phi_i$). In the k-min setting, this algorithm is α-competitive, where α is the unique solution of

$$\frac{1 - 1/\theta}{1 - 1/\alpha} = \left(1 + \frac{1}{\alpha k}\right)^k. \tag{4.3}$$

For the k-max variant, this algorithm is ω-competitive, where ω is the unique solution of

$$\frac{\theta - 1}{\omega - 1} = \left(1 + \frac{\omega}{k}\right)^k. \tag{4.4}$$

The sequence of thresholds $\{\Phi_i\}_{i \in [1,k]}$ for both variants of the problem are constructed by analyzing possible input cases, "hedging" against the risk that future (unknown) prices will jump to the worst possible value, i.e., U for k-min search, L for k-max search. These potential cases can be enumerated for different values of i, where $0 \leq i \leq k$ denotes the number of prices accepted so far. By simultaneously *balancing* the competitive ratios for each of these cases (setting each ratio equal to the others), the optimal threshold values and the optimal competitive ratios are derived. We refer to this technique as the *balancing rule* and a rigorous proof of this approach, with corresponding lower bounds, can be found in [79]. The lower bounds highlight that the α and ω which solve the expressions for the competitive ratios above are optimal for any deterministic k-min and k-max search algorithms, respectively. Further, α and ω provide insight into a fundamental difference between the minimization and maximization settings of k-search. As discussed in [79], for large θ, the best algorithm for k-max search is roughly $O(k\sqrt[k]{\theta})$-competitive, while the best algorithm for k-min search is at best $O(\sqrt{\theta})$-competitive. Similarly, for fixed θ and large k, the optimal competitive ratio for k-max search is roughly $O(\ln \theta)$, while the optimal competitive ratio for k-min search converges to $O(\sqrt{\theta})$.

4.3 Double Threshold Pause and Resume (DTPR) Algorithm

A fundamental challenge in algorithm design for OPR is how to characterize threshold functions that incorporate the presence of switching costs in their design. Our key algorithmic insight is to incorporate the switching cost into the threshold function by defining *two distinct threshold functions*, where the function to be used for price admittance changes based on the current state (i.e., whether or not the previous price was accepted by the algorithm).

To provide intuition for the state-dependence of the threshold function, consider the setting of OPR-min. At a high level, if the player has not accepted the previous

Algorithm 8: Double Threshold Pause and Resume for OPR-min (DTPR-min)

Input: threshold values $\{\ell_i\}_{i \in [1,k]}$ and $\{u_i\}_{i \in [1,k]}$ defined in Eq. (4.5), deadline T

Output: online decisions $\{x_t\}_{t \in [1,T]}$

initialize: $i = 1$

while price c_t arrives **and** $i \le k$ **do**

  // close to the deadline T, we must accept remaining prices

  if $(k - i) \ge (T - t)$ **then**

    price c_t is accepted, set $x_t = 1$

  // If the previous price was not accepted, use the lower thresholds

  else if $x_{t-1} = 0$ **then**

    if $c_t \le \ell_i$ **then**

      price c_t is accepted, set $x_t = 1$

    else

      price c_t is rejected, set $x_t = 0$

  // If the previous price was accepted, use the upper thresholds

  else if $x_{t-1} = 1$ **then**

    if $c_t \le u_i$ **then**

      price c_t is accepted, set $x_t = 1$

    else

      price c_t is rejected, set $x_t = 0$

  update $i = i + x_t$

end

price, they should wait to accept anything until prices are sufficiently low to justify incurring a cost to switch decisions. On the other hand, if the player has accepted the previous price, they might be willing to accept a slightly higher price—if they do not accept this price, they will incur a cost to switch decisions. While this high-level idea is intuitive, characterizing the form of threshold functions such that the resulting algorithms are competitive is challenging.

The DTPR-min algorithm Our proposed algorithm, Double Threshold Pause and Resume (DTPR) for OPR-min is summarized in Algorithm 8. Prior to any prices arriving online, DTPR-min computes two families of threshold values, $\{\ell_i\}_{i \in [1,k]}$ and $\{u_i\}_{i \in [1,k]}$, where $\ell_i \le u_i \ \forall i \in [1, k]$, whose values are defined as follows.

Definition 4.3.1 (DTPR-min Threshold Values)**.** *For each integer i on the interval $[1, k]$, the following expressions give the corresponding threshold values of u_i and*

ℓ_i *for* `DTPR-min`.

$$u_i = U\left[1 - \left(1 - \frac{1}{\alpha}\right)\left(1 + \frac{1}{k\alpha}\right)^{i-1}\right] + 2\beta\left[\left(\frac{1}{k\alpha} - \frac{1}{k} + 1\right)\left(1 + \frac{1}{k\alpha}\right)^{i-1}\right],$$
$$\ell_i = u_i - 2\beta, \tag{4.5}$$

where α is the competitive ratio of `DTPR-min` *defined in Equation* (4.9).

The role of these thresholds is to incorporate the switching cost into the algorithm's decisions, and to alter the acceptance criteria of `DTPR-min` based on the current state. For OPR-min, the current state is *whether the previous item was accepted*, i.e., whether x_{t-1} is 0 or 1. As prices are sequentially revealed to the algorithm at each time t, the ith price accepted by `DTPR-min` will be the first price which is at most ℓ_i if $x_{t-1} = 0$, or at most u_i if $x_{t-1} = 1$. We note that L does not explicitly appear in this definition. As i approaches k, the values of these thresholds decrease, getting closer to L (See Figure 4.1). Note that, as indicated in Line 4, `DTPR-min` may be forced to accept the last prices of the sequence, which can be "worse" than the current threshold values, to satisfy the deadline constraint of `OPR`. Since T (the deadline) does not appear explicitly in the threshold definition, our analysis can handle the case where T is not known to the online player, and the forced acceptance is triggered by some external signal.

The `DTPR-max` algorithm Pseudocode is summarized in the appendix, in Algorithm 9. The logical flow of `DTPR-max` shares a similar structure to that of `DTPR-min`, with a few important differences highlighted here. For OPR-max, the ith price accepted by `DTPR-max` will be the first price which is at least u_i if $x_{t-1} = 0$, or at least ℓ_i if $x_{t-1} = 1$. Further, the threshold functions are defined as follows.

Definition 4.3.2 (`DTPR-max` Threshold Values)**.** *For each integer i on the interval $[1, k]$, the following expressions give the corresponding threshold values of ℓ_i and u_i for* `DTPR-max`.

$$\ell_i = L\left[1 + (\omega - 1)\left(1 + \frac{\omega}{k}\right)^{i-1}\right] - 2\beta\left[\left(\frac{\omega}{k} - \frac{1}{k} + 1\right)\left(1 + \frac{\omega}{k}\right)^{i-1}\right],$$
$$u_i = \ell_i + 2\beta, \tag{4.6}$$

where ω is the competitive ratio of `DTPR-max` *defined in Equation* (4.10).

In Figures 4.1 and 4.2, we plot threshold values for `DTPR-min` and `DTPR-max`, respectively, using example parameters of $U = 30, L = 5, k = 10$, and $\beta = 3$.

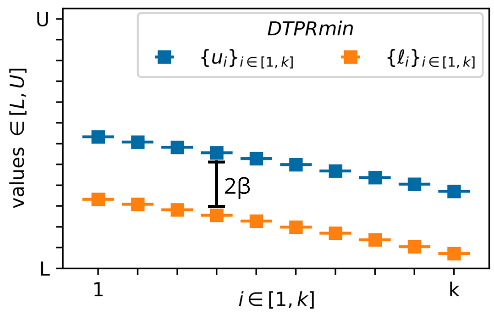


Figure 4.1: DTPR-min thresholds ℓ_i and u_i for $i \in [1, k]$ plotted using example parameters ($k = 10$).

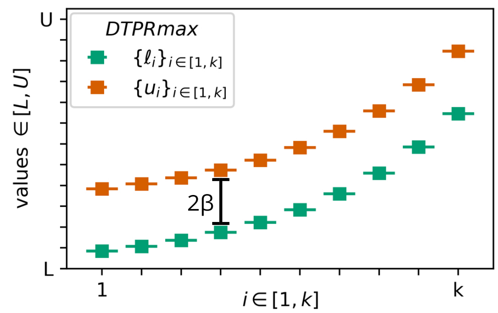


Figure 4.2: DTPR-max thresholds u_i and ℓ_i for $i \in [1, k]$ plotted using example parameters ($k = 10$).

We annotate the difference of 2β between ℓ_i and u_i; recall that each of these thresholds corresponds to a *current state* for DTPR, i.e., whether the previous item was accepted. Note that the DTPR-min threshold values *decrease* as k gets larger, while the DTPR-max threshold values *increase* as k gets larger. At a high-level, each ith threshold "hedges" against a scenario where none of the future prices meet the current threshold. In this case, even if the algorithm is forced to accept the *worst possible prices* at the end of the sequence, we want competitive guarantees against an offline OPT. Such guarantees rely on the fact that in the worst-case, OPT cannot accept prices that are all significantly better than DTPR's ith "unseen" threshold value because such prices did not exist in the sequence.

Designing the Double Threshold Values

A key component of the DTPR algorithms for both variants are the thresholds in Equations (4.5) and (4.6). The key idea is to design the thresholds by incorporating the switching cost into the balancing rules as a hedge against possible worst-case scenarios. To accomplish this, we enumerate three difficult cases that DTPR may encounter. (CASE-1): Consider an input sequence where DTPR does not accept any prices before it is forced to accept the last k prices. Here, the enforced prices in the worst-case sequence will be U for OPR-min and L for OPR-max. This sequence occurs only if no price in the sequence meets the first threshold for acceptance. On the other hand, in the case that DTPR does accept prices before the end of the sequence, we can further divide the possible sequences into two extreme cases for the *switching cost* it incurs. (CASE-2): In one extreme, the algorithm incurs only the minimum switching cost of 2β, meaning that k contiguous prices are accepted by DTPR. (CASE-3): In the other extreme, DTPR incurs the maximum switching cost of $k2\beta$, meaning that k non-contiguous prices are accepted. Intuitively, in order for `DTPR` to be competitive in either of these extreme cases, the prices accepted in the latter case should be sufficiently "good" to absorb the extra switching cost of $(k-1)2\beta$.

Given the insight from these cases, we use can use the balancing rule (see Section 4.2) to derive the two threshold families. Let σ be any arbitrary sequence for OPR. Given these extreme input sequences, we now concretely show how to write the balancing rule equations. We consider the cases of `DTPR-min` and `DTPR-max` separately below.

Balancing equations for `DTPR-min` To balance between possible inputs for OPR-min, consider the following examples for three different values of $c_{\min}(\sigma) > \ell, \ell = \{\ell_1, \ell_2, \ell_3\}$. If $c_{\min}(\sigma) > \ell_i$, we know that `OPT` cannot do better than $k\ell_i + 2\beta$. Suppose that α is the target competitive ratio. Then each term in equation (4.7) corresponds to a different *case* (e.g., a possible input), and we solve for the threshold values by "balancing" between all of these possible cases:

$$
\begin{aligned}
\frac{\texttt{DTPR-min}(\sigma)}{\texttt{OPT}(\sigma)} &\leq \underbrace{\frac{kU+2\beta}{k\ell_1+2\beta}}_{c_{\min}(\sigma)>\ell_1} = \underbrace{\frac{\ell_1+(k-1)U+4\beta}{k\ell_2+2\beta} = \frac{u_1+(k-1)U+2\beta}{k\ell_2+2\beta}}_{c_{\min}(\sigma)>\ell_2} \\
&= \underbrace{\frac{\ell_1+\ell_2+(k-2)U+6\beta}{k\ell_3+2\beta} = \frac{\ell_1+u_2+(k-2)U+4\beta}{k\ell_3+2\beta} = \frac{u_1+u_2+(k-2)U+2\beta}{k\ell_3+2\beta}}_{c_{\min}(\sigma)>\ell_3} \\
&= \cdots = \alpha. \qquad (4.7)
\end{aligned}
$$

As an example, consider $c_{\min}(\sigma) > \ell_2$ and the corresponding cases enumerated above. Suppose `DTPR-min` accepts one price before the end of the sequence σ, and the other prices accepted are all U. In the first case, where the competitive ratio is $\frac{\ell_1+(k-1)U+4\beta}{k\ell_2+2\beta}$, we consider the scenario where `DTPR-min` switches twice: once to accept the price ℓ_1, and once to accept $(k-1)$ prices at the end of the sequence, incurring switching cost of 4β.

In the second case, where the competitive ratio is $\frac{u_1+(k-1)U+2\beta}{k\ell_2+2\beta}$, we consider the hypothetical scenario where `DTPR-min` only switches once to accept some value u_1 followed by $(k-1)$ prices at the end of the sequence, incurring switching cost of 2β. By enumerating cases in this fashion for the other possible values of $c_{\min}(\sigma)$, we derive a relationship between the lower thresholds ℓ_i and the upper thresholds u_i in terms of the switching cost.

Balancing equations for $\texttt{DTPR-max}$ The same idea extends to balance between possible inputs for OPR-max. Consider the following examples for a few values of $c_{\max}(\sigma)$. If $c_{\max}(\sigma) < u_i$, we know that $\texttt{OPT}$ cannot do better than $ku_i - 2\beta$. Suppose that ω is the target competitive ratio, and we balance between the following possible cases (e.g., possible inputs):

$$\begin{aligned}
&\frac{\texttt{OPT}(\sigma)}{\texttt{DTPR-max}(\sigma)} \leq \underbrace{\frac{ku_1 - 2\beta}{kL - 2\beta}}_{c_{\max}(\sigma)<u_1} = \underbrace{\frac{ku_2 - 2\beta}{u_1 + (k-1)L - 4\beta} = \frac{ku_2 - 2\beta}{\ell_1 + (k-1)L - 2\beta}}_{c_{\max}(\sigma)<u_2} \\
&= \underbrace{\frac{ku_3 - 2\beta}{u_1 + u_2 + (k-2)L - 6\beta} = \frac{ku_3 - 2\beta}{u_1 + \ell_2 + (k-2)L - 4\beta} = \frac{ku_3 - 2\beta}{\ell_1 + \ell_2 + (k-2)L - 2\beta}}_{c_{\max}(\sigma)<u_3} \\
&= \cdots = \omega.
\end{aligned} \tag{4.8}$$

Solving for the threshold values Given the above balancing equations for both the minimization and maximization variants, the next step is to solve for the unknown values of ℓ_i and u_i. The following observation summarizes the key insight that enables this. We show that one can express each ℓ_i in terms of u_i and β, which facilitates the analysis required to solve for thresholds in each balancing equation (given by Equations (4.7) and (4.8)).

Observation 4.3.3. *By letting $u_i = \ell_i + 2\beta \quad \forall i \in [1, k]$, we obtain each possible worst-case permutation of ℓ_i thresholds, u_i thresholds, and switching cost. Let $y \in [1, k-1]$ denote the number of switches incurred by $\texttt{DTPR}$.*
For $\texttt{DTPR-min}$, suppose that $c_{\min}(\sigma) > \ell_{j+1}$. By the definition of $\texttt{DTPR-min}$, we know that accepting any u_i helps avoid a switching cost of $+2\beta$ in the worst case. Thus,

$$\begin{aligned}
&\sum_{i=0}^{j} u_i + (k-j)U + 2\beta = \underbrace{\ell_i + \ldots}_{y} + \underbrace{u_i + \ldots}_{j-y} + (k-j)U + (y+1)2\beta \\
&= \sum_{i=0}^{j} \ell_i + (k-j)U + (j+1)2\beta.
\end{aligned}$$

For $\texttt{DTPR-max}$, suppose that $c_{\max}(\sigma) < u_{j+1}$. By the definition of $\texttt{DTPR-max}$, we know that accepting any ℓ_i helps avoid a switching cost of -2β in the worst case.

Thus,

$$\sum_{i=0}^{j} \ell_i + (k-j)L - 2\beta = \underbrace{u_i + \ldots +}_{y} \underbrace{\ell_i + \ldots +}_{j-y} (k-j)L - (y+1)2\beta$$

$$= \sum_{i=0}^{j} u_i + (k-j)L - (j+1)2\beta.$$

With the above observation, for DTPR-min, one can substitute $u_i - 2\beta$ for each ℓ_i. By comparing adjacent terms in Equation (4.7), standard algebraic manipulations give a closed form for each u_i in terms of u_1. Setting $\frac{kU+2\beta}{k(u_1-2\beta)+2\beta} = \alpha$, we obtain the explicit expression for u_1, yielding a closed formula for $\{u_i\}_{i\in[1,k]}$ and $\{\ell_i\}_{i\in[1,k]}$ in Equation (4.5). Considering the balancing rule in Equation (4.7) for the case where $c_{\min}(\sigma) \geq \ell_{k+1}$, it follows that $\ell_{k+1} = L$, and thus $u_{k+1} = L + 2\beta$. By substituting this value into Definition 4.3.1, we obtain an explicit expression for α as shown in Equation (4.9).

Conversely, for DTPR-max, we substitute $\ell_i + 2\beta$ for each u_i. By comparing adjacent terms in Equation (4.8), standard methods give a closed form for each ℓ_i in terms of ℓ_1. Setting $\frac{k(\ell_1+2\beta)-2\beta}{kL-2\beta} = \omega$, we obtain the explicit expression for ℓ_1, yielding the closed formula for $\{\ell_i\}_{i\in[1,k]}$ and $\{u_i\}_{i\in[1,k]}$ in Equation (4.6). Considering the balancing rule in Equation (4.8) for the case where $c_{\max}(\sigma) \leq u_{k+1}$, it follows that $u_{k+1} = U$, and thus $\ell_{k+1} = U - 2\beta$. By substituting this value into Definition 4.3.2, we obtain an explicit expression for ω as shown in Equation (4.10).

4.4 Main Results

We now present competitive results of DTPR for both variants of OPR and discuss the significance of the results in relation to other algorithms for related problems. Our results for the competitive ratios of DTPR-min and DTPR-max are summarized in Theorems 4.4.1 and 4.4.2. We also state the lower bound results for any deterministic online algorithms for OPR-min and OPR-max in Theorems 4.4.5 and 4.4.6. Proofs of the results for DTPR-min and DTPR-max are deferred to Section 4.5 and Section 4.B, respectively. Formal proofs of lower bound theorems are given in Section 4.D, and a sketch is shown in Section 4.5. Note that in the competitive results, $W(x)$ denotes the Lambert W function, i.e., the inverse of $f(x) = xe^x$. It is well-known that $W(x)$ behaves like $\ln(x)$ for large x [172, 173]. We start by presenting our competitive bounds on DTPR-min and DTPR-max.

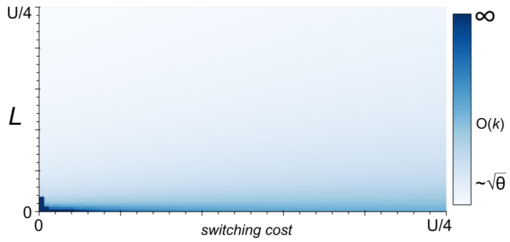


Figure 4.3: `DTPR-min`: Plotting actual values of competitive ratio α for fixed $k \geq 1$, fixed $U > L$, and varying values for L and β (switching cost). Color represents the order of α for a given setting of θ and β.

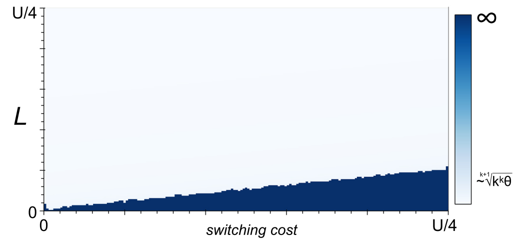


Figure 4.4: `DTPR-max`: Plotting actual values of competitive ratio ω for fixed $k \geq 1$, fixed $U > L$, and varying values for L and β (switching cost). Color represents the order of ω for a given setting of θ and β.

Theorem 4.4.1. *`DTPR-min` is an α-competitive deterministic algorithm for `OPR-min`, where α is the unique positive solution of*

$$\frac{U - L - 2\beta}{U(1 - 1/\alpha) - \left(2\beta - \frac{2\beta}{k} + \frac{2\beta}{k\alpha}\right)} = \left(1 + \frac{1}{k\alpha}\right)^k. \tag{4.9}$$

Theorem 4.4.2. *`DTPR-max` is an ω-competitive deterministic algorithm for `OPR-max`, where ω is the unique positive solution of*

$$\frac{U - L - 2\beta}{L(\omega - 1) - 2\beta\left(1 - \frac{1}{k} + \frac{\omega}{k}\right)} = \left(1 + \frac{\omega}{k}\right)^k. \tag{4.10}$$

These theorems present upper bounds on the competitive ratios, showing their dependence on the problem parameters. To investigate the behavior of these competitive ratios, in Figures 4.3 and 4.4, we show the competitive ratios of both algorithms as problem parameters are varied. More specifically, in Figure 4.3, we visualize α as a function of β and L, where k and U are fixed. The color (shown as an annotated color bar on the right-hand side of the plot) represents the order of α. If $\beta > 0$ and $L \to 0$, Figure 4.3 shows that α is roughly $O(k)$, which we discuss further in Corollary 4.4.3(a). In Figure 4.4, we visualize ω as a function of β and L, where k and U are fixed. The color represents the order of ω. In the dark blue region of the plot, Figure 4.4 shows that $\omega \to \infty$ when $b = \frac{2\beta}{L} \to k$, which provides insight into the extreme case for switching cost when $\beta \gtrsim \frac{kL}{2}$.

To obtain additional insight into the form of the competitive ratios in Theorems 4.4.1 and 4.4.2, we present the following corollaries for two asymptotic regimes of interest: `REGIME-1` captures the order of the competitive ratio when k is fixed and

α or ω are sufficiently large, and `REGIME-2` captures the order of the competitive ratio when $k \to \infty$.

Corollary 4.4.3. ***(a)*** *For* `REGIME-1`*, with fixed $k \geq 1$ and $\beta \in (0, \frac{U-L}{2})$, the competitive ratio of* `DTPR-min` *is*

$$\alpha \sim \frac{k\beta}{kL + 2\beta} + \sqrt{\frac{k^2LU + 2kL\beta + 2kU\beta + 4\beta^2 + k^2\beta^2}{k^2L^2 + 4kL\beta + 4\beta^2}}, \quad \text{and } \alpha \sim O(k) \text{ for } L \to 0.$$

(b) *Furthermore, for* `REGIME-2`*, with $k \to \infty$ and $c = \frac{2\beta}{U}, c \in (0, \frac{U-L}{U})$, the competitive ratio of* `DTPR-min` *is*

$$\alpha \sim \left[W\left(\frac{\left(c + \frac{1}{\theta} - 1\right) e^c}{e} \right) - c + 1 \right]^{-1}.$$

Corollary 4.4.4. ***(a)*** *For* `REGIME-1`*, with fixed $k \geq 1$ and $b = \frac{2\beta}{L}, b \in (0, k)$, the competitive ratio of* `DTPR-max` *is*

$$\omega \sim O\left(\sqrt[k+1]{k^k \frac{k\theta}{k-b}} \right),$$

and ***(b)*** *for* `REGIME-2`*, with $k \to \infty$ and $b = \frac{2\beta}{L}, b \in (0, k)$, the competitive ratio of* `DTPR-max` *is*

$$\omega \sim W\left(\frac{\theta - 1 - b}{e^{1+b}} \right) + 1 + b.$$

Corollary 4.4.3(a) contextualizes the behavior of α (the competitive ratio of `DTPR-min`) in the most relevant OPR-min setting (when $\beta \in (0, \frac{U-L}{2})$). Note that in this minimization setting, as β grows, the competitive ratio improves. Let us also briefly discuss the other cases for the switching cost β, and why this interval makes sense. When $\beta > \frac{U-L}{2}$, the switching cost is large enough such that `OPT` only incurs a switching cost of 2β. In this regime, α does not fully capture the competitive ratio of `DTPR-min`, since every value in the threshold family $\{u_i\}_{i \in [1,k]}$ is at least U; in other words, whenever the algorithm begins accepting prices, it will accept k prices in a single continuous segment, incurring minimal switching cost of 2β. As $\beta \to \infty$, the competitive ratio of `DTPR-min` approaches 1. This theoretical result corresponds nicely with the empirical observation in [158] that a large switching overhead can nullify carbon emission reductions from temporal shifting if the job is interrupted frequently.

Conversely, Corollary 4.4.4(a) contextualizes the behavior of ω in the most relevant OPR-max setting (when $\beta \in (0, \frac{kL}{2})$), but we also discuss the other cases for the switching cost β, and why this interval makes sense. When $\beta \geq \frac{kL}{2}$, the switching cost is too large, and the competitive ratio may become unbounded. Note that this is shown explicitly in Figure 4.4. Consider an adversarial sequence which forces any OPR-max algorithm to accept k prices with value L at the end of the sequence. On such a sequence, even a player which incurs the minimum switching cost of 2β achieves zero or negative profit of $kL - 2\beta \leq 0$, and this is not well-defined.

Next, to begin to investigate the tightness of Theorems 4.4.1 and 4.4.2, it is interesting to consider special cases that correspond to models studied in previous work. In particular, when $\beta = 0$, i.e., there is no switching cost, OPR degenerates to the k-search problem [79]. For fixed $k \geq 1$ and $\theta \to \infty$, the optimal competitive ratios shown by [79] are $\sqrt{\theta/2}$ for k-min, and $\sqrt[k+1]{k^k\theta}$ for k-max (see Section 4.2).

Both versions of DTPR exactly recover the optimal k-search algorithms [79].[2] Figure 4.3 shows that if $\beta = 0$ and $L \to 0$, then $\alpha \to \infty$, which matches the k-min result of $\sqrt{\theta/2} \sim \infty$. Similarly, Figure 4.4 shows that if $\beta = 0$ and $L \to 0$, then $\omega \to \infty$, which matches the k-max result of $\sqrt[k+1]{k^k\theta} \sim \infty$.

More generally, one can ask if the competitive ratios of DTPR can be improved upon by other online algorithms outside of the special case of k-search. Our next set of results highlights that no improvement is possible, i.e., that `DTPR-min` and `DTPR-max` maintain the optimal competitive ratios possible for any deterministic online algorithm for OPR.

Theorem 4.4.5. *Let $k \geq 1$, $\theta \geq 1$, and $\beta \in (0, \frac{U-L}{2})$. Then α given by Equation (4.9) is the best competitive ratio that a deterministic online algorithm for OPR-min can achieve.*

Theorem 4.4.6. *Let $k \geq 1$, $\theta \geq 1$, and $\beta \in (0, \frac{kL}{2})$. Then ω given by Equation (4.10) is the best competitive ratio that a deterministic online algorithm for OPR-max can achieve.*

[2] To see this, note that by eliminating all β terms from Equations (4.9) and (4.10), we exactly recover Equations (4.3) and (4.4), which are the definitions of the k-search algorithms. When $\theta \to \infty$ as $L \to 0$, `DTPR-min` and `DTPR-max` match each k-search result exactly when $\beta = 0$. In Corollaries 4.4.3(b) and 4.4.4(b), `DTPR-min` and `DTPR-max` also match each k search result exactly when $k \to \infty$ and $\beta = 0$. (See Sec. 4.2)

By combining Theorems 4.4.1 and 4.4.2 with Theorems 4.4.5 and 4.4.6, these results imply that the competitive ratios of `DTPR-min` and `DTPR-max` are optimal for OPR-min and OPR-max.

Finally, it is interesting to contrast the upper and lower bounds for OPR with those for k-search, since the contrast highlights the impact of switching costs. In OPR-min, when $\beta > 0$, we find that the `DTPR-min` competitive results *improve* on optimal results for k-min search (where $\beta = 0$ is assumed), particularly in the case where L approaches 0 (i.e., $\theta \to \infty$). Since Theorem 4.4.5 implies that `DTPR-min` is optimal, this shows that the addition of switching cost in OPR-min enables an online algorithm to achieve a better competitive ratio compared to k-min search, which is a surprising result. In contrast, for OPR-max with $\beta > 0$, `DTPR-max`'s competitive bounds are *worse* than existing results for k-max search, particularly for large β. Since Theorem 4.4.6 implies that `DTPR-max` is optimal, this suggests that OPR-max is fundamentally a *more difficult* problem compared to k-max search.

We note that although the lower bounds shown in Theorems 4.4.5 and 4.4.6 specifically apply to deterministic algorithms, there are lower bounds in the literature for randomized k-search [79]. The randomized bound for k-min search are not an order-improvement over the deterministic lower bound, while the randomized results for k-max search improve the lower bound to $\Omega(\ln \theta)$. However, in the regimes of k which are interesting for applications (where k is sufficiently large), there will be a small difference between the deterministic upper bound and the randomized lower bound in practice. Combined, these results for k-search suggest that randomization similarly may not yield large improvements in the OPR setting. Exploring this dynamic further for OPR is an interesting direction for future work.

4.5 Proofs

We now prove the results described in the previous section. In Section 4.5, we prove the `DTPR-min` results presented in Theorem 4.4.1 and Corollary 4.4.3. In Section 4.5, we provide a proof sketch for the lower bound results in Theorems 4.4.5 and 4.4.6, and defer the formal proofs to Section 4.D. The competitive results for `DTPR-max` in Theorem 4.4.2 and Corollary 4.4.4 are deferred to Section 4.B.

Competitive Results for `DTPR-min`

We begin by proving Theorem 4.4.1 and Corollary 4.4.3. The key novelty in the proof of the main competitive results (Theorems 4.4.1 and 4.4.2) lies in our effort to derive two threshold functions and balance the competitive ratio in several worst-case instances with respect to these thresholds, as outlined in Section 4.3.

Proof of Theorem 4.4.1. For $0 \le j \le k$, let $\mathcal{S}_j \subseteq \mathcal{S}$ be the sets of OPR-min price sequences for which `DTPR-min` accepts exactly j prices (excluding the $k-j$ prices it is forced to accept at the end of the sequence). Then, all of the possible price sequences for OPR-min are represented by $\mathcal{S} = \bigcup_{j=0}^{k} \mathcal{S}_j$. Also, recall that by definition, $\ell_{k+1} = L$. Let $\epsilon > 0$ be a fixed constant, and define the following two price sequences σ_j and ρ_j:

$$\forall j \in [2,k] : \sigma_j = \ell_1, u_2, \dots, u_j, U, \underbrace{\ell_{j+1}+\epsilon, \dots, \ell_{j+1}+\epsilon}_{k}, \underbrace{U, U, \dots, U}_{k}.$$

$$\forall j \in [2,k] : \rho_j = \ell_1, U, \ell_2, U, \dots, U, \ell_j, U, \underbrace{\ell_{j+1}+\epsilon, \dots, \ell_{j+1}+\epsilon}_{k}, \underbrace{U, U, \dots, U}_{k}.$$

There are two special cases for $j = 0$ and $j = 1$. For $j = 0$, we have that $\sigma_0 = \rho_0$, and this sequence simply consists of $\ell_1 + \epsilon$ repeated k times, followed by U repeated k times. For $j = 1$, we also have that $\sigma_1 = \rho_1$, and this sequence consists of one price with value ℓ_1 and one price with value U, followed by $\ell_2 + \epsilon$ repeated k times and U repeated k times.

Observe that as $\epsilon \to 0$, σ_j and ρ_j are sequences yielding the worst-case ratios in $\mathcal{S}_j$, as `DTPR-min` is forced to accept $(k-j)$ worst-case U values at the end of the sequence, and each accepted value is exactly equal to the corresponding threshold.

Note that σ_j and ρ_j also represent two extreme possibilities for the additive switching cost. In σ_j, `DTPR-min` only switches twice, but it mostly accepts values u_i. In ρ_j, `DTPR-min` must switch $j+1$ times because there are many intermediate U values, but it only accepts values ℓ_i.

In the worst case, we have

$$\frac{\texttt{DTPR-min}(\sigma_j)}{\texttt{OPT}(\sigma_j)} = \frac{\texttt{DTPR-min}(\rho_j)}{\texttt{OPT}(\rho_j)}.$$

Also, the optimal solutions for both sequences are lower bounded by the same quantity: $kc_{\min}(\sigma_j) + 2\beta = kc_{\min}(\rho_j) + 2\beta$. For any sequence s in $\mathcal{S}_j$, we have that $c_{\min}(s) > \ell_{j+1}$, so $\texttt{OPT}(\rho_j) = \texttt{OPT}(\sigma_j) \le k\ell_{j+1} + 2\beta$.

By definition of the threshold families $\{\ell_i\}_{i\in[1,k]}$ and $\{u_i\}_{i\in[1,k]}$, we know that

$\sum_{i=1}^{j} \ell_i + j2\beta = \sum_{i=1}^{j} u_i$ for any value $j \geq 2$:

$$\begin{aligned}
\texttt{DTPR-min}(\rho_j) &= \left(\sum_{i=1}^{j} \ell_i + (k-j)U + (j+1)2\beta \right) \\
&= \left(\ell_1 + \sum_{i=2}^{j} u_i + (k-j)U + 4\beta \right) \\
&= \texttt{DTPR-min}(\sigma_j).
\end{aligned}$$

Note that whenever $j < 2$, we have that $\sigma_0 = \rho_0$, and $\sigma_1 = \rho_1$. Thus, $\texttt{DTPR-min}(\rho_j) = \texttt{DTPR-min}(\sigma_j)$ holds for any value of j. By definition of ℓ_1, we simplify $\ell_1 + \sum_{i=2}^{j} u_i + (k-j)U + 4\beta$ to $\sum_{i=1}^{j} u_i + (k-j)U + 2\beta$. Then, for any sequence $s \in \mathcal{S}_j$, we have the following:

$$\begin{aligned}
\frac{\texttt{DTPR-min}(s)}{\texttt{OPT}(s)} &\leq \frac{\texttt{DTPR-min}(\sigma_j)}{\texttt{OPT}(\sigma_j)} \\
&= \frac{\texttt{DTPR-min}(\rho_j)}{\texttt{OPT}(\rho_j)} \\
&\leq \frac{\sum_{i=1}^{j} u_i + (k-j)U + 2\beta}{k\ell_{j+1} + 2\beta}.
\end{aligned} \tag{4.11}$$

Before proceeding to the next step, we use an intermediate result stated in the following lemma with a proof given in Section 4.C.

Lemma 4.5.1. *For any $0 \leq j \leq k$, by definition of $\{\ell_i\}_{i\in[1,k]}$ and $\{u_i\}_{i\in[1,k]}$,*

$$\sum_{i=1}^{j} u_i + (k-j)U + 2\beta \leq \alpha \cdot (k\ell_{j+1} + 2\beta).$$

For $\epsilon \longrightarrow 0$, the competitive ratio $\texttt{DTPR-min}/\texttt{OPT}$ is exactly α:

$$\forall 0 \leq j \leq k : \quad \frac{\texttt{DTPR-min}(\sigma_j)}{\texttt{OPT}(\sigma_j)} = \frac{\sum_{i=1}^{j} u_i + (k-j)U + 2\beta}{k\ell_{j+1} + 2\beta} = \alpha,$$

and thus for any sequence $s \in \mathcal{S}$,

$$\forall s \in \mathcal{S} : \quad \frac{\texttt{DTPR-min}(s)}{kc_{\min}(s) + 2\beta} \leq \alpha.$$

Since $\texttt{OPT}(s) \geq kc_{\min}(s) + 2\beta$ for any sequence s, this implies that $\texttt{DTPR-min}$ is α-competitive. $\square$

Proof of Corollary 4.4.3. To show part **(a)** for `REGIME-1`, with fixed $k \geq 1$, observe that we can expand the right-hand side of Equation (4.9) using the binomial theorem to obtain the following:

$$\frac{U - L - 2\beta}{U\left(1 - \frac{1}{\alpha}\right) - 2\beta\left(1 - \frac{1}{k} + \frac{1}{k\alpha}\right)} = 1 + \frac{1}{\alpha} + \Theta\left(\alpha^{-2}\right).$$

Next, observe that $\alpha^\star$ solving the following expression satisfies $\alpha^\star \geq \alpha \;\; \forall k : k \geq 1$, (i.e., $\alpha^\star$ is an upper bound of α):

$$\frac{U - L - 2\beta}{U\left(1 - \frac{1}{\alpha^\star}\right) - 2\beta\left(1 - \frac{1}{k} + \frac{1}{k\alpha^\star}\right)} = 1 + \frac{1}{\alpha^\star}.$$

By solving the above for $\alpha^\star$, we obtain

$$\alpha \sim \alpha^\star = \frac{k\beta}{kL + 2\beta} + \sqrt{\frac{k^2LU + 2kL\beta + 2kU\beta + 4\beta^2 + k^2\beta^2}{k^2L^2 + 4kL\beta + 4\beta^2}}.$$

Last, note that as $L \longrightarrow 0$, we obtain the following result:

$$\alpha \sim \frac{k}{2} + \sqrt{\frac{kU}{2\beta} + 1 + \frac{k^2}{4}} \approx O\left(k\right).$$

To show part **(b)** for `REGIME-2`, we first observe that the right-hand side of Equation 4.9 can be approximated as $\left(1 + \frac{1}{k\alpha}\right)^k \approx e^{1/\alpha}$ when $k \longrightarrow \infty$. Then by taking limits on both sides, we obtain the following:

$$\frac{U - L - 2\beta}{U\left(1 - \frac{1}{\alpha}\right) - 2\beta\left(1\right)} = e^{1/\alpha}.$$

For simplification purposes, let $\beta = cU/2$, where c is a small constant on the interval $\left(0, \frac{U-L}{U}\right)$. We then obtain the following:

$$\frac{U - L - cU}{U\left(1 - \frac{1}{\alpha}\right) - cU} = e^{1/\alpha} \Longrightarrow L/U + c - 1 = \left(\frac{1}{\alpha} + c - 1\right)e^{1/\alpha}.$$

By definition of Lambert W function, solving this equation for α obtains the result in Corollary 4.4.3(b). $\square$

Lower Bound Analysis: Proof of Theorem 4.4.5 (OPR-min Lower Bound)

In Theorems 4.4.5 and 4.4.6, we state that *any* deterministic strategy achieves a competitive ratio of at least α for OPR-min, and at least ω for OPR-max. In this section, we formalize the lower bound construction which proves Theorem 4.4.5. A similar construction is used to prove Theorem 4.4.6 in Section 4.D. These two results jointly imply that our proposed `DTPR` algorithms are both optimal.

Proof of Theorem 4.4.5. Let ALG be a deterministic online algorithm for OPR-min, and suppose that the adversary uses the price sequence $\ell_1, \ldots, \ell_k$, which is exactly the sequence defined by (4.5). ℓ_1 is presented to ALG, at most k times or until ALG accepts it. If ALG never accepts ℓ_1, the remainder of the sequence is all U, and ALG achieves a competitive ratio of $\frac{kU+2\beta}{k\ell_1+2\beta} = \alpha$, as defined in (4.7).

If ALG accepts ℓ_1, the next price presented is U, repeated at most k times *or until ALG switches to reject* U. After ALG has switched, ℓ_2 is presented to ALG, at most k times or until ALG accepts it. Again, if ALG never accepts ℓ_2, the remainder of the sequence is all U, and ALG achieves a competitive ratio of at least $\frac{\ell_1+(k-1)U+4\beta}{k\ell_2+2\beta} = \alpha$, as defined in (4.7).

As the sequence continues, whenever ALG does not accept some ℓ_i after it is presented k times, the adversary increases the price to U for the remainder of the sequence. Otherwise, if ALG accepts k prices before the end of the sequence, the adversary concludes by presenting L at least k times.

Observe that any ALG which does not immediately reject the first U presented to it after accepting some ℓ_i obtains a competitive ratio strictly worse than α. To illustrate this, suppose ALG has just accepted ℓ_1, incurring a cost of $\ell_1 + \beta$ so far. The adversary begins to present U, and ALG accepts $y \leq (k-1)$ of these U prices before switching away. If $y = (k-1)$, ALG will accept k prices before the end of the sequence and achieve a competitive ratio of $\frac{\ell_1+(k-1)U+2\beta}{kL+2\beta} > \alpha$. Otherwise, if $y < (k-1)$, the cost incurred by ALG so far is at least $\ell_1 + 2\beta + yU$, while the cost incurred by ALG if it had immediately switched away ($y = 0$) would be $\ell_1 + 2\beta$—since any price which might be accepted by ALG in the future should be $\leq U$, the latter case strictly improves the competitive ratio of ALG.

Assuming that ALG does immediately reject any U presented to it, and that ALG accepts some prices before the end of the sequence, the competitive ratio attained by ALG is at least $\frac{\sum_{i=1}^{j} \ell_i+(j+1)2\beta+(k-j)U}{k\ell_{j+1}+2\beta} = \alpha$, as defined in (4.7).

Similarly, if ALG accepts k prices before the end of the sequence, the competitive ratio attained by ALG is at least $\frac{\sum_{i=1}^{k} \ell_i+k2\beta}{kL+2\beta} = \alpha$, as defined in (4.7).

Since any arbitrary deterministic online algorithm ALG cannot achieve a competitive ratio better than α playing against this adaptive adversary, our proposed algorithm DTPR-min is optimal. $\square$

4.6 Case Study: Carbon-Aware Temporal Workload Shifting

We now present experimental results for the `DTPR` algorithms in the context of the carbon-aware temporal workload shifting problem. We evaluate `DTPR-min` (and `DTPR-max` in Section 4.A) against existing algorithms from the literature that have been adapted for OPR.

Experimental Setup

We consider a carbon-aware load shifting system that operates on a hypothetical datacenter. An algorithm is given a deferrable and interruptible job that takes k time slots to complete, along with a deadline $T \geq k$, such that the job must be completed at most T slots after its arrival. The objective is to selectively run units of the job such that the total carbon emissions are minimized while still completing the job before its deadline.

For the minimization variant (OPR-min) of the experiments, we consider *carbon emissions intensities*, as the price values. At each timestep t, the electricity supply has a carbon intensity c_t, i.e., if the job is being processed during the timestep t ($x_t = 1$), the datacenter's carbon emissions during that timestep are proportional to c_t. If the job is *not* being processed during the timestep t ($x_t = 0$), we assume for simplicity that carbon emissions in the idle state are negligible and essentially 0. To model the combined computational overhead of interrupting, checkpointing, and restarting the job, the algorithm incurs a fixed switching cost of β whenever $x_{t-1} \neq x_t$, whose values are selected relative to the price values.

Carbon data traces We use real-world carbon traces from Electricity Maps [171], which provide time-series information about the *average carbon emissions intensity* of the electric grid. We use traces from three different regions: the Pacific Northwest of the U.S., New Zealand, and Ontario, Canada. The data is provided at an hourly granularity and includes the current average carbon emissions intensity in grams of CO_2 equivalent per kilowatt-hour (gCO_2eq/kWh), and the percentage of electricity being supplied from carbon-free sources. In Figure 4.9 (in Section 4.A), we plot three representative actual traces for carbon intensity over time for a 96-hour period in each region.

Parameter settings We test for time horizons (T) of 48 hours, 72 hours, and 96 hours. The chosen time horizon represents the time at which the job with length k

Table 4.2: Summary of carbon trace data sets

Location	**Pacific NW, U.S.**	**New Zealand**	**Ontario, Canada**
Number of Data Points	10,144	1,324	17,898
Max. Carbon Intensity (U)	648 gCO_2eq/kWh	165 gCO_2eq/kWh	181 gCO_2eq/kWh
Min. Carbon Intensity (L)	18 gCO_2eq/kWh	54 gCO_2eq/kWh	15 gCO_2eq/kWh
Duration (mm/dd/yy)	04/20/22 - 12/06/22	10/19/21 - 11/16/21	10/19/21 - 12/06/22

must be completed. As is given in the carbon trace data, we consider time slots of one hour.

The online algorithms we use in experiments take L and U as parameters for their threshold functions. To set these parameters, we examine the entire carbon trace for the current location. For the Pacific NW trace and the Ontario trace, these values represent lower and upper bounds of the carbon intensity values for a full year. For the New Zealand trace, these values are a lower and upper bound for the values during a month of data, which is reflected by a smaller fluctuation ratio. We set L and U to be the minimum and maximum observed carbon intensity over the entire trace.

To generate each input sequence, a contiguous segment of size T is randomly sampled from the given carbon trace. In a few experiments, we simulate greater *volatility* over time by "scaling up" each price's deviation from the mean. First, we compute the average value over the entire sequence. Next, we compute the difference between each price and this average. Each of these differences is scaled by a *noise factor* of $m \geq 1$. Finally, new carbon values are computed by summing each scaled difference with the average. If $m = 1$, we recover the same sequence, and if $m > 1$, any deviation from the mean is proportionately amplified. Any values which become negative after applying this transformation are truncated to 0. This technique allows us to evaluate algorithms under different levels of volatility. As we are in the regime where $L = 0$, none of the other online algorithms considered have competitive guarantees, since their competitive ratios become unbounded when $L \to 0$. Instead, our `DTPR` algorithm maintains its optimal bound defined in (4.9) and (4.10) due to the presence of switching cost β in the competitive bounds. Performance in the presence of greater carbon volatility is important, as on-site renewable generation is seeing greater adoption as a supplementary power source for datacenters [155, 156].

Benchmark algorithms To evaluate the performance of `DTPR`, we use a dynamic programming approach to calculate the offline optimal solution for each given se-

Table 4.3: Summary of algorithms tested in our OPR experiments

Algorithm	**Carbon-aware**	**Switching-aware**	**Description**
OPT (offline)	YES	YES	Optimal offline solution
Carbon-Agnostic	NO	YES	Runs job in the first k time slots
Const. Threshold	YES	NO	Runs job if carbon meets threshold $\sqrt{UL}$ [78]
k-search	YES	NO	Runs ith slot of job if carbon meets threshold Φ_i [79]
DTPR	YES	YES	This work (algorithms proposed in Section 4.3)

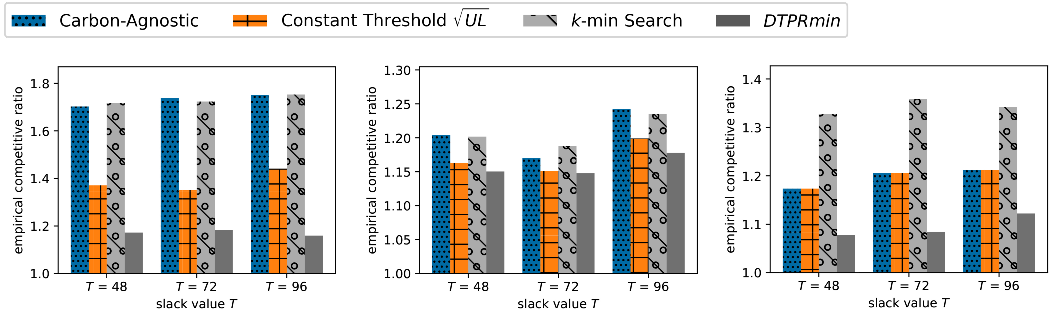
quence and objective, which allows us to report the empirical competitive ratio for each tested algorithm. We compare DTPR against two categories of benchmark algorithms, which are summarized in Table 4.3.

The first category of benchmark algorithms is *carbon-agnostic* algorithms, which run the jobs during the first k time slots in order, i.e., accepting prices $c_1, \ldots, c_k$. This approach incurs the minimal switching cost of 2β, because it does not interrupt the job while it is being processed. The carbon-agnostic approach simulates the behavior of a scheduler that runs the job to completion as soon as it is submitted, without any focus on reducing carbon emissions. Note that the performance of this approach significantly varies based on the randomly selected sequence, since it will perform well if low-carbon electricity is available in the first few slots, and will perform poorly if the first few slots are high-carbon.

We also compare DTPR against *switching-cost-agnostic* algorithms, which only consider carbon cost. We have two algorithms of this type, each drawing from existing online search methods in the literature. Although they do not consider the switching cost in their design, they still incur a switching cost whenever their decision in adjacent time slots differs.

The first such algorithm is a *constant threshold algorithm*, which uses the $\sqrt{UL}$ threshold value first presented for online search in [78]. In our minimization experiments, this algorithm runs the workload during the first k time slots where the carbon intensity is at most $\sqrt{UL}$.

The other switching-cost-agnostic algorithm tested is the k-search algorithm shown by [79] and described in Section 4.2. The k-min search algorithm chooses to run the ith hour of the job during the first time slot where the carbon intensity is at most Φ_i.



(a) Ontario, Canada (b) U.S. Pacific Northwest (c) New Zealand

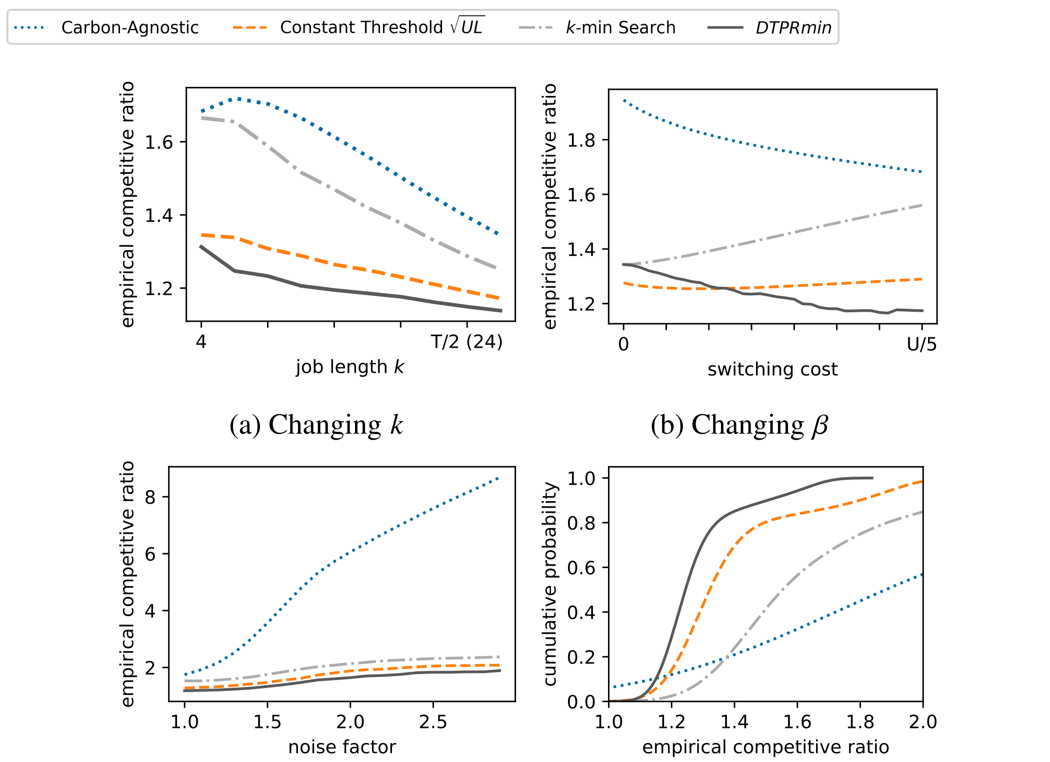
Figure 4.5: Experiments for three distinct *time horizons*, where $T \in \{48, 72, 96\}$. (a): Ontario, Canada carbon trace, with $\theta = 12.0\bar{6}$. (b): U.S. Pacific Northwest carbon trace, with $\theta = 36$. (c): New Zealand carbon trace, with $\theta = 3.0\bar{5}$.

Experimental Results

We now present our experimental results. Our focus is on the empirical competitive ratio (a lower competitive ratio is better). We report the performance of all algorithms for each experimental setting, in each tested region. Throughout the minimization experiments, we observe that `DTPR-min` outperforms the benchmark algorithms. The 95th percentile worst-case empirical competitive ratio achieved by `DTPR-min` is a 48.2% improvement on the carbon-agnostic method, a 15.6% improvement on the k-min search algorithm, and a 14.4% improvement on the constant threshold algorithm.

In Figure 4.5, we show results for three different values of horizon T in each carbon trace, with fixed $\beta \approx U/20$, fixed $k = \lceil T/6 \rceil$, and no added volatility. Although our experiments test three distinct values for T, we later observe that the *ratio between k and* T is the primary factor that changes the performance of the algorithms we test; in this figure, `DTPR` and the benchmark algorithms compare very similarly on the same carbon trace for different T values. As such, we set $T = 48$ in the rest of the experiments in this section for brevity. This represents a *time horizon* of 48 hours.

In the first experiment, we test all algorithms for different job lengths k in the range from 4 hours to $T/2$ (24 hours). The switching cost β is non-zero and fixed to $\approx U/20$, and no volatility is added to the carbon trace. By testing different values for k, this experiment tests different ratios between the workload length and the horizon provided to the algorithm. In Figures 4.6(a), 4.7(a), and 4.8(a), we show that the competitive ratio of `DTPR-min` outperforms others, and it compares particularly favorably for *short* job lengths. Averaging over all regions and job lengths, the competitive ratio achieved by `DTPR-min` is a 11.4% improvement on



(a) Changing k

(b) Changing β

(c) Changing volatility

(d) CDF

Figure 4.6: Experiments on Ontario, Canada carbon trace, with $\theta = 12.0\bar{6}$, and $T = 48$. (a): Changing job length k w.r.t. time horizon T (x-axis), vs. competitive ratio. (b): Changing switching cost β w.r.t. U (x-axis), vs. competitive ratio. (c): Different volatility levels w.r.t. U (x-axis), vs. competitive ratio. (d): Cumulative distribution function of competitive ratios.

the carbon-agnostic method, a 14.0% improvement on the k-min search algorithm, and a 5.5% improvement on the constant threshold algorithm.

In the second experiment, we test all algorithms for different switching costs β in the range from 0 to $U/5$. The job length k is set to 10 hours, and no volatility is added to the carbon trace. By testing different values for β, this experiment tests how an increasing switching cost impacts the performance of `DTPR-min` with respect to other algorithms which do not explicitly consider the switching cost. In Figures 4.6(b), 4.7(b), and 4.8(b), we show that the observed competitive ratio of `DTPR-min` outperforms the benchmark algorithms for most values of β in all regions. Unsurprisingly, the carbon-agnostic technique (which incurs minimal switching cost) performs better as β grows. While the constant threshold algorithm has relatively consistent performance, the k-min search algorithm performs noticeably worse as β grows. Averaging over all regions and switching cost values, the competitive ratio

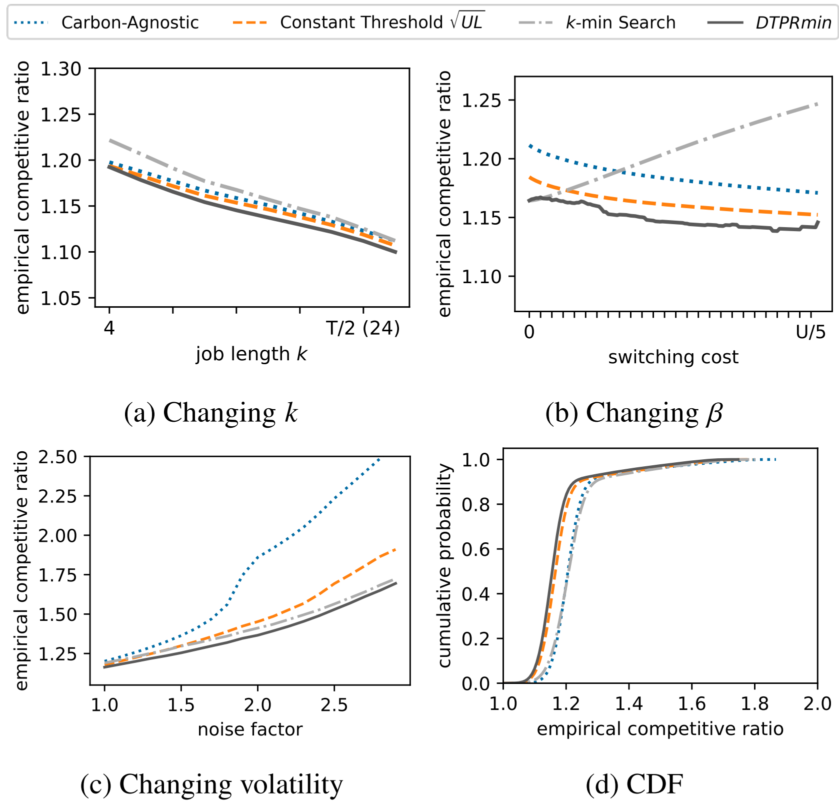
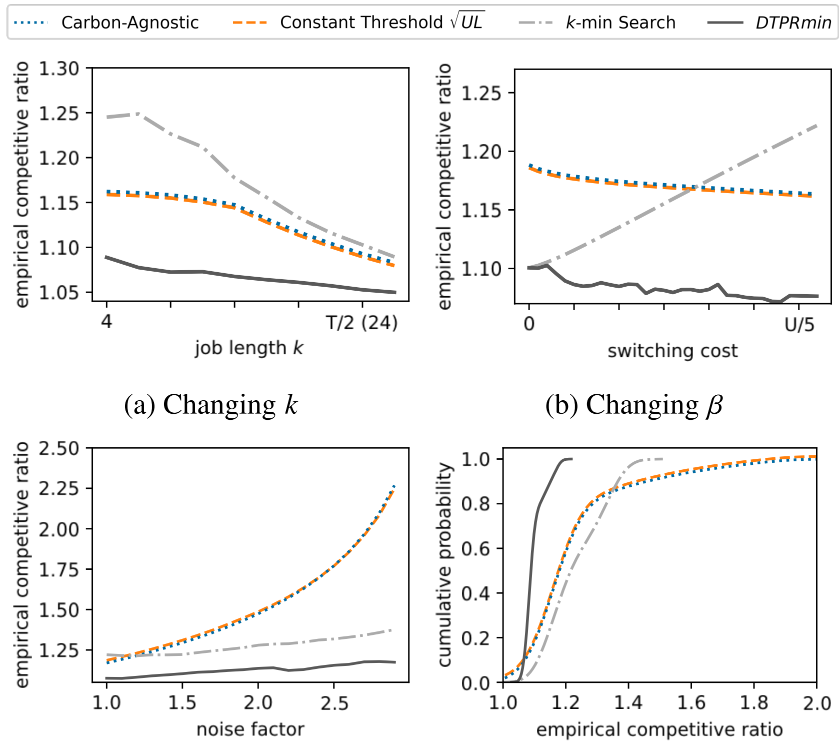


Figure 4.7: Experiments on U.S. Pacific Northwest carbon trace, with $\theta = 36$, and $T = 48$. (a): Changing job length k w.r.t. time horizon T (x-axis), vs. competitive ratio. (b): Changing switching cost β w.r.t. U (x-axis), vs. competitive ratio. (c): Different volatility levels w.r.t. U (x-axis), vs. competitive ratio. (d): Cumulative distribution function of competitive ratios.

achieved by `DTPR-min` is a 18.2% improvement on the carbon-agnostic method, a 8.9% improvement on the k-min search algorithm, and a 4.1% improvement on the constant threshold algorithm.

In the final experiment, we test all algorithms on sequences with different volatility. The job length k and switching cost β are both fixed as previously. We add volatility by setting a *noise factor* from the range 1.0 to 3.0. By testing different values for this volatility, this experiment tests how each algorithm handles larger fluctuations in the carbon intensity of consecutive timesteps. In Figures 4.6(c), 4.7(c), and 4.8(c), we show that the observed competitive ratio of `DTPR-min` outperforms the benchmark algorithms for all noise factors in all regions. Intuitively, higher volatility values cause the online algorithms to perform worse in general. Averaging over all regions and noise factors, the competitive ratio achieved by `DTPR-min` is a 53.6%



(a) Changing k

(b) Changing β

(c) Changing volatility

(d) CDF

Figure 4.8: Experiments on New Zealand carbon trace, with $\theta = 3.0\bar{5}$, and $T = 48$. *Note: the line for Carbon-Agnostic overlaps the line for Constant Threshold in some of the above plots.* (a): Changing job length k w.r.t. time horizon T (x-axis), vs. competitive ratio. (b): Changing switching cost β w.r.t. U (x-axis), vs. competitive ratio. (c): Different volatility levels w.r.t. U (x-axis), vs. competitive ratio. (d): Cumulative distribution function of competitive ratios.

improvement on the carbon-agnostic method, a 13.5% improvement on the k-min search algorithm, and a 14.3% improvement on the constant threshold algorithm.

By averaging over all experiments for a given region, we obtain the cumulative distribution function plot for each algorithm's competitive ratio in Figures 4.6(d), 4.7(d), and 4.8(d). Compared to the carbon-agnostic, constant threshold, and k-min search algorithms, `DTPR-min` achieves a lower average empirical competitive ratio distribution for all tested regions. Across *all regions* at the 95th percentile, `DTPR-min` achieves a worst-case empirical competitive ratio of 1.40. This represents a 48.2% improvement over the *carbon-agnostic* algorithm, and improvements of 15.6% and 14.4% over the k-min search and constant threshold *switching-cost-agnostic* algorithms, respectively.

4.7 Related Work

This chapter contributes directly to three lines of work: (i) work on online search and related problems, e.g., k-search, one-way trading, and online knapsack; (ii) work on online optimization problems with switching costs, e.g., metrical task systems and convex function chasing; and (iii) work on carbon-aware load shifting. We describe the relationship to each below.

Online Search. The OPR problem is related to the online k-search problem [79, 96], as discussed in the introduction and Section 4.2. It also has several similar counterparts, including online conversion problems such as one-way trading [78, 167, 174, 175] and online knapsack problems [168–170], with practical applications to stock trading [79], cloud pricing [164], electric vehicle charging [176], etc. The k-search problem can be viewed as an integral version of the online conversion problem, while the general online conversion problem allows continuous one-way trading. The basic online knapsack problem studies how to pack arriving items of different sizes and values into a knapsack with limited capacity, while its extensions to item departures [164, 169] and multidimensional capacity [170] have also been studied recently. Another line of research leverages ML predictions to design learning-augmented online algorithms for online k-search [96] and online conversion [169]. However, to the best of our knowledge, none of these works consider the switching cost of changing decisions.

Metrical Task Systems. The metrical task systems (MTS) problem was introduced by Borodin et al. in [76]. Several decades of progress on upper and lower bounds on the competitive ratio of MTS recently culminated with a tight bound of $\Theta(\log^2 n)$ for the competitive ratio of MTS on an arbitrary n-point metric space, with $\Theta(\log n)$ being possible on certain metric spaces such as trees [138, 142]. Several modified forms of MTS have also seen significant attention in the literature, such as smoothed online convex optimization (SOCO) and convex function chasing (CFC), in which the decision space is an n-dimensional normed vector space and cost functions are restricted to be convex [38, 77]. The best known upper and lower bounds on the competitive ratio of CFC are $O(n)$ and $\Omega(\sqrt{n})$, respectively, in n-dimensional Euclidean spaces [101, 106]. However, algorithms with competitive ratios independent of dimension can be obtained for certain special classes of functions, such as α-polyhedral functions [102]. Several recent works have also investigated the design of learning-augmented algorithms for various cases of CFC/SOCO and MTS which exploit the performance of ML predictions of the optimal decisions [75, 113, 140, 177, 178] (see, in particular, Chapters 2 and 3 of this thesis). The key characteristic distinguishing OPR from MTS is the presence of a deadline constraint. None of the algorithms for MTS-like problems are designed to handle long-term constraints while being competitive.

Carbon-Aware Temporal Workload Shifting. The goal of shifting workloads in time to allow more sustainable operations of datacenters has been of interest for more than a decade, e.g., [37, 179–181]. Traditionally, such papers have used models that build on one of convex function chasing, k-search, or online knapsack to design algorithms; however such models do not capture both the switching costs and long-term deadlines that are crucial to practical deployment. In recent years, the load shifting literature has focused specifically on reducing the carbon footprint of operations, e.g., [33, 155–157]. Perhaps most related to this chapter is [33], which explores the problem of carbon-aware temporal workload shifting and proposes a threshold-based algorithm that suspends the job when the carbon intensity is higher than a threshold value and resumes it when it drops below the threshold. However, it does not consider switching nor does it provide any deadline guarantees. Other recent work on carbon-aware temporal shifting seeks to address the resultant increase in job completion times. In [182], authors leverage the pause and resume approach to reduce the carbon footprint of ML training and high-performance computing applications such as BLAST [183]. However, instead of resuming at normal speed (1×) during the low carbon intensity periods, their applications resume operation at a faster speed ($m\times$), where the scale factor m depends on the application characteristics. It uses a threshold-based approach to determine the low carbon intensity periods but does not consider switching costs or provide any deadline guarantees. A future direction is to extend the DTPR algorithms to consider the ability to scale up speed after resuming jobs.

In addition to our direct contributions in the above fields, our work is adjacent to several existing studies which have considered switching costs and *hysteretic control* in queueing models for single servers, server farms, and clouds. In [184], an M/M/1 queueing system is presented where the decision maker chooses arrival and service rates at each epoch and incurs a switching cost to change the rates. In this regime, they show that the optimal policy is a hysteretic policy, which exhibits resistance to change due to the switching cost. Gandhi et al. [185] present an M/M/k queueing system for server farms with setup costs, where turning a server on incurs a time delay. Similarly, [186] presents and analyzes a nearly-optimal mechanism to control the performance and power consumption of a server farm, where the setup cost incurs time and energy. A few works have also considered similar problems with different assumptions, such as job arrivals distributed according to a stochastic fluid model [187], and modeling the control policy as a Markov decision process [188]. It is notable that nearly all of these works derive hysteretic control policies based on the queue length, which essentially use a double threshold technique to resist changing decisions as a function of switching cost, a similar flavor of the result as we present in our setting. However, OPR is foundationally different as compared to the above works since we consider a single workload, a single deadline, and costs are exogenous to the online decision; this results in an algorithm design and analysis technique that differ substantially from these queueing models.

4.8 Concluding Remarks

Motivated by carbon-aware load shifting, we introduce and study the online pause and resume problem (OPR), which bridges gaps between several online optimization problems. To our knowledge, it is the first online optimization problem that includes both long-term constraints and switching costs. Our main results provide optimal online algorithms for the minimization and maximization variants of this problem, as well as lower bounds for the competitive ratio of any deterministic online algorithm. Notably, our proposed algorithms match existing optimal results for the related k-search problem when the switching cost is 0, and improve on the k-min search competitive bounds for non-zero switching cost. The key to our results is a novel double threshold algorithm that we expect to be applicable in other online problems with switching costs.

There are a number of interesting directions in which to continue the study of OPR. We have highlighted the application of OPR to carbon-aware load shifting, but OPR also applies to many other problems where pricing changes over time and frequent switching is undesirable. Pursuing these applications is important. Theoretically, there are several interesting open questions. First, considering the target application of carbon-aware load shifting, some workloads are *highly parallelizable* [182], which adds another dimension of scaling to the problem (i.e., instead of choosing to run 1 unit of the job in each time slot, the online player must decide how many units to allocate at each time slot). Furthermore, considering *heterogeneous switching costs* would be a logical extension of the setting we have considered here, modeling, for example, switching models which act as a function of the

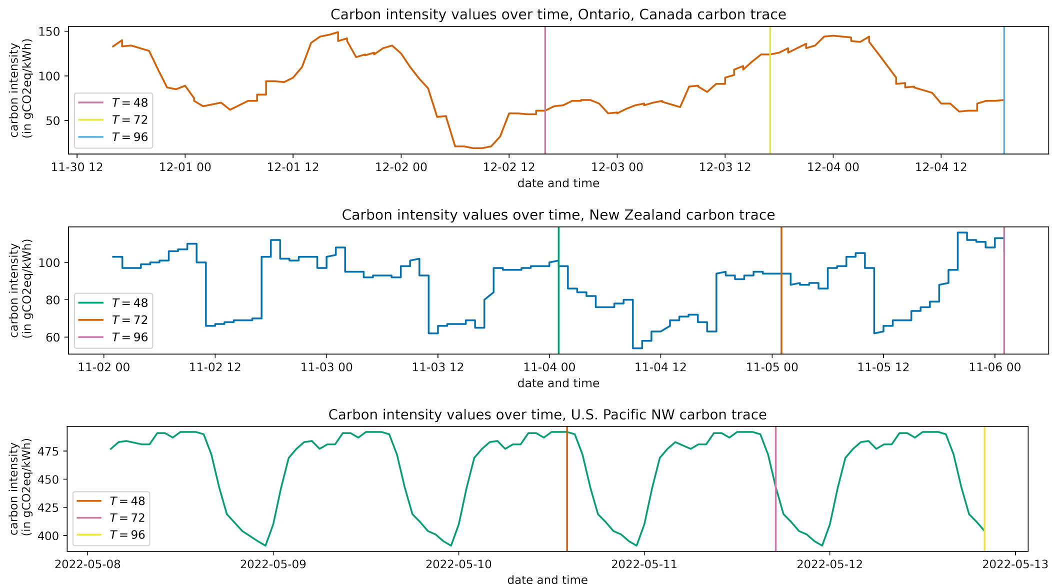


Figure 4.9: Carbon intensity (in gCO_2eq/kWh) values plotted for each region tested in our numerical experiments, with one-hour granularity. We plot a representative random interval of 96 hours, with vertical lines demarcating the different values for T (time horizon) tested in our experiments. In all regions, carbon values roughly follow a diurnal (daily cycle) pattern. Actual values and observed intensities significantly vary in different regions.

time spent in the current state. Both of these make the theoretical problem more challenging, and are important considerations for future work. Additionally, very recent work has incorporated machine-learned advice to achieve better performance on related online problems, including k-search [96, 167], CFC/SOCO [140, 178], and MTS [75, 113, 177]. Designing learning-augmented algorithms for OPR is a very promising line of future work, particularly considering applications such as carbon-aware load shifting, where accurate predictions can significantly improve the algorithm's understanding of the future in the best case, without sacrificing worst-case guarantees. This challenge motivates our work in the next chapter, where we design learning-augmented algorithms for a setting that generalizes OPR to allow for both temporal and *spatial* load shifting.

Appendix

In these appendix sections, we present additional experimental results for the `DTPR-max` algorithm as well as deferred proofs of theoretical results in the main body of the paper.

Algorithm 9: Double Threshold Pause and Resume for OPR-max (`DTPR-max`)

Input: threshold values $\{u_i\}_{i\in[1,k]}$ and $\{\ell_i\}_{i\in[1,k]}$ defined in Equation (4.6), deadline T
Output: online decisions $\{x_t\}_{t\in[1,T]}$

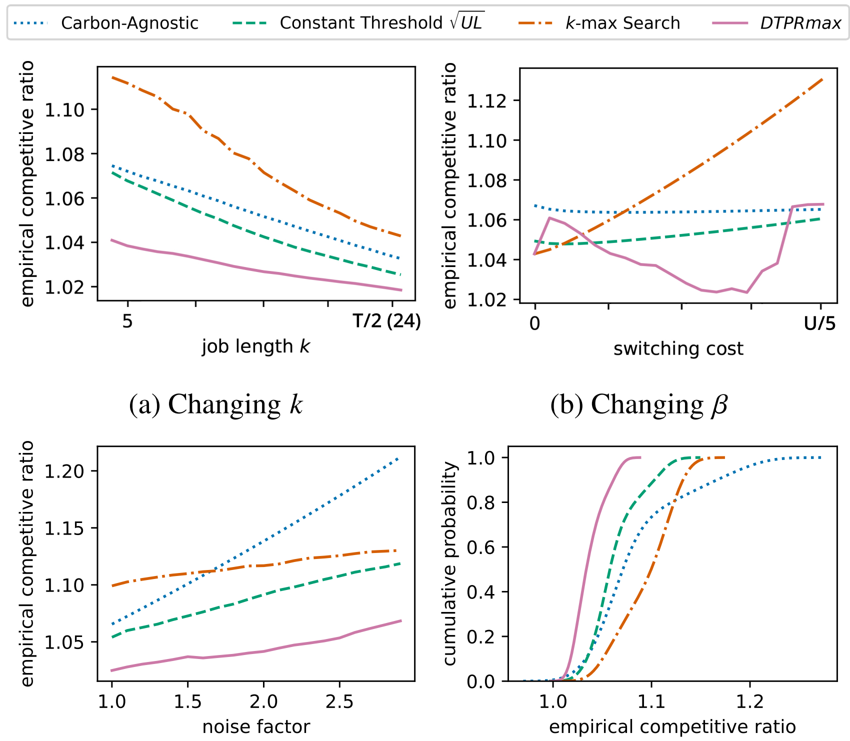
initialize: i = 1;
while price c_t arrives **and** $i \le k$ **do**
    // close to the deadline T, we must accept remaining prices
    if $(k - i) \ge (T - t)$ **then**
        price c_t is accepted, set $x_t = 1$
    // If the previous price was not accepted, use the upper thresholds
    else if $x_{t-1} = 0$ **then**
        if $c_t \ge u_i$ **then**
            price c_t is accepted, set $x_t = 1$
        else
            price c_t is rejected, set $x_t = 0$
    // If the previous price was accepted, use the lower
    else if $x_{t-1} = 1$ **then**
        if $c_t \ge \ell_i$ **then**
            price c_t is accepted, set $x_t = 1$
        end
        else
            price c_t is rejected, set $x_t = 0$
        end
    update $i = i + x_t$
end

4.A Case Study Results for `DTPR-max` Algorithm

This section presents and discusses the deferred experimental results for the `DTPR-max` algorithm (pseudocode summarized in Algorithm 9) in the carbon-aware temporal workload shifting case study. We evaluate `DTPR-max` against the same benchmark algorithms described in Section 4.6.

For the *maximization metric*, we consider the *percentage of carbon-free electricity* powering the grid. At each timestep t, the electricity supply has a carbon-free percentage c_t, i.e., if the job is being processed during time slot t ($x_t = 1$), the electricity powering the datacenter's is c_t% *carbon-free*, and the objective is to maximize this percentage over all k slots of the active running of the workload.

In these maximization experiments, the switching-cost-agnostic k-max-search algorithm chooses to run the ith hour of the job during the first time slot where the carbon-free supply is at least Φ_i. Similarly, the constant threshold algorithm chooses



(a) Changing k

(b) Changing β

(c) Changing volatility

(d) CDF

Figure 4.10: Maximization experiments on Ontario, Canada carbon trace, with $\theta \approx 1.51$ and $T = 48$. (a): Changing job length k w.r.t. time horizon T (x-axis), vs. competitive ratio. (b): Changing switching cost β w.r.t. U (x-axis), vs. competitive ratio. (c): Different volatility levels w.r.t. U (x-axis), vs. competitive ratio. (d): Cumulative distribution function of competitive ratios.

to run the job whenever the carbon-free supply is at least $\sqrt{UL}$. We set L and U to be the minimum and maximum carbon-free supply percentages over the entire trace being studied.

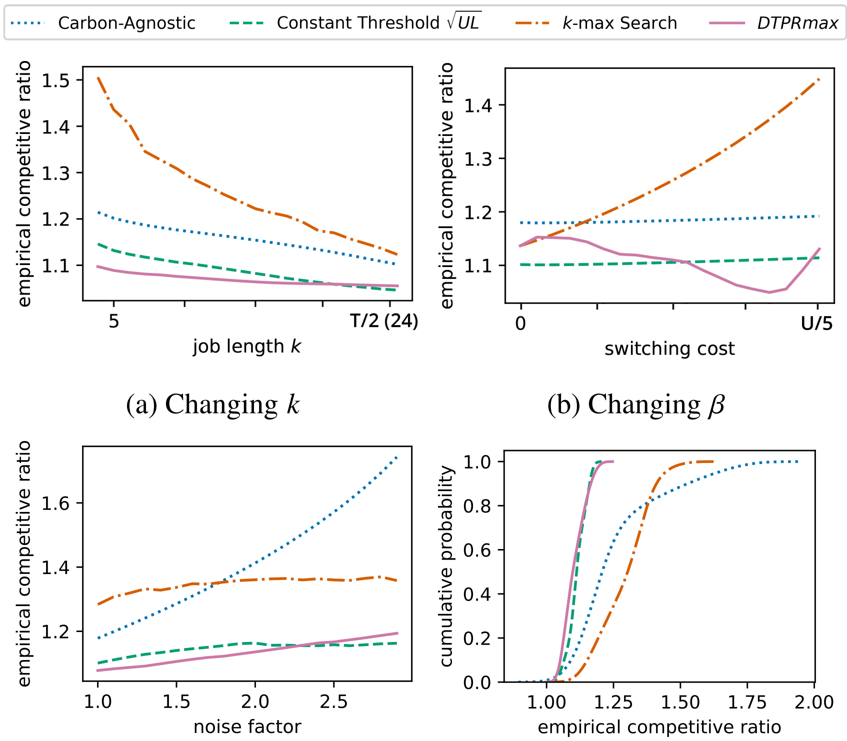
As in Section 4.6, our focus is on the competitive ratio (lower competitive ratio is better). We report the performance of all algorithms for each experiment setting, in each tested region.

In the first experiment, we test all algorithms for different job lengths k in the range from 4 hours to $T/2(24)$. The switching cost β is non-zero and fixed, and no volatility is added to the carbon trace. By testing different values for k, this experiment tests different ratios between the workload length and the slack provided to the algorithm. In Figures 4.10(a), 4.11(a), and 4.12(a), we show that the observed average competitive ratio of `DTPR-max` narrowly outperforms the benchmark algorithms for all values of k in all regions, and it compares particularly favorably for *short* job lengths. Averaging over all regions and job lengths, the competitive ratio achieved by `DTPR-max` is a 4.9% improvement on the carbon-agnostic method, a 8.4% improvement on the k-max search algorithm, and a 2.1% improvement on the constant threshold algorithm.

In the second experiment, we test all algorithms for different switching costs β in the range from 0 to $U/5$. The job length k is set to 10 hours, and no volatility is added to the carbon trace. By testing different values for β, this experiment tests how an increasing switching cost impacts the performance of `DTPR-max` with respect to other algorithms which do not explicitly consider the switching cost. In Figures 4.10(b), 4.11(b), and 4.12(b), we show that the average competitive ratio of `DTPR-max` notably outperforms the other algorithms for a wide range of β values in all regions. Unsurprisingly, the carbon-agnostic technique (which only incurs a switching cost of 2β) is more competitive as β grows. The k-max search algorithm performs noticeably worse as β grows. While the constant threshold algorithm has relatively consistent performance, the k-max search algorithm performs noticeably worse as β grows. Averaging over all regions and switching cost values, the competitive ratio achieved by `DTPR-max` is a 2.5% improvement on the carbon-agnostic method, a 6.4% improvement on the k-max search algorithm, and a 0.1% improvement on the constant threshold algorithm.

In the final experiment, we test all algorithms on sequences with different volatility. The job length k and switching cost β are both fixed. We add volatility by setting a *noise factor* from the range 1.0 to 3.0. By testing different values for this volatility, this experiment tests how each algorithm handles larger fluctuations in the carbon intensity of consecutive timesteps. In Figures 4.10(c), 4.11(c), and 4.12(c), we show that the observed average competitive ratio of `DTPR-max` outperforms the other algorithms for most noise factors in all regions, with a slight degradation in the Pacific Northwest region. Intuitively, higher volatility values cause the online algorithms to perform worse in general. Averaging over all regions and noise factors, the competitive ratio achieved by `DTPR-max` is a 13.0% improvement on the carbon-agnostic method, a 11.2% improvement on the k-max search algorithm, and a 2.1% improvement on the constant threshold algorithm.

By averaging over all experiments for a given region, we obtain the cumulative distribution function plot for each algorithm's competitive ratio in Figures 4.10(d), 4.11(d), and 4.12(d). Compared to the carbon-agnostic, constant threshold, and k-max search algorithms, `DTPR-max` generally exhibits a lower average empirical



(a) Changing k (b) Changing β

(c) Changing volatility (d) CDF

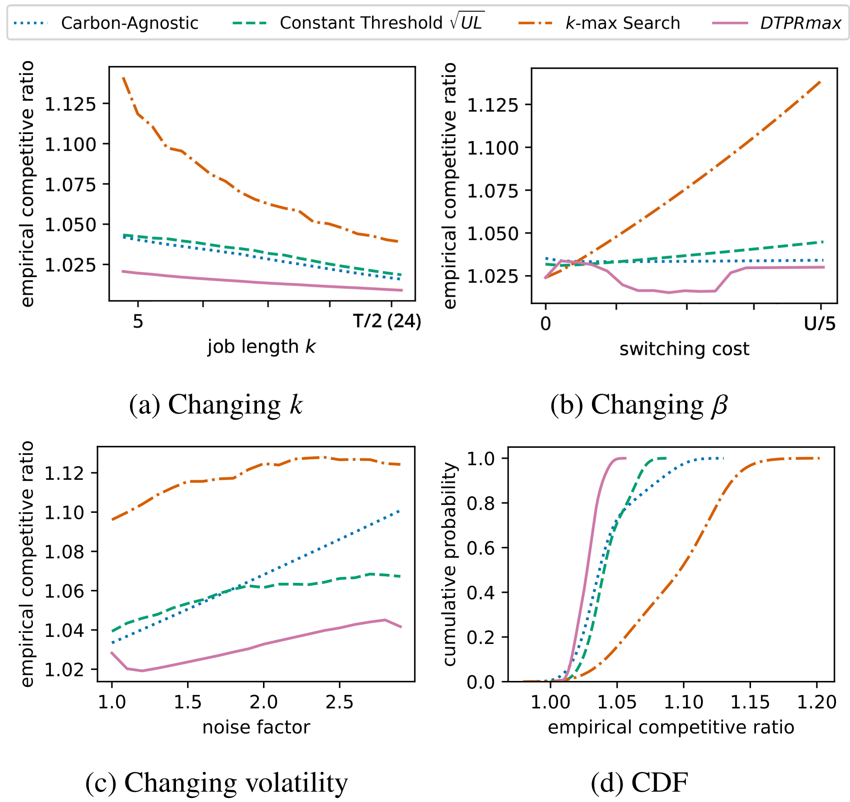
Figure 4.11: Maximization experiments on U.S. Pacific Northwest carbon trace, with $\theta \approx 5.24$ and $T = 48$. (a): Changing job length k w.r.t. time horizon T (x-axis), vs. competitive ratio. (b): Changing switching cost β w.r.t. U (x-axis), vs. competitive ratio. (c): Different volatility levels w.r.t. U (x-axis), vs. competitive ratio. (d): Cumulative distribution function of competitive ratios.

competitive ratio over the tested regions. Notably, all of the algorithms are nearly 1-competitive in our experiments. Compared to our minimization experiments, `DTPR-max` outperforms the baseline algorithms by a smaller margin. Across *all regions* at the 95th percentile, `DTPR-max` achieves a worst-case empirical competitive ratio of 1.08. This represents a 16.1% improvement over the *carbon-agnostic* algorithm, and improvements of 11.4% and 2.19% over the k-max search and constant threshold *switching-cost-agnostic* algorithms, respectively.

We conjecture that one dynamic contributing to this is the relatively low values of θ observed for the carbon-free supply percentage in these real-world carbon traces.

4.B Competitive Analysis of `DTPR-max`: Proof of Theorem 4.4.2

Here we prove the `DTPR-max` results presented in Theorem 4.4.2 and Corollary 4.4.4.



(a) Changing k

(b) Changing β

(c) Changing volatility

(d) CDF

Figure 4.12: Maximization experiments on New Zealand carbon trace, with $\theta \approx 1.35$ and $T = 48$. (a): Changing job length k w.r.t. time horizon T (x-axis), vs. competitive ratio. (b): Changing switching cost β w.r.t. U (x-axis), vs. competitive ratio. (c): Different volatility levels w.r.t. U (x-axis), vs. competitive ratio. (d): Cumulative distribution function of competitive ratios.

Proof of Theorem 4.4.2. For $0 \leq j \leq k$, let $\mathcal{S}_j \subseteq \mathcal{S}$ be the sets of OPR-max price sequences for which `DTPR-max` accepts exactly j prices (excluding the $k - j$ prices it is forced to accept at the end of the sequence). Then all of the possible price sequences for OPR-max are represented by $\mathcal{S} = \bigcup_{j=0}^{k} \mathcal{S}_j$. By definition, $u_{k+1} = U$. Let $\epsilon > 0$ be fixed, and define the following two price sequences σ_j and ρ_j:

$$\forall 0 \leq j \leq k : \sigma_j = u_1, \ell_2, \dots, \ell_j, L, \underbrace{u_{j+1} - \epsilon, \dots, u_{j+1} - \epsilon}_{k}, \underbrace{L, L, \dots, L}_{k}.$$

$$\forall 0 \leq j \leq k : \rho_j = u_1, L, u_2, L, \dots, L, u_j, L, \underbrace{u_{j+1} - \epsilon, \dots, u_{j+1} - \epsilon}_{k}, \underbrace{L, L, \dots, L}_{k}.$$

We have two special cases for $j = 0$ and $j = 1$. For $j = 0$, we have that $\sigma_0 = \rho_0$, and this sequence simply consists of $u_1 - \epsilon$ repeated k times, followed by L repeated k times. For $j = 1$, we also have that $\sigma_1 = \rho_1$, and this sequence consists of one price with value u_1 and one price with value L, followed by $u_2 - \epsilon$ repeated k times and L repeated k times.

Observe that as $\epsilon \to 0$, σ_j and ρ_j are sequences yielding the worst-case ratios in $\mathcal{S}_j$, as DTPR-max is forced to accept $(k-j)$ worst-case L values at the end of the sequence, and each accepted value is exactly equal to the corresponding threshold.

σ_j and ρ_j also represent two extreme possibilities for the switching cost. In σ_j, DTPR-max only switches twice, but it mostly accepts values ℓ_i. In ρ_j, DTPR-max must switch $j+1$ times because there are many intermediate L values, but it only accepts values which are at least u_i.

Observe that $\texttt{OPT}(\sigma_j)/\texttt{DTPR-max}(\sigma_j) = \texttt{OPT}(\rho_j)/\texttt{DTPR-max}(\rho_j)$. First, the optimal solution for both sequences is exactly the same: $kc_{\max}(\sigma_j) - 2\beta = kc_{\max}(\rho_j) - 2\beta$.
For any sequence s in $\mathcal{S}_j$, we also know that $c_{\max}(s) < u_{j+1}$, so $\texttt{OPT}(\rho_j) = \texttt{OPT}(\sigma_j) \leq ku_{j+1} - 2\beta$.

By definition of the threshold families $\{u_i\}_{i\in[1,k]}$ and $\{\ell_i\}_{i\in[1,k]}$, we know that $\sum_{i=1}^{j} u_i - j2\beta = \sum_{i=1}^{j} \ell_i$ for any value $j \geq 2$:

$$\begin{aligned}
\texttt{DTPR-max}(\rho_j) &= \left(u_1 + \sum_{i=2}^{j} \ell_i + (k-j)L - 4\beta\right) \\
&= \left(\sum_{i=1}^{j} u_i + (k-j)L - (j+1)2\beta\right) \\
&= \texttt{DTPR-max}(\sigma_j).
\end{aligned}$$

Note that whenever $j < 2$, we have that $\sigma_0 = \rho_0$, and $\sigma_1 = \rho_1$. Thus, $\texttt{DTPR-min}(\rho_j) = \texttt{DTPR-min}(\sigma_j)$ holds for any value of j.

By definition of u_1, we simplify $u_1 + \sum_{i=2}^{j} \ell_i + (k-j)L - 4\beta$ to $\sum_{i=1}^{j} \ell_i + (k-j)L - 2\beta$. For any sequence $s \in \mathcal{S}_j$, we have the following:

$$\begin{aligned}
\frac{\texttt{OPT}(s)}{\texttt{DTPR-max}(s)} &\leq \frac{\texttt{OPT}(\sigma_j)}{\texttt{DTPR-max}(\sigma_j)} \\
&= \frac{\texttt{OPT}(\rho_j)}{\texttt{DTPR-max}(\rho_j)} \\
&\leq \frac{ku_{j+1} - 2\beta}{\sum_{i=1}^{j} \ell_i + (k-j)L - 2\beta}. \qquad (4.12)
\end{aligned}$$

Lemma 4.B.1. *For any $j \in [0, k]$, by definition of $\{u_i\}_{i\in[1,k]}$ and $\{\ell_i\}_{i\in[1,k]}$,*

$$\omega \cdot \left(\sum_{i=1}^{j} \ell_i + (k-j)L - 2\beta \right) \le k u_{j+1} - 2\beta.$$

The proof is deferred to Section 4.C.

For $\epsilon \to 0$, the competitive ratio $\texttt{OPT}/\texttt{DTPR-max}$ is exactly ω:

$$\forall 0 \le j \le k: \quad \frac{\texttt{OPT}(\sigma_j)}{\texttt{DTPR-max}(\sigma_j)} = \frac{k u_{j+1} - 2\beta}{\sum_{i=1}^{j} \ell_i + (k-j)L - 2\beta} = \omega.$$

and thus for any sequence $s \in \mathcal{S}$,

$$\forall s \in \mathcal{S}: \quad \frac{k c_{\max}(s) - 2\beta}{\texttt{DTPR-max}(s)} \le \omega.$$

Since $\texttt{OPT}(s) \le k c_{\max}(s) - 2\beta$ for any sequence s, this implies that $\texttt{DTPR-max}$ is ω-competitive. □

Proof of Corollary 4.4.4. For simplification purposes, let $\beta = bL/2$, where b is a real constant on the interval $(0, k)$. To show part **(a)** for $\texttt{REGIME-1}$, with fixed $k \ge 1$, observe that for sufficiently large ω, we have the following:

$$\begin{aligned} \theta - b - 1 &= (\omega - 1)\left(1 + \frac{\omega}{k}\right)^k - \left(b - \frac{b}{k} + \frac{b\omega}{k}\right)\left(1 + \frac{\omega}{k}\right)^k \\ &\approx (1 + o(1)) \left[\omega \left(\frac{\omega}{k}\right)^k - b\left(\frac{\omega}{k}\right)^{k+1} - b \right]. \end{aligned}$$

Let $\omega_+ = \sqrt[k+1]{k^k \cdot \frac{k\theta}{k-b}}$. Then, for sufficiently large ω, we have the following:

$$\begin{aligned} (1 + o(1)) \left[\omega_+ \left(\frac{\omega_+}{k}\right)^k - b\left(\frac{\omega_+}{k}\right)^{k+1} - b \right] &= (1 + o(1)) \frac{(k-b)(\theta)}{k-b} \\ &= (1 + o(1)) \left[\theta - b \right]. \end{aligned}$$

Furthermore, let $\varepsilon > 0$ and set $\omega_- = (1 - \varepsilon) \sqrt[k+1]{k^k \cdot \frac{k\theta}{k-b}}$. A similar calculation as above shows that for sufficiently large θ we have:

$$(\omega_- - 1)\left(1 + \frac{\omega_-}{k}\right)^k - \left(b - \frac{b}{k} + \frac{a\omega_-}{k}\right)\left(1 + \frac{\omega_-}{k}\right)^k \ge (1 - 3k\varepsilon) \left[\theta - b \right].$$

Thus, $\omega = O\left(\sqrt[k+1]{k^k \frac{k\theta}{k-b}} \right)$ satisfies (4.10) for sufficiently large ω, fixed $k \ge 1$, and

$\beta = \frac{bL}{2}$ s.t. $b \in (1, k)$.

To show part **(b)** for `REGIME-2`, observe that the right-hand side of (4.10) can be approximated as $\left(1 + \frac{\omega}{k}\right)^k \approx e^{\omega}$ when $k \to \infty$. Then by taking limits on both sides, we obtain the following:

$$\frac{U - L - 2\beta}{L(\omega - 1) - 2\beta(1)} = e^{\omega}.$$

Let $\beta = bL/2$ as outlined above. We then obtain the following:

$$\frac{U - L - bL}{L(\omega - 1) - bL} = \frac{\theta - 1 - b}{\omega - 1 - b} = e^{\omega} \implies \theta - 1 - b = (\omega - 1 - b)\, e^{\omega}.$$

By definition of the Lambert W function, solving this equation for ω obtains part (2). □

4.C Proofs of Lemmas 4.5.1 and 4.B.1

In this section, we give the deferred proofs of Lemmas 4.5.1 and 4.B.1, which are used in the proofs of Theorem 4.4.1 and Theorem 4.4.2, respectively.

Proof of Lemma 4.5.1. We show that the following holds for any $j \in [0, k]$, by Definition 4.3.1:

$$\sum_{i=1}^{j} u_i + (k - j)U + 2\beta \le \alpha \cdot (k\ell_{j+1} + 2\beta).$$

First, note that $k\ell_{j+1} = k(u_{j+1} - 2\beta)$ for all $j \in [0, k]$, by Observation 4.3.3. This gives us the following:

$$\sum_{i=1}^{j} u_i + (k - j)U + 2\beta \le \alpha k u_{j+1} + \alpha 2\beta - \alpha k 2\beta,$$

$$\sum_{i=1}^{j} u_i + (k - j)U + [2\beta - \alpha 2\beta + \alpha k 2\beta] \le \alpha k u_{j+1},$$

$$\frac{(k - j)U}{\alpha k} + \frac{\sum_{i=1}^{j} u_i}{\alpha k} + \left[\frac{2\beta}{\alpha k} - \frac{2\beta}{k} + 2\beta\right] \le u_{j+1}.$$

By substituting Def. 4.3.1 into $\sum_{i=1}^{j} u_i$, the above can be simplified exactly to the closed form for u_{j+1}:

$$\frac{U}{\alpha} - \frac{jU}{\alpha k} + \left(\frac{\sum_{i=1}^{j} u_i}{\alpha k}\right) + \left[\frac{2\beta}{\alpha k} - \frac{2\beta}{k} + 2\beta\right] = u_{j+1},$$

$$\left[U - \left(U - \frac{1}{\alpha}\right)\left(1 + \frac{1}{\alpha k}\right)^j\right] + \left[\left(\frac{2\beta}{\alpha k} - \frac{2\beta}{k} + 2\beta\right)\left(1 + \frac{1}{\alpha k}\right)^j\right] = u_{j+1}$$

and the claim follows by the definition of u_{j+1}. □

Proof of Lemma 4.B.1. We show that the following holds for any $j \in [0, k]$, by Definition 4.3.2:

$$\omega \cdot \left(\sum_{i=1}^{j} \ell_i + (k-j)L - 2\beta \right) \leq k u_{j+1} - 2\beta.$$

First, note that $k u_{j+1} = k(\ell_{j+1} + 2\beta)$ for all $j \in [0, k]$, by Observation 4.3.3. This gives us the following:

$$\sum_{i=1}^{j} \ell_i + (k-j)L - 2\beta \leq \frac{k\ell_{j+1}}{\omega} - \frac{2\beta}{\omega} + \frac{k2\beta}{\omega},$$

$$\sum_{i=1}^{j} \ell_i + (k-j)L - \left[2\beta - \frac{2\beta}{\omega} + \frac{k2\beta}{\omega} \right] \leq \frac{k\ell_{j+1}}{\omega},$$

$$\frac{\omega \left(\sum_{i=1}^{j} \ell_i \right)}{k} + \frac{\omega(k-j)L}{k} - \left[\frac{\omega 2\beta}{k} - \frac{2\beta}{k} + 2\beta \right] \leq \ell_{j+1}.$$

By substituting Def. 4.3.2 into $\sum_{i=1}^{j} \ell_i$, the above can be simplified exactly to the closed form for ℓ_{j+1}:

$$\omega L - \frac{\omega j L}{k} + \frac{\omega \left(\sum_{i=1}^{j} \ell_i \right)}{k} - \left[\frac{\omega 2\beta}{k} - \frac{2\beta}{k} + 2\beta \right] = \ell_{j+1},$$

$$\left[L + (\omega L - L) \left(1 + \frac{\omega}{k} \right)^{j} \right] - \left[\left(\frac{\omega 2\beta}{k} - \frac{2\beta}{k} + 2\beta \right) \left(1 + \frac{\omega}{k} \right)^{j} \right] = \ell_{j+1}$$

and the claim follows by the definition of ℓ_{j+1}. □

4.D Proofs of Lower Bound Results

This section formally proves the lower bound results for OPR-max, building on the proof for OPR-min provided in Section 4.5.

Proof of Theorem 4.4.6 (OPR-max Lower Bound)

Proof of Theorem 4.4.6. Let ALG be a deterministic online algorithm for OPR-max, and suppose that the adversary uses the price sequence $u_1, \ldots, u_k$, which is exactly the sequence defined by (4.6). u_1 is presented to ALG, at most k times or until ALG accepts it. If ALG never accepts u_1, the remainder of the sequence is all L, and ALG achieves a competitive ratio of $\frac{ku_1 - 2\beta}{kL - 2\beta} = \omega$, as defined in (4.8).

If ALG accepts u_1, the next price presented is L, repeated at most k times *or until ALG switches to reject L*. After ALG has switched, u_2 is presented to ALG, at most k

times or until ALG accepts it. Again, if ALG never accepts u_2, the remainder of the sequence is all L, and ALG achieves a competitive ratio of at least $\frac{ku_2-2\beta}{u_1+(k-1)L-4\beta} = \omega$, as defined in (4.8).

As the sequence continues, whenever ALG does not accept some u_i after it is presented k times, the adversary drops the price to L for the remainder of the sequence. Otherwise, if ALG accepts k prices before the end of the sequence, the adversary concludes by presenting U at least k times.

Observe that any ALG which does not immediately reject the first L presented to it after accepting some u_i obtains a competitive ratio strictly worse than ω. To illustrate this, suppose ALG has just accepted u_1, achieving a profit of $u_1 - \beta$ so far. The adversary begins to present L prices, and ALG accepts $y \leq (k-1)$ of these L prices before switching away. If $y = (k-1)$, ALG will accept k prices before the end of the sequence and achieve a competitive ratio of $\frac{kU-2\beta}{u_1+(k-1)L-2\beta} > \omega$. Otherwise, if $y < (k-1)$, the profit achieved by ALG so far is at most $u_1 - 2\beta + yL$, while the profit achieved by ALG if it had immediately switched away ($y = 0$) would be $u_1 - 2\beta$—since any price which might be accepted by ALG in the future should be $\geq L$, the latter case strictly improves the competitive ratio of ALG.

Assuming that ALG does immediately reject any L presented to it, and that ALG accepts some prices before the end of the sequence, the competitive ratio attained by ALG is at least $\frac{ku_{j+1}-2\beta}{\sum_{i=1}^{j} u_i-(j+1)2\beta+(k-j)L} = \omega$, as defined in (4.8).

Similarly, if ALG accepts k prices before the end of the sequence, the competitive ratio attained by ALG is at least $\frac{kU-2\beta}{\sum_{i=1}^{k} u_i-k2\beta} = \omega$, as defined in (4.8).

Since any arbitrary deterministic online algorithm ALG cannot achieve a competitive ratio better than ω playing against this adaptive adversary, our proposed algorithm DTPR-max is optimal. □

Chapter 5

LEARNING-AUGMENTED COMPETITIVE ALGORITHMS FOR SPATIOTEMPORAL ONLINE ALLOCATION WITH DEADLINE CONSTRAINTS

We now return to the general task of designing learning-augmented algorithms in a setting that unites the general metric structure of Chapter 3 and the long-term deadline constraint of Chapter 4. In particular, we introduce and study spatiotemporal online allocation with deadline constraints (SOAD), a new online problem motivated by emerging challenges in energy and sustainability. In SOAD, an online player completes a workload by allocating and scheduling it on the points of a metric space (X, d) while subject to a deadline T. At each timestep, a service cost function is revealed that represents the cost of servicing the workload at each point, and the player must irrevocably decide the current allocation of work to points. Whenever the player moves this allocation, they incur a movement cost defined by the distance metric $d(\cdot, \ \cdot)$ that captures, e.g., an overhead cost. SOAD formalizes the open problem of combining general metrics and deadline constraints in the online algorithms literature, unifying problems such as metrical task systems and online search. We propose a competitive algorithm for SOAD along with a matching lower bound establishing its optimality. Our main algorithm, ST-CLIP, is a learning-augmented algorithm that takes advantage of predictions (e.g., forecasts of relevant costs) and achieves an optimal consistency-robustness tradeoff. We evaluate our proposed algorithms in a simulated case study of carbon-aware spatiotemporal workload management, an application in sustainable computing that schedules a delay-tolerant batch compute job on a distributed network of datacenters. In these experiments, we show that ST-CLIP substantially improves on heuristic baseline methods.

This chapter is primarily based on the following paper:

[1] A. Lechowicz, N. Christianson, B. Sun, N. Bashir, M. Hajiesmaili, A. Wierman, and P. Shenoy, "Learning-Augmented Competitive Algorithms for Spatiotemporal Online Allocation with Deadline Constraints," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 9, no. 1, 8:1–8:49, Mar. 2025. DOI: 10.1145/3711701. [Online]. Available: https://dl.acm.org/doi/10.1145/3711701.

5.1 Introduction

We introduce and study spatiotemporal online allocation with deadline constraints (SOAD), an online optimization problem motivated by emerging challenges in sustainability. In SOAD, an online player aims to service a workload by allocating and scheduling it on one of n points represented by a metric space (X, d). They pay a service cost at a point if the workload is currently being serviced there, a spatial movement cost defined by the metric whenever they change the allocation between points, and a temporal switching cost when bringing the workload into or out of service at a single point. The workload arrives with a *deadline constraint* T that gives the player some *slack*, i.e., the workload can be *paused* for some time to avoid high cost periods without violating the constraint.

SOAD builds on a long history of related problems in online algorithms. In particular, two lines of work share specific features in common with our setting. One line of work focuses on metrical task systems (MTS) and smoothed online convex optimization (SOCO), where problems consider online optimization with movement costs over general metrics, but do not accommodate long-term constraints, such as deadlines [76, 101, 102, 138, 142, 146, 189]. A complementary line of work is that of online search problems with long-term constraints, such as one-way trading (OWT) and online knapsack—these problems enforce that a player's cumulative decisions satisfy a constraint over the entire input, but do not consider general metric (decision) spaces or movement costs [78, 79, 174, 176]. SOAD extends both MTS/SOCO-type problems and OWT-type problems by simultaneously considering general metric movement/switching costs and deadline (i.e., long-term) constraints.

For many applications, the underlying problem to be solved often requires a model with *both* smoothed optimization (i.e., movement costs) *and* deadline constraints. Furthermore, for an application such as carbon-aware workload management in datacenters, where the spatial movement cost corresponds to, e.g., network delays (see Section 5.2), it is necessary to consider a general metric space, since pairwise network latencies do not necessarily correspond to simple distances such as Euclidean (geographic) distances. The question of whether it is possible to design competitive online algorithms in this combined setting has remained open for over a decade, with theoretical progress emerging only in the last few years in special cases such as the unidimensional setting, in ℓ_1 vector spaces, or with different performance metrics such as regret [190–195] (see, in particular, Chapter 4 of this thesis). This work seeks to close this gap by answering the following question:

Is it possible to design online algorithms for SOAD *that manage the challenges of **general metrics** and provide competitive guarantees without violating the **deadline constraint**?*

It is well known that problems related to SOAD, such as MTS and OWT, are difficult in the sense that their competitive ratios scale in the size of the decision space or the ratio between maximal and minimal prices. However, these pessimistic lower bounds hold in the worst case, while in practice a decision-maker can often leverage data-driven machine learning approaches to obtain algorithms that perform better empirically. Recent work in the online algorithms literature has leveraged the paradigm of learning-augmented algorithms [69, 70] to design and analyze algorithms that can take advantage of patterns in the input via untrusted "advice" (e.g., predictions from a machine learning model) without losing adversarial competitive bounds. Such learning-augmented algorithms have been designed for precursor problems to SOAD, including MTS and OWT [75, 140, 167, 177] (see, in particular, Chapters 2 and 3 of this thesis). In the SOAD setting, supported by the availability of practical predictions for our motivating applications and a lack of learning-augmented algorithmic strategies that accommodate both general metrics and deadline constraints, we additionally consider the question:

Can we design algorithms for SOAD *that integrate **untrusted advice** (such as machine-learned predictions) to further improve performance without losing worst-case guarantees?*

Related work

Our results address a long-standing open problem of combining online optimization with general switching costs (MTS/SOCO) and deadline constraints. Although general MTS is a famously well-studied problem in online algorithms [76, 116, 143, 189, 196, 197], it has not been studied under the general form of long-term intertemporal constraints that we consider in the SOAD formulation. Amongst the two lines of related work, prior work has focused on either designing MTS-style algorithms using techniques such as mirror descent [142], primal-dual optimization [197], and work functions [198], or OWT-style algorithms using techniques such as threshold-based algorithm design [79, 168], pseudo-reward maximization [170, 176], and protection-level policies [78]. As these distinct techniques have been tailored to their respective problem settings, there has been almost no cross-pollination between MTS-type and OWT-type problems until this work. The combination of smoothed optimization and long-term constraints has drawn recent attention in the paradigm of regret analysis in problems such as bandits with knapsacks and OCO with long-term constraints [190, 191]. However, despite established connections between MTS and online learning [116], the problem of optimal *competitive* algorithm design in general settings of this form has yet to be explored.

A select few works [192–195] consider both *switching costs* and deadline constraints in a competitive regime, although they are restricted to special cases such as unidimensional decisions or ℓ_1 metrics. Due to these assumptions, their results and algorithms fail to capture the general problem that we consider. As just one example, all of these works assume that the switching cost is only temporal, in the sense that the online player pays the same cost whether they are switching into or out of a state that makes progress towards the deadline constraint. This assumption is overly restrictive because it cannot accommodate switching cost situations that may arise in motivating applications, e.g., the case where the player chooses to move between points of the metric while simultaneously switching into an ON state.

Our work also contributes to the field of learning-augmented algorithms, designed to bridge the performance of untrusted advice and worst-case competitive guarantees [69, 70]. Learning-augmented design has been studied in many online problems including ski rental [110], bipartite matching [199], and several related problems including MTS/SOCO and OWT [75, 96, 140, 167, 193, 195]. For MTS/SOCO, a dominant algorithmic paradigm is to adaptively combine the actions of a robust decision-maker and those of, e.g., a machine-learning model [75, 140]; optimal tradeoffs between robustness and consistency have also been shown in the case of general metrics [177] (see Chapter 3). For OWT and k-search, several works have given threshold-based algorithms incorporating predictions that are likewise shown to achieve an optimal robustness-consistency tradeoff [96, 167]. The advice models for these two tracks of literature are quite different and lead to substantially different algorithms—namely, online search problems typically assume that the algorithm receives a prediction of, e.g., the *best price*, while MTS/SOCO typically consider *black-box advice* predicting the optimal decision at each timestep. Amongst the limited prior literature that considers learning-augmentation in problems with switching costs and deadline constraints [193–195], both advice models have been considered, underscoring the challenge of the SOAD setting, where the optimal choice of advice model (and corresponding design techniques) is not obvious *a priori*.

Contributions

Our main technical contributions make progress on a longstanding problem in online optimization that models emerging practical problems in areas such as sustainability. Our algorithms and lower bounds for SOAD are the first results to consider competitive analysis for deadline-constrained problems on general metrics. We obtain positive results for both of the questions posed above under assumptions informed by practice. In particular, we provide the first competitive algorithm, PCM (**P**seudo-**C**ost **M**inimization, see Algorithm 10), for this type of problem in Section 5.3, and show that it achieves the best possible competitive ratio up to log factors that result from the generality of the metric. Surprisingly, the competitive upper bound we prove for SOAD (Theorem 5.3.3) compares favorably against known strong lower bounds for precursor problems such as MTS and OWT; which suggests an insight that additional structure imposed by constraints can actually facilitate competitive decision making, despite the added complexity of the general metric and deadline constraint. To achieve this result, we develop theoretical tools from both lines of related work that help us tackle challenging components of SOAD. For instance, we leverage randomized metric embeddings and optimal transport to endow a general metric (X, d) with a structure that facilitates analysis. From the online search literature, we leverage techniques that balance the tradeoff between cost and constraint satisfaction, specifically generalizing these ideas to operate in the more complex metric setting necessitated by SOAD.

In Section 5.4, we introduce our main learning-augmented algorithm, ST-CLIP, which integrates black-box decision advice based on, e.g., machine-learned predictions to significantly improve performance without losing worst-case competitive bounds. In our approach, we first prove an impossibility result on the robustness-consistency tradeoff for any algorithm. Using an adaptive optimization-based framework first proposed by [195], we design an algorithm that combines the theoretical tools underpinning PCM in concert with a constraint hedging against worst-case service costs and movement costs that threaten the desired consistency bound. This ensures ST-CLIP attains the optimal tradeoff (up to log factors) in general metric spaces.

In Section 5.5, motivated by real-world applications where the movement cost between points of the metric may not be constant, we present a generalization of SOAD where distances are allowed to be *time-varying* and show that our algorithms extend to this case. Finally, in Section 5.6, we evaluate our algorithms in a case study of carbon-aware spatiotemporal workload management (see Section 5.2) on a simulated global network of datacenters. We show that ST-CLIP is able to leverage *imperfect* advice and significantly improve on heuristic baselines for the problem.

5.2 Problem Formulation, Motivating Applications, Challenges, and Preliminaries

In this section, we introduce the spatiotemporal online allocation with deadline constraints (SOAD) problem and provide motivating applications as examples. We also discuss some intrinsic challenges in SOAD that prevent the direct application of existing techniques, and introduce relevant preliminaries from related work.

Spatiotemporal online allocation with deadline constraints (SOAD)

Problem statement. Consider a decision-maker that manages n points defined on a metric space (X, d), where X denotes the set of points and $d(u, v)$ denotes the distance between any two points $u, v \in X$. In a time-slotted system, the player aims to complete a unit-size workload before a deadline T while minimizing the total service cost by allocating the workload across points and time.

Allocation definition. The decision-maker specifies a spatial allocation to one of the points $u \in X$. At the chosen point, they also make a temporal allocation that *fractionally* divides the allocation between two states, $\texttt{ON}^{(u)}$ and $\texttt{OFF}^{(u)}$, where the allocation to $\texttt{ON}^{(u)}$ represents the amount of resources actively servicing the workload. Let $\mathbf{x}_t := \{x_t^{\texttt{ON}^{(u)}}, x_t^{\texttt{OFF}^{(u)}}\}_{u \in X}$ denote the allocation decision at time t across all points and states, where $x_t^{\texttt{ON}^{(u)}}$ and $x_t^{\texttt{OFF}^{(u)}}$ denote the allocation to the ON and OFF states at point u, respectively. The feasible set for this vector allocation is given by $\mathcal{X} := \{\mathbf{x} \subseteq [0,1]^{2n} : x^{\texttt{ON}^{(u)}} + x^{\texttt{OFF}^{(u)}} \in \{0,1\}, \forall u \in X, \|\mathbf{x}\|_1 = 1\}$.

Deadline constraint. Let $c(\mathbf{x}_t) : \mathcal{X} \to [0,1]$ denote a constraint function that is known to the decision-maker and models the fraction of the workload completed by an allocation $\mathbf{x}_t$. Specifically, we let $c(\mathbf{x}_t) = \sum_{u \in X} c^{(u)} \cdot x_t^{\texttt{ON}^{(u)}}$, where $c^{(u)}$ is a positive *throughput constant* that encodes how much of the workload is completed during one time slot with a full allocation to the state $\texttt{ON}^{(u)}$. Across the entire time horizon, the decision-maker is subject to a *deadline constraint* stipulating that the cumulative allocations must satisfy $\sum_{t=1}^{T} c(\mathbf{x}_t) \geq 1$. This encodes the requirement that sufficient allocations must be assigned to the ON states to finish a unit-size workload by the deadline T. The assumption that the workload is of unit size is made without loss of generality, as $c(\cdot)$ can be scaled appropriately—e.g., if a workload doubles in size, $c(\cdot)$ can be scaled by a factor of $1/2$ to reflect this.

Service and switching costs. At each time t, the cost of allocation $\mathbf{x}_t$ consists of a *service cost* $f_t(\mathbf{x}_t) = \sum_{u \in X} f_t^{(u)} \cdot x_t^{\mathtt{ON}^{(u)}}$ for allocations to ON states, where $f_t^{(u)}$ represents the service cost of point u at time t; and a *switching cost* $g(\mathbf{x}_t, \mathbf{x}_{t-1})$ that includes a spatial movement cost of moving the allocation between points and a temporal switching cost incurred between ON and OFF states within one point. Specifically, whenever the decision-maker changes the allocation across points, they pay a movement cost $d(u_{t-1}, u_t)$, where $u_{t'} = \{u \in X : x_{t'}^{\mathtt{ON}^{(u)}} + x_{t'}^{\mathtt{OFF}^{(u)}} = 1\}$ is the location of the allocation at time t' (in Section 5.5, we give a generalization where $d(\cdot, \cdot)$ also varies with time). Within each point, the decision-maker pays a switching cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\boldsymbol{\beta})} = \sum_{u \in X} \beta^{(u)} |x_t^{\mathtt{ON}^{(u)}} - x_{t-1}^{\mathtt{ON}^{(u)}}|$, where $\beta^{(u)}$ is the switching overhead factor when changing the ON / OFF allocation at point u. The overall switching cost is $g(\mathbf{x}_t, \mathbf{x}_{t-1}) \coloneqq d(u_{t-1}, u_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\boldsymbol{\beta})}$, and g is known in advance. The decision-maker starts (at $t = 0$) with a full allocation at some OFF state, and they *must* end (at $t = T+1$) with their full allocation at any OFF state.

Spatiotemporal allocation with deadline constraints. The objective of the player is to minimize the total cost while satisfying the workload's deadline constraint. Let $\mathcal{I} \coloneqq \{f_t(\cdot)\}_{t \in [T]}$ denote an input sequence of SOAD. For a given $\mathcal{I}$, the offline version of the problem can be formulated as:

$$
[\mathsf{SOAD}] \qquad \begin{aligned} \min_{\{\mathbf{x}_t\}_{t \in [T]}} \quad & \underbrace{\sum_{t=1}^{T} f_t(\mathbf{x}_t)}_{\text{Service cost}} + \underbrace{\sum_{t=1}^{T+1} g\left(\mathbf{x}_t, \mathbf{x}_{t-1}\right)}_{\text{Switching cost (e.g., overhead)}} \\ \text{s.t.} \quad & \underbrace{\sum_{t=1}^{T} c(\mathbf{x}_t) \geq 1,}_{\text{Deadline constraint}} \quad \mathbf{x}_t \in \mathcal{X}. \end{aligned} \tag{5.1}
$$

In the offline setting, we note that a randomized version of the above formulation, without binary constraints, is convex, implying that it can be solved efficiently using, e.g., iterative methods. However, our aim is to design an online algorithm that chooses an allocation $\mathbf{x}_t$ for each time t without knowing future costs $\{f_{t'}(\cdot)\}_{t' > t}$.

Motivating applications

In this section, we give examples of applications that motivate the SOAD problem. We are particularly motivated by an emerging class of *carbon-aware sustainability* *problems* that have attracted significant attention in recent years—see the first example. SOAD also generalizes canonical online search problems such as one-way trading [78], making it broadly applicable across domains as we discuss in detail below. We focus on key components of each setting without exhaustively discussing idiosyncrasies, although we mention some extensions of SOAD in each setting. We defer a few more problem examples to Section 5.A.

Carbon-aware workload management in datacenters. Consider a delay-tolerant compute job scheduled on a distributed network of datacenters with the goal of minimizing the total carbon (CO_2) emissions of the job. Each job arrives with a deadline T that represents its required completion time, typically in minutes or hours. Service costs $f_t^{(u)}$ represent the carbon emissions of executing a workload at full speed in datacenter u at time t. The metric space (X, d) and the spatial movement cost $d(u_t, u_{t-1})$ capture the *carbon emissions overhead* of geographically migrating a compute workload between datacenters. The temporal switching cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\boldsymbol{\beta})}$ captures the carbon emissions overhead due to reallocation of resources (e.g., scaling up/down) within a single datacenter [158]. Finally, the constraint function $c(\mathbf{x}_t)$ encodes what fraction of the job is completed by a given scheduling decision $\mathbf{x}_t$. The topic of shifting compute in time and space to decrease its carbon footprint has seen significant attention in recent years [33, 155–158, 200–202], particularly for compute needs with long time scales and flexible deadlines (e.g., ML training), which realize the most benefits from temporal shifting. These works build on a long line of work advancing sustainable datacenters more broadly (e.g., in terms of energy efficiency), some of which leverage techniques from online optimization [37, 38, 164, 179–181, 203–210]. We comment that SOAD is the first online formulation that can model the necessary combined dimensions of spatial and temporal switching costs with deadlines. However, we also note that some aspects of this problem may not yet be fully captured by SOAD—for instance, it might be necessary to consider *multiple concurrent* batch workloads rather than a single one, resource contention, datacenter capacity constraints, or processing delays caused by migration that are not fully captured by the current formulation. In this sense, SOAD serves as a building block that could accommodate extensions to consider these aspects of the problem—we consider one such extension in Section 5.5.

Supply chain procurement. Consider a firm that must source a certain amount of a good before a deadline T, where the good is stored in several regional warehouses [211]. Service costs $f_t^{(u)}$ are proportional to the per unit cost of purchasing and transporting goods from warehouse u during time slot t. The metric space (X, d) and the spatial movement cost $d(u_t, u_{t-1})$ capture the overhead of switching warehouses, including, e.g., personnel costs to travel and inspect goods. The temporal switching cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\beta)}$ captures the overhead of stopping or restarting the purchasing and transport of goods from a single warehouse. Finally, the constraint function $c(\mathbf{x}_t)$ dictates how many goods can be shipped during time t according to purchasing decision $\mathbf{x}_t$. We note that in practice, the firm may need to purchase from multiple warehouses concurrently—they are restricted to purchase from only one in the strict SOAD formulation given above, but this can be relaxed without affecting the algorithms or results that we present in the rest of the chapter.

Mobile battery storage. Consider a mobile battery storage unit (e.g., a battery trailer [212]) that must service several discharge locations by, e.g., the end of the day (deadline T), with the goal of choosing when and where to discharge based on the value that storage can provide in that time and place. Service costs $f_t^{(u)}$ can represent the value of discharging at location u during time slot t (lower is better). The metric space (X, d) and the spatial movement cost $d(u_t, u_{t-1})$ capture the *overhead* (e.g., lost time or fuel cost) due to moving between locations, and the temporal switching cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\beta)}$ captures the overhead of connecting or disconnecting from a discharge point at a single location, including, e.g., cell degradation due to cycling [150]. The constraint function $c(\mathbf{x}_t)$ captures how much energy has been discharged during time t according to decision $\mathbf{x}_t$. The problem of maximizing the utility that mobile battery storage provides may be useful in, e.g., emergency relief situations where the main power grid has gone down. A light extension of SOAD may capture the case where the travel time from point to point significantly affects the feasible discharge time at the destination; such an extension would factor any lost time into the constraint function $c(\mathbf{x}_t)$.

Background & assumptions

In this section, we provide background on the competitive analysis used throughout the chapter and formalize our assumptions on the costs in SOAD, motivated by the structure of applications.

Competitive analysis. We evaluate the performance of an online algorithm for this problem via the *competitive ratio* [76, 145]: let $\mathtt{OPT}(\mathcal{I})$ denote the cost of an optimal offline solution for instance $\mathcal{I}$, and let $\texttt{ALG}(\mathcal{I})$ denote the cost incurred by running an online algorithm ALG over the same instance. Then the competitive ratio of ALG is defined as $\text{CR}(\texttt{ALG}) \coloneqq \sup_{\mathcal{I}\in\Omega} {}^{\texttt{ALG}(\mathcal{I})}/_{\texttt{OPT}(\mathcal{I})} \eqqcolon \eta$, where Ω is the set of all feasible inputs for the problem, and ALG is said to be η*-competitive*. Note that $\text{CR}(\texttt{ALG})$ is always at least 1, and a *smaller* competitive ratio implies that the online algorithm is guaranteed to be *closer* to the offline optimal solution. If ALG is randomized, we replace the cost $\texttt{ALG}(\mathcal{I})$ with the expected cost (over the randomness of the algorithm).

Assumption 5.1. *Each service cost function $f_t(\cdot)$ satisfies bounds, i.e., $f_t^{(u)} \in [c^{(u)}L, c^{(u)}U]$ for all $\texttt{ON}^{(u)} : u \in X$ and for all $t \in [T]$, where L and U are known, positive constants.*

This assumption encodes the physical idea that there exist upper and lower bounds on the service cost faced by the player. L and U are normalized by the throughput coefficient $c^{(a)}$ so that they can be independent of the amount of the deadline constraint satisfied by servicing the workload at a specific point $a \in X$.

Assumption 5.2. *The temporal switching cost factor is bounded by $\beta^{(a)} \leq \tau c^{(a)}$ for all $a \in X$. The normalized spatial distance between any two ON states is upper bounded by D, i.e., $D = \sup_{u,v\in X:u\neq v} \frac{d(u,v)}{\min\{c^{(u)},c^{(v)}\}}$. Further, we assume that $D + 2\tau \leq U - L$.*

In SOAD, τ represents the worst-case overhead of stopping, starting, or changing the rate of service at a single point, while D represents the worst-case overhead incurred by moving the allocation between the two most distant points. In, e.g., the applications mentioned above, we typically expect τ to be much smaller than D. Note that there are two ON states with a normalized distance greater than $U - L$, one of these states should be pruned from the metric, because moving the allocation between them would negate any benefit to the service cost. Specifically, recall that L and U give bounds on the total service cost of the workload, and consider an example of two points u, v that are normalized distance $D' > U - L$ apart, with $c^{(u)} = c^{(v)} = 1$, and $\tau = 0$. Let the starting point u and other point v have service costs that are the worst-possible and best-possible (i.e., U and L), respectively. Observe that if the player opts to move the allocation to v, they incur a movement cost of D' and a total objective of $L + D'$. In contrast, if they had stayed at point u, their total objective would be U, which is $< L + D'$ by assumption.

Connections to existing models and challenges

As discussed in the related work, SOAD exhibits similarities to two long-standing tracks of literature in online algorithms; however, SOAD is distinct from and cannot be solved by existing models.

The first of these is the work on the classic *metrical task systems* (MTS) problem introduced by Borodin et al. [76] and related forms, including smoothed online convex optimization (SOCO) [102]. In these works, an online player makes decisions with the objective of minimizing the sum of the service cost and switching cost. However, standard algorithms for MTS/SOCO are not designed to handle the type of long-term constraints (e.g., such as deadlines) that SOAD considers. Moreover, standard MTS and SOCO algorithms are designed to address either movement cost over points (i.e., $d(a_{t-1}, a_t)$) or temporal switching cost (i.e., $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\beta)}$). SOAD requires a spatiotemporal allocation that considers both types of switching costs simultaneously.

On the other hand, the *one-way trading* (OWT) problem introduced by El-Yaniv et al. [78] and related online knapsack problems [168, 176] consider online optimization with long-term constraints. To address these constraints, canonical algorithms use techniques such as threat-based or threshold-based designs to "hedge" between the extremes of quickly fulfilling the constraint and waiting for better opportunities that may not materialize. However, these works do not consider switching costs and rarely address multidimensional decision spaces.

The design of algorithms via competitive analysis for MTS/SOCO with long-term constraints has long been an open problem, and has seen only limited progress in a few recent works. These works have primarily leveraged techniques from online search, generalizing unidimensional problems such as k-search [192] (see Chapter 4) and OWT [193, 194] to additionally consider a *temporal switching cost*. A recent study considered a generalization to the multidimensional case, introducing the problem of convex function chasing with a long-term constraint (CFL) [195]. The authors of this work propose a competitive algorithm for CFL, although their results depend on a very specific metric structure (ℓ_1 vector spaces or weighted star metrics), which cannot be used to model the general spatial and temporal switching costs in SOAD. Furthermore, even in the multidimensional case, these existing works that consider switching costs and long-term constraints assume a *single source* of switching costs, i.e., that the cost to switch into a state making progress towards the long-term constraint is the same as the cost to switch out of that state. This type of structure and analysis fails in the SOAD setting due to the generality of the metric (i.e., moving to a new point while "switching ON" to complete some of the workload and "switching OFF" within the new point have different costs).

Worst-case competitive guarantees provide robustness against non-stationarities in the underlying environment, which may be desired for applications such as carbon-awareness due to the demonstrated non-stationarity associated with such signals [201]. However, algorithms that are purely optimized for worst-case guarantees are often overly pessimistic. To address this, we study learning-augmented design in the SOAD setting, which brings additional challenges. For instance, existing learning-augmented results for MTS/SOCO and OWT each leverage distinct algorithm design strategies based on different advice models that separately address features of their problem setting (i.e., switching costs, deadline constraints). These prior results naturally prompt questions about how to incorporate ML advice in a performant way that can simultaneously handle the generality of the switching costs in SOAD while ensuring that the deadline constraint is satisfied.

Preliminaries of technical foundations

In this section, we introduce and discuss techniques from different areas of the online algorithms literature that we use in subsequent sections to address the SOAD challenges discussed above.

Unifying arbitrary metrics. The generality of the metric space in SOAD is a key challenge that precludes the application of algorithm design techniques from prior work requiring specific metric structures. In classic MTS, the online player also makes decisions in an arbitrary metric space (X, d), which poses similar challenges for algorithm design. A key result used to address this is that of Fakcharoenphol et al. [213], who show that for any n-point metric space (X, d), there exists a probabilistic embedding into a *hierarchically separated tree* (HST) $\mathcal{T} = (V, E)$ with at most $O(\log n)$ distortion, i.e., $\mathbb{E}_{\mathcal{T}} \left[d^{(\mathcal{T})}(u, v) \right] \leq O(\log n) d(u, v)$ for any $u, v \in X$. For MTS, this result implies that any η-competitive algorithm for MTS *on trees* is immediately $O(\log n)\eta$-competitive in expectation for MTS on general n-point metrics, exactly by leveraging this embedding.

To solve MTS using such a tree, Bubeck et al. [142] consider a randomized algorithm on the leaves of $\mathcal{T}$ denoted by $\mathcal{L}$, where the nodes of $\mathcal{L}$ correspond to points in X. This randomized metric space is given by $(\Delta_{\mathcal{L}}, \mathbb{W}^1)$, where $\Delta_{\mathcal{L}}$ is the probability simplex over the leaves of $\mathcal{T}$, and $\mathbb{W}^1$ denotes the *Wasserstein-1 distance*. For two probability distributions $\mathbf{p}, \mathbf{p}' \in \Delta_{\mathcal{L}}$, the Wasserstein-1 distance is defined as $\mathbb{W}^1(\mathbf{p}, \mathbf{p}') := \min_{\pi_{x,x'} \in \Pi(\mathbf{p},\mathbf{p}')} \mathbb{E}\left[d(x, x')\right]$, where $\Pi(\mathbf{p}, \mathbf{p}')$ is the set of transport distributions over $\mathcal{L}^2$ with marginals $\mathbf{p}$ and $\mathbf{p}'$. A randomized algorithm that produces marginal distributions $\mathbf{p} \in \Delta_{\mathcal{L}}$ then *couples* consecutive decisions according to the optimal transport plan $\pi_{x,x'}$ defined by Wasserstein-1. Bubeck et al. [142] further show that $(\Delta_{\mathcal{L}}, \mathbb{W}^1)$ is bijectively isometric to a convex set K with a weighted ℓ_1 norm $\| \cdot \|_{\ell_1(\mathbf{w})}$ based on edge weights in the tree.

Metric tree embedding for SOAD. Prior approaches by Fakcharoenphol et al. [213] and Bubeck et al. [142] are able to manage the spatial movement cost from moving allocation between points. To further accommodate the temporal switching cost between `ON` and `OFF` states within a single point, we develop a probabilistic tree embedding $\mathcal{T} = (V, E)$ and the corresponding vector space $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$ in the following Definition 5.2.1 and Definition 5.2.2, respectively.

Definition 5.2.1 (Probabilistic tree embedding $\mathcal{T} = (V, E)$ for SOAD)**.** *Let (X, d) denote the underlying metric space over n points, and let $\mathcal{T}'$ denote an HST constructed on the points of X according to the method by Fakcharoenphol et al. [213]. Label the leaves of $\mathcal{T}'$ according to the n `ON` states, one for each point. Then the final tree $\mathcal{T}$ is constructed by adding n edges and n nodes to just the leaves of $\mathcal{T}'$—each new node represents the corresponding `OFF` state at that point, and the new edge is weighted according to the temporal switching cost at that point (i.e., $\beta^{(u)}$). The resultant "state set" $\mathcal{S}$ includes both the leaves of $\mathcal{T}$ (`OFF` states) and their immediate predecessors (`ON` states).*

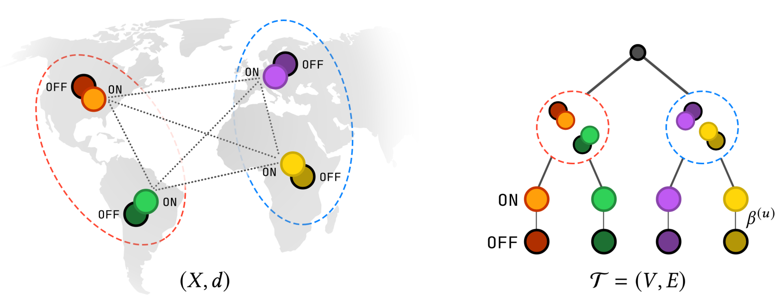


Figure 5.1: An illustration of the probabilistic tree embedding (Definition 5.2.1) for the motivating application. Points in the metric are represented as pairs of circles on the left. On the right, the first three levels of the tree approximate the metric space (X, d) [213], and the last level captures the `ON` / `OFF` structure of SOAD.

Note that $\mathcal{T}$ preserves distances between the points of X with expected $O(\log n)$ distortion, while the switching cost between `ON` and `OFF` states at a single point is preserved exactly. Our competitive algorithm (see Section 5.3) operates in a convex subset of a vector space constructed according to this HST embedding.

Definition 5.2.2 (Vector space $(K, \|\cdot\|_{\ell_1(\mathbf{w})})$). *Given a hierarchically separated tree $\mathcal{T} = (V, E)$ constructed according to Definition 5.2.1, with root $r \in V$, state set $\mathcal{S} \subseteq V$, and leaf set $\mathcal{L} \subset \mathcal{S}$, let $P^{(u)}$ denote the parent of any node $u \in V \setminus r$. Construct the following set:*

$$K := \left\{ \mathbf{k} \in \mathbb{R}^{|V|} \;\middle|\; \begin{aligned} &\mathbf{k}^{(r)} = 1, \\ &\mathbf{k}^{(u)} = \sum_{v : P^{(v)} = u} \mathbf{k}^{(v)} \quad \text{for all } u \in V \setminus \mathcal{L}, \\ &\mathbf{k}^{(u)} \in [0, 1] \qquad\qquad\quad \text{for all } u \in \mathcal{S} \end{aligned} \right\}.$$

Let $\mathbf{w}$ be a non-negative weight vector on vertices of $\mathcal{T}$, where $\mathbf{w}^{(r)} = 0$ and $\mathbf{w}^{(u)} > 0$ for all $u \in V \setminus r$. Recall that the edges of $\mathcal{T}$ are weighted—we let $\mathbf{w}^{(u)}$ denote the weight of edge $\{P^{(u)}, u\}$, and define a weighted ℓ_1 norm as $\|\mathbf{k}\|_{\ell_1(\mathbf{w})} := \sum_{u \in V} \mathbf{w}^{(u)} |\mathbf{k}^{(u)}|$, for any $\mathbf{k} \in K$. Finally, we define a linear map from $\Delta_{\mathcal{S}}$ to K (and its corresponding inverse), given by a matrix map $\Phi : \mathbb{R}^{2n} \to \mathbb{R}^{|V|}$. We let $A^{(u)}$ denote the set of node u's ancestors in $\mathcal{T}$, and (with a slight abuse of notation) let $\mathcal{S}(i) : \{1, \ldots, 2n\} \to \mathcal{S}$ and $K(i) : \{1, \ldots, |V|\} \to V$ denote indexing maps that recover the object in $\mathcal{S}$ or V, respectively. Then $\Phi \in \mathbb{R}^{|V| \times 2n}$ and $\Phi^{-1} \in \mathbb{R}^{2n \times |V|}$ are defined as

$$\Phi_{i,j} := \begin{cases} 1 & \text{if } K(i) = r \\ 1 & \text{if } K(i) = \mathcal{S}(j) \\ 1 & \text{if } K(i) \in \mathcal{S} \text{ and } P^{(\mathcal{S}(j))} = K(i) \\ 1 & \text{if } K(i) \in V \setminus \mathcal{S} \text{ and } K(i) \in A^{(\mathcal{S}(j))} \\ 0 & \text{otherwise} \end{cases}$$

and

$$\Phi^{-1}_{i,j} := \begin{cases} 1 & \text{if } \mathcal{S}(i) = K(j) \\ -1 & \text{if } K(j) \in \mathcal{L} \text{ and } P^{(K(j))} = \mathcal{S}(i) \\ 0 & \text{otherwise.} \end{cases}$$

In words, Φ maps a distribution over $\Delta_{\mathcal{S}}$ to the corresponding vector in K by accumulating probability mass upwards from the leaves of $\mathcal{T}$ towards the root. Φ^{-1}

reverses this by selecting the appropriate indices for $u \in \mathcal{S}$ from K, and recovers probabilities by subtracting the mass at the `OFF` state from the `ON` state (since the `OFF` state is a leaf, the `ON` state accumulates its probability in K).

Randomized algorithm for SOAD. We define some shorthand notation. For a decision $\mathbf{k} \in K$, $\mathbf{p} = \Phi^{-1}\mathbf{k}$ gives a corresponding probability distribution on $\Delta_{\mathcal{S}}$. Note that $\mathcal{X} \subset \Delta_{\mathcal{S}} \subset \mathbb{R}^{2n}$, and by linearity of expectation, the service and constraint functions $f_t(\cdot), c(\cdot) : \mathcal{X} \to \mathbb{R}$ remain well-defined (in expectation) on $\Delta_{\mathcal{S}}$. Within K, we let $\overline{f}_t(\mathbf{k}) = f_t(\Phi^{-1}\mathbf{k})$ and $\overline{c}(\mathbf{k}) = c(\Phi^{-1}\mathbf{k})$. For a given *starting point* $s \in X$, we slightly abuse notation and let $\delta_s \in \Delta_{\mathcal{S}}$ denote the Dirac measure supported at $\texttt{OFF}^{(s)}$. Recall that the SOAD formulation specifies an allocation that is *discrete* in terms of choosing a point in the metric, and *fractional* in terms of the resource allocation at a given point. To capture this structure while using the embedding results discussed above, we consider a *mixed setting* that is probabilistic in spatial assignment but deterministic in the `ON` / `OFF` allocation. We state the equivalence of this setting and the fully probabilistic one below, deferring the proof to Section 5.C.

Theorem 5.2.3. *For a randomized* SOAD *decision* $\mathbf{p}_t \in \Delta_{\mathcal{S}}$*, the expected cost is equivalent if a point in* X *is first chosen probabilistically and the* `ON` / `OFF` *probabilities at that point are interpreted as (deterministic) fractional allocations.*

Enforcing a deadline constraint using pseudo-cost. Existing algorithms for MTS-type problems are not designed to handle a deadline constraint while remaining competitive. For SOAD, we draw from the *pseudo-cost minimization* [176] approach for online search problems, where the player is subject to a long-term *buying/selling* constraint that poses similar algorithmic challenges.

Under the pseudo-cost framework, we start by assuming that a *mandatory allocation* condition exists to strictly enforce the deadline constraint. Let $z^{(t)}$ denote the fraction of the deadline constraint satisfied (in expectation) up to time t (we henceforth call this the ***utilization***). To avoid violating the constraint, a mandatory allocation begins at time j, as soon as $(T - (j+1))c^{(u)} < (1 - z^{(j)})\ \forall u \in X$, i.e., when the remaining time after the current slot would be insufficient to satisfy the constraint. Note that in practice, $z^{(j)}$ would be replaced by the actual constraint satisfaction so far. During the mandatory allocation, a cost-agnostic player takes control and makes maximal allocation decisions to ensure the workload is finished before the deadline.

Intuitively, the mandatory allocation complicates competitive analysis—in the worst-case, an adversary can present the worst service cost (U) during the final steps. The key idea behind pseudo-cost minimization is to rigorously characterize a *tradeoff* between completing the constraint "too early" and waiting too long (i.e., risking a mandatory allocation) using a *pseudo-cost function*. Such a function takes the lower and upper bounds on service cost (i.e., L and U) as parameters, and assigns a *pseudo-cost* to each increment of progress towards the constraint. In an algorithm, this function is used by solving a small minimization problem at each step, whose objective considers the true cost of a potential decision and an integral over the pseudo-cost function—generating decisions using this technique creates a connection between the utilization and the best service cost encountered throughout the sequence, ensuring that the algorithm completes "exactly enough" of the constraint before mandatory allocation in order to achieve a certain competitive ratio against the best service cost, which is a lower bound on OPT. We also note that in the learning-augmented setting, the pseudo-cost minimization problem can be combined with a *consistency constraint*, as shown by [195], to integrate certain forms of advice without losing the robust (i.e., competitive) qualities of the pseudo-cost.

5.3 PCM: A Competitive Online Algorithm

This section presents a randomized competitive algorithm for SOAD that leverages the metric tree embeddings and pseudo-cost minimization design discussed above in Section 5.2. We further show that our algorithm achieves a competitive ratio that is optimal for SOAD up to log factors.

Algorithm description

We present a randomized pseudo-cost minimization algorithm (PCM) in Algorithm 10. PCM operates on the metric space $(K, \|\cdot\|_{\ell_1(\mathbf{w})})$ defined in Definition 5.2.2 and extends the original pseudo-cost minimization framework [176] to address the setting where the decision space is given by an arbitrary convex set K with distances given by $\|\cdot\|_{\ell_1(\mathbf{w})}$.

We define a pseudo-cost function $\psi(z) : [0, 1] \to [L, U]$, where z is the utilization (i.e., the completed fraction of the deadline constraint in expectation). Our construction of ψ takes advantage of additional structure in the SOAD setting—this function depends on the parameters of the SOAD problem, including U, L, D, and τ specified in Assumptions 5.1 and 5.2.

Algorithm 10: Pseudo-cost minimization algorithm for SOAD (PCM)

Input: constraint function $c(\cdot)$, convex set K with distance metric $\|\cdot\|_{\ell_1(\mathbf{w})}$, pseudo-cost function $\psi(z)$, starting `OFF` state $s \in \mathcal{S}$.

initialize: $z^{(0)} = 0$; $\mathbf{k}_0 = \Phi\delta_s$; $\mathbf{p}_0 = \delta_s$.

while cost function $f_t(\cdot)$ is revealed and $z^{(t-1)} < 1$ **do**

Solve **pseudo-cost minimization** problem:

$$\mathbf{k}_t = \underset{\mathbf{k}\in K:\overline{c}(\mathbf{k})\leq 1-z^{(t-1)}}{\arg\min} \overline{f}_t(\mathbf{k}) + \|\mathbf{k}-\mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)}+\overline{c}(\mathbf{k})} \psi(u)du, \tag{5.2}$$

$$\mathbf{p}_t = \Phi^{-1}\mathbf{k}_t. \tag{5.3}$$

Update utilization $z^{(t)} = z^{(t-1)} + c(\mathbf{p}_t)$.

end

Definition 5.3.1 (Pseudo-cost function ψ for SOAD)**.** *For a given parameter $\eta > 1$, the pseudo-cost function is defined as $\psi(z) = U - \tau + (U/\eta - U + D + \tau)\exp(z/\eta)$, $z \in [0, 1]$.*

Given the pseudo-cost function from Definition 5.3.1, PCM solves a minimization problem (5.2) at each step t to generate a decision $\mathbf{k}_t \in K$; the objective of this problem is to minimize a combination of the per-step cost plus a pseudo-cost term that encourages (deadline) constraint satisfaction. At a high level, the ψ term enforces that $\mathbf{k}_t$ satisfies "exactly enough" of the deadline constraint (in expectation) to make adequate progress and maintain an expected competitive ratio of η against the current estimate of `OPT`, without "overbuying" and preventing better costs from being considered in the future. At a glance, it is not obvious that the pseudo-cost minimization problem is straightforward to actually solve in practice. In the following, we show that (5.2) is a convex minimization problem.

Theorem 5.3.2. *Under the assumptions of* SOAD*, the pseudo-cost minimization* (5.2) *is a convex minimization problem.*

We defer the proof of Theorem 5.3.2 to Section 5.D. At a high-level, the result implies that the solution to (5.2) can be found efficiently using convex programming techniques [214, 215]. Compared to prior works [176, 195], our design of ψ differentiates between the spatial movement cost and temporal switching cost (in particular, D only appears within the exponential, while τ appears inside and outside of the exponential term). This removes a source of pessimism—when PCM makes a decision to move to a distant point in the metric (i.e., paying a worst-case factor of

D), it can safely assume that it will only have to pay a factor of τ to switch OFF if the next cost function is bad. This allows PCM to take advantage of spatially distributed OFF states, where all of the existing works that use a pseudo-cost paradigm for temporal load-shifting cannot.

We note that PCM's decisions in K are marginal probability distributions over $\Delta_{\mathcal{S}}$; we now briefly detail how feasible deterministic decisions in $\mathcal{X}$ are extracted from these outputs. We assume the player interprets distributions according to a mixed random / fractional setting (see Theorem 5.2.3), allowing them to make fractional resource allocation decisions within a single point while the allocation is probabilistically assigned to a single point. At each timestep, PCM generates $\mathbf{p}_t \in \Delta_{\mathcal{S}}$. We let $\mathbf{r}_t := \{r_t^{(u)} \leftarrow p_t^{\mathtt{ON}(u)} + p_t^{\mathtt{OFF}(u)} : u \in X\}$ aggregate the ON / OFF probabilities at each location of X. Given this *spatial* distribution over X, consecutive decisions should be jointly distributed according to the optimal transport plan between $\mathbf{r}_{t-1}$ and $\mathbf{r}_t$, given by $(\mathbf{r}_t, \mathbf{r}_{t-1}) \sim \pi_t := \arg\min_{\pi \in \Pi(\mathbf{r}_t, \mathbf{r}_{t-1})} \mathbb{E}\left[d(u_t, u_{t-1})\right]$, where $(u_t, u_{t-1}) \sim \pi_t$ and $\Pi(\mathbf{r}_t, \mathbf{r}_{t-1})$ is the set of distributions over X^2 with marginals $\mathbf{r}_t$ and $\mathbf{r}_{t-1}$. If the decision-maker couples decisions according to π_t, then the *expected* spatial movement cost of the deterministic decisions is equivalent to $\mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1})$, the spatial Wasserstein-1 distance between $\mathbf{r}_t$ and $\mathbf{r}_{t-1}$. Given a previous deterministic point assignment u_{t-1}, the player can obtain the point assignment u_t by sampling through the conditional distribution $\pi_t(u_t | u_{t-1})$. The fractional ON / OFF allocation in $\mathbf{x}_t$ at the chosen location $u_t \in X$ is then given by $p_t^{\mathtt{ON}(u)} / r_t^{(u)}$ and $p_t^{\mathtt{OFF}(u)} / r_t^{(u)}$, respectively; by Theorem 5.2.3, this gives that $\mathbb{E}\left[g(\mathbf{x}_t, \mathbf{x}_{t-1})\right] = \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1})$.

Main results

In Theorem 5.3.3, we state a bound on the competitive ratio of PCM.

Theorem 5.3.3. *Under Assumptions 5.1 and 5.2,* PCM *is $O(\log n)\eta$-competitive for* SOAD*, where η is the solution to $\ln\left(\frac{U-L-D-2\tau}{U-U/\eta-D}\right) = \frac{1}{\eta}$ and given by:*

$$\eta := \left[W\left(\frac{(D + L - U + 2\tau) \exp\left(\frac{D-U}{U}\right)}{U} \right) + \frac{U - D}{U} \right]^{-1}, \tag{5.4}$$

where W is the Lambert W function [216], and the $O(\log n)$ factor is due to the tree embedding [213].

Compared to previous works such as MTS and OWT, the competitive bound in Theorem 5.3.3 compares favorably. In particular, the upper bound is *better* than one might expect from, e.g., combining the bounds of MTS and OWT. For the minimization variant of OWT, the optimal competitive ratio due to Lorenz et al. [79] is $\left[W\left(\left(1/\theta - 1\right)e^{-1}\right) + 1\right]^{-1}$, where θ is defined as M/m, and $M \geq m$ are bounds on the prices (i.e., $(m, M) \approx (L, U)$). For MTS, the randomized state-of-the-art due to Bubeck et al. [142] is $O(\log^2 n)$. Asymptotically, compared to both of these bounds, η "loses" a log factor depending on the number of points in the metric, and it is known that $W(x) \sim \ln(x)$ as $x \to \infty$ [172, 173]. Compared to OWT, η adds a dependency on D and τ, parameters describing the cost due to the metric and switching, but we note that Assumption 5.2 (i.e., bounds on D and τ in terms of U and L) prevents the competitive ratio from significantly increasing.

Given the result in Theorem 5.3.3, a natural question is whether any online algorithm for SOAD can achieve a better competitive bound. We answer this in the negative, showing that PCM's competitive ratio is the best achievable up to log factors that are due to the metric embedding. In particular, we show a class of difficult instances on which no algorithm can achieve a competitive ratio better than η; since the definition of the competitive ratio covers all valid inputs, this gives a corresponding lower bound on the competitive ratio of any algorithm for SOAD.

Theorem 5.3.4. *For any U, L, τ, and $D \in [0, (U - L))$, there exists a set of* SOAD *instances on a weighted star on which no algorithm* $\texttt{ALG}$ *can achieve* $\texttt{ALG}/\texttt{OPT}$ *better than η (for η defined in (5.4)).*

Proof overviews

We give proof sketches of Theorems 5.3.3 and 5.3.4, relegating the full proofs of both to Section 5.D.

Proof Sketch of Theorem 5.3.3. To show this result, we give two lemmas to characterize the cost of $\texttt{OPT}$ and the expected cost of PCM, respectively. First, note that the solution given by PCM is feasible, by definition of the mandatory allocation (i.e., $\sum_{t=1}^{T} c(\mathbf{p}_t) = 1$). On an arbitrary SOAD instance $\mathcal{I} \in \Omega$, we denote the final utilization (before the mandatory allocation) by $z^{(j)}$.

Lemma 5.3.5. *The offline optimum is lower bounded by* $\texttt{OPT}(\mathcal{I}) \geq \frac{\max\left\{\psi(z^{(j)}) - D, L\right\}}{O(\log n)}$.

We can show by contradiction that for any instance, the definition of the pseudo-cost minimization enforces that $\psi(z^{(j)}) - D$ is a lower bound on the *best service cost* seen in the sequence (see (5.12)). Note that the best choice for $\texttt{OPT}$ is to service the entire workload at the minimum cost (if it is feasible). This yields a corresponding lower bound on $\texttt{OPT}$—formally, $\texttt{OPT}(\mathcal{I}) \geq {}^{\max\{\psi(z^{(j)})-D,\ L\}}/_{O(\log n)}$, where the log factor appears due to the distortion in the metric tree embedding.

Lemma 5.3.6. PCM*'s expected cost is bounded by* $\mathbb{E}[\text{PCM}(\mathcal{I})] \leq \int_0^{z^{(j)}} \psi(u)du + (1 - z^{(j)})U + \tau z^{(j)}$.

The definition of the pseudo-cost provides an automatic bound on the expected cost incurred during any timestep where progress is made towards the deadline constraint (i.e., whenever the service cost is non-zero).

We show that $\tau z^{(j)}$ is an upper bound on the *excess cost* that can be incurred by PCM in the other timesteps (i.e., due to temporal switching costs, see (5.13)). Summing over all timesteps, this gives that the expected cost of PCM is upper bounded by $\int_0^{z^{(j)}} \psi(u)du + (1 - z^{(j)})U + \tau z^{(j)}$, where $(1 - z^{(j)})U$ is due to the mandatory allocation.

Combining the two lemmas and using the definition of the pseudo-cost function to observe that $\int_0^{z^{(j)}} \psi(u)du + (1 - z^{(j)})U + \tau z^{(j)} \leq \eta\left[\psi(z^{(j)}) - D\right]$ (see (5.14)) completes the proof. □

Proof Sketch of Theorem 5.3.4. In Definition 5.D, we define a class of *y-adversaries* denoted by $\mathcal{G}_y$ and $\mathcal{A}_y$ for $y \in [L, U]$, along with a corresponding weighted star metric X that contains n points, each with 2 states ($\texttt{ON}$ and $\texttt{OFF}$), where the distance between any two points in the metric is exactly D. These adversaries present cost functions at the $\texttt{ON}$ states of $\mathcal{X}$ in an adversarial order that forces an online algorithm to incur a large switching cost. The $\mathcal{G}_y$ adversary presents a cost function at each step that is "bad" (i.e., U) in all $\texttt{ON}$ states except for one which is *not* at the starting point or the current state of online algorithm $\texttt{ALG}$. The $\mathcal{A}_y$ adversary starts by exactly mimicking $\mathcal{G}_y$ and presenting "good" cost functions at distant points, before eventually presenting "good" cost functions at the starting point. Both adversaries present "good" cost functions in an adversarial non-increasing order, such that the optimal solutions approach y—formally, $\texttt{OPT}(\mathcal{G}_y) \to \min\{y + D + \tau, U\}$, and $\texttt{OPT}(\mathcal{A}_y) \to y$. By competing against both adversaries simultaneously, this construction captures a tradeoff between being too eager/reluctant to move away from the starting point.

Under this special metric and class of adversaries, the cost of any (potentially randomized) online algorithm $\texttt{ALG}$ can be fully described by two arbitrary *constraint satisfaction functions* $s(y), t(y) : [L, U] \to [0, 1]$ (see (5.16)), where each function corresponds to one of two *stages* of the adversary (i.e., "good" cost functions arriving at spatially distant points, or at the starting point). For `ALG` to be $\eta^\star$-competitive (where $\eta^\star$ is unknown), we give corresponding conditions on $s(y)$ and $t(y)$ expressed as differential inequalities (see (5.17)). By applying Grönwall's Inequality [217, Theorem 1, p. 356], this gives a *necessary condition* such that $\eta^\star$ must satisfy: $\eta^\star \ln\left(\frac{U-L-D-2\tau}{U-U/\eta^\star-D}\right) - \frac{\eta^\star D+\eta^\star 2\tau}{U/\eta^\star-U+D} \leq s(L) \leq 1-t(L) \leq 1-\frac{\eta^\star D+\eta^\star 2\tau}{U/\eta^\star-U+D}$.

The optimal $\eta^\star$ is obtained by solving for the transcendental equation that arises when the inequalities are binding, yielding the result. □

5.4 ST-CLIP: A Learning-Augmented Algorithm

In this section, we consider how a learning-augmented algorithm for SOAD can leverage *untrusted advice* to improve on the average-case performance of PCM while retaining worst-case guarantees. For learning-augmented algorithms, competitive ratio is interpreted via the notions of *consistency* and *robustness* [69, 70]. Letting `ALG` denote a learning-augmented online algorithm provided with advice denoted by Adv, `ALG` is said to be α-**consistent** if it is α-competitive with respect to Adv, and γ-**robust** if it is γ-competitive with respect to `OPT` when given any advice (i.e., regardless of Adv's performance). We present ST-CLIP (see Algorithm 11), which uses an adaptive optimization-based approach combined with the robust design of PCM to achieve an optimal consistency-robustness tradeoff. We start by formally defining the advice model we use below.

Definition 5.4.1 (Black-box advice model for SOAD)**.** *For a given* SOAD *instance* $\mathcal{I} \in \Omega$*, we let* $\text{Adv}(\mathcal{I})$ *denote* ***untrusted black-box decision advice****, i.e.,* $\text{Adv}(\mathcal{I}) := \{\mathbf{a}_t \in \Delta_{\mathcal{S}} : t \in [T]\}$*. If* Adv *is correct, a player that plays* $\mathbf{a}_t$ *at each step attains the optimal solution (*$\text{Adv}(\mathcal{I}) = \texttt{OPT}(\mathcal{I})$*).*

Although $\mathbf{a}_t$ is defined on the probability simplex $\Delta_{\mathcal{S}}$, a deterministic Adv at time t is given by combining the Dirac measure supported at a point and a specific `ON` / `OFF` allocation. We henceforth assume that Adv is *feasible*, satisfying the constraint $(\sum_{t=1}^{T} c(\mathbf{a}_t) \geq 1)$.

While it is not obvious a machine learning model could directly provide such feasible predictions, in practice, we leverage the black-box nature of the definition to combine, e.g., machine-learned predictions of relevant costs with a post-processing pipeline that solves for a *predicted optimal solution* (see Section 5.6).

ST-CLIP: an optimal learning-augmented algorithm

We present ST-CLIP (**s**patio**t**emporal **c**onsistency-**li**mited **p**seudo-cost minimization, Algorithm 11), which obtains a near-optimal robustness-consistency tradeoff for SOAD (Theorem 5.4.4).

ST-CLIP takes a hyperparameter $\varepsilon \in (0, \eta - 1]$, which parameterizes a tradeoff between following the untrusted advice ($\varepsilon \to 0$) and prioritizing robustness ($\varepsilon \to \eta - 1$). We start by defining a *target robustness factor* $\gamma^{(\varepsilon)}$, which is the unique solution to the following equation:

$$\gamma^{(\varepsilon)} = \varepsilon + \frac{U}{L} - \frac{\gamma^{(\varepsilon)}(U - L + D)}{L} \ln\left(\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right). \tag{5.5}$$

We note that $\gamma^{(\varepsilon \to 0)} \to U/L$, which is a trivial competitive ratio for any mandatory allocation scheme (i.e., if the entire constraint is satisfied at the deadline for the worst price U). The precise value of $\gamma^{(\varepsilon)}$ originates from a robustness-consistency lower bound (Theorem 5.4.4), and ST-CLIP uses it to define a pseudo-cost function $\psi^{(\varepsilon)}$ that enforces $\gamma^{(\varepsilon)}$-robustness in its decisions.

Definition 5.4.2 (Pseudo-cost function $\psi^{(\varepsilon)}$ for SOAD)**.** *For $\rho \in [0, 1]$ and $\gamma^{(\varepsilon)}$ given by (5.5), let $\psi^{(\varepsilon)}(\rho)$ be defined as: $\psi^{(\varepsilon)}(\rho) = U + D - \tau + (U+D/\gamma^{(\varepsilon)} - U + D + \tau)\exp(\rho/\gamma^{(\varepsilon)})$.*

Similarly to PCM (see Section 5.3), $\psi^{(\varepsilon)}$ is used in a minimization problem solved at each timestep to obtain a decision. However, since ST-CLIP must also consider the actions of Adv, it follows the consistency-limited pseudo-cost minimization paradigm, which places a *consistency constraint* on the aforementioned minimization. This constraint enforces that ST-CLIP always satisfies $(1 + \varepsilon)$-consistency, which is salient when Adv is close to optimal. Within this feasible set, the pseudo-cost minimization drives ST-CLIP towards decisions that are "as robust as possible."

Additional challenges in algorithm design. In contrast to prior applications of the CLIP technique [195], the SOAD setting introduces a disconnect between the advice and the robust algorithm (e.g., PCM); specifically, Adv furnishes decisions that are supported on the (randomized) metric $\Delta_{\mathcal{S}}$, while PCM makes decisions on the tree metric given by $(K, \|\cdot\|_{\ell_1(\mathbf{w})})$. Since the CLIP technique effectively "combines" Adv with a robust algorithm, this poses a challenge in the SOAD setting, introducing a $O(\log n)$ dependency in the consistency bound.[1] With ST-CLIP (see Algorithm 11), we carefully decouple the "advice side" and the "robust side" of the CLIP technique to achieve a $(1 + \varepsilon)$-consistency bound. While an $O(\log n)$ factor is likely unavoidable on arbitrary metrics in the adversarial setting of robustness (e.g., as is the case for metrical task systems [142, 177]), the non-adversarial setting of consistency (i.e., when advice is correct) implies that such a factor should be avoidable. Furthermore, removing a factor of $O(\log n)$ allows ST-CLIP to achieve consistency arbitrarily close to 1, which is often desirable in practice when the advice is often of high quality.

To accomplish this decoupling, ST-CLIP uses the pseudo-cost minimization defined in (5.6) to generate intermediate "robust decisions" ($\mathbf{k}_t \in K$) on the tree embedding (see Definition 5.2.1). These decisions are converted into marginal probability distributions on the underlying simplex (i.e., $\mathbf{p}_t \in \Delta_{\mathcal{S}}$) *before* evaluating the consistency constraint. Since Adv also specifies decisions on $\Delta_{\mathcal{S}}$, this decoupling allows the constraint to directly compare the running cost of ST-CLIP and Adv, without losing a $\log(n)$ factor due to the tree embedding. To hedge against worst-case scenarios that might cause ST-CLIP to violate the desired $(1+\varepsilon)$-consistency, the consistency constraint in (5.7) extrapolates the cost of such scenarios on the randomized decision space $\Delta_{\mathcal{S}}$.

Notation. We introduce some shorthand notation to simplify the algorithm's pseudocode as follows: we let SC_t denote the expected cost of ST-CLIP's decisions up to time t, i.e., $\text{SC}_t := \sum_{j=1}^{t} f_j(\mathbf{p}_j) + \mathbb{W}^1(\mathbf{p}_j, \mathbf{p}_{j-1})$, and similarly let Adv_t denote the (expected) cost of the advice up to time t: $\text{Adv}_t := \sum_{j=1}^{t} f_j(\mathbf{a}_j) + \mathbb{W}^1(\mathbf{a}_j, \mathbf{a}_{j-1})$. As $z^{(t)}$ denotes the utilization of ST-CLIP, we let $A^{(t)}$ denote the utilization of Adv at time t (i.e., the expected fraction of the deadline constraint satisfied by Adv so far). In addition, ST-CLIP also keeps track of a *robust pseudo-utilization* $\rho^{(t)} \in [0, 1]$; this term describes the portion of its decisions thus far that are attributable to the robust pseudo-cost minimization, and we have $\rho^{(t)} \leq z^{(t)}$ for all $t \in [T]$. This quantity is updated according to the $\overline{\mathbf{k}}_t$ that solves *an unconstrained minimization* in (5.8), ensuring that when Adv has incurred a "bad" service cost that would otherwise not be considered by the robust algorithm, the pseudo-cost $\psi^{(\varepsilon)}$ maintains some headroom to accept better service costs that might arrive in the future.

[1]Directly applying the CLIP technique to the $(K, \|\cdot\|_{\ell_1(\mathbf{w})})$ decision space considered in PCM yields an unremarkable consistency upper bound of $O(\log n)(1+\varepsilon)$, due to the distortion in the tree metric.

Consistency constraint intuition. Within the constraint (5.7), ST-CLIP encodes several "worst-case" scenarios that threaten the desired consistency bound. The first three terms on the left-hand side and the $\textsc{Adv}_t$ term on the right-hand side consider the actual cost of ST-CLIP and $\textsc{Adv}$ so far, along with the current decision under consideration, where the expected switching cost is captured by the optimal transport plan with respect to the previous decision.

The $\mathbb{W}^1(\mathbf{p}, \mathbf{a}_t)$ term on the left-hand side charges ST-CLIP in advance for the expected movement cost between it and the advice—the reasoning for this term is to hedge against the case where the constraint becomes binding in future steps, thus requiring ST-CLIP to *move* and follow $\textsc{Adv}$. If the constraint did not charge for this potential movement cost in advance, a binding constraint in future timesteps might result in either an infeasible problem or a violation of $(1+\varepsilon)$-consistency. The $\tau c(\mathbf{a}_t)$ term on both sides charges both $\textsc{Adv}$ and ST-CLIP in advance for the temporal switching cost they must incur before the deadline—ST-CLIP is charged according to $\mathbf{a}_t$ (as opposed to $\mathbf{p}_t$) to continue hedging against the case where it must move to follow the advice in future timesteps, finally paying $\tau c(\mathbf{a}_t)$ to switch `OFF` at the deadline.

On the right-hand side, the $(1 - A^{(t)})L$ term assumes that $\textsc{Adv}$ can satisfy the remaining deadline constraint at the best marginal service cost L. In contrast, the final terms on the left-hand side $(1 - z^{(t-1)} - c(\mathbf{p}))L + \max((A^{(t)} - z^{(t-1)} - c(\mathbf{p})),\ 0)(U - L)$ balance between two scenarios—namely, they assume that ST-CLIP can satisfy a fraction of the remaining constraint (up to $(1 - A^{(t)})$) at the best cost by following $\textsc{Adv}$, but any excess beyond this (given by $(A^{(t)} - z^{(t)})$), must be fulfilled at the worst service cost U, possibly during a mandatory allocation.

At a high level, ST-CLIP's constraint on $\Delta_{\mathcal{S}}$ combined with the pseudo-cost minimization on $(K, \|\cdot\|_{\ell_1(\mathbf{w})})$ generates decisions that are *maximally robust* while preserving consistency.

Main results

In Theorem 5.4.3, we give upper bounds on the robustness and consistency of ST-CLIP.

Theorem 5.4.3. *For any $\varepsilon \in (0, \eta - 1]$,* ST-CLIP *is $(1+\varepsilon)$-consistent and $O(\log n)\gamma^{(\varepsilon)}$-robust for* SOAD*, where $\gamma^{(\varepsilon)}$ is the solution to* (5.5).

Algorithm 11: ST-CLIP (spatiotemporal consistency-limited pseudo-cost minimization) for SOAD

Input: Consistency parameter ε, constraint function $c(\cdot)$, pseudo-cost $\psi^{(\varepsilon)}(\cdot)$, starting `OFF` state $s \in \mathcal{S}$.

initialize:

$z^{(0)} = 0;\ \rho^{(0)} = 0;\ A^{(0)} = 0;\ \mathrm{SC}_0 = 0;\ \textsc{Adv}_0 = 0;\ \mathbf{k}_0 = \Phi\delta_s;\ \mathbf{p}_0 = \mathbf{a}_0 = \delta_s.$

while cost function $f_t(\cdot)$ is revealed, untrusted advice $\mathbf{a}_t$ is revealed, and $z^{(t-1)} < 1$ **do**

Update advice cost $\textsc{Adv}_t \leftarrow \textsc{Adv}_{t-1} + f_t(\mathbf{a}_t) + \mathbb{W}^1(\mathbf{a}_t, \mathbf{a}_{t-1})$ and advice utilization $A^{(t)} \leftarrow A^{(t-1)} + c(\mathbf{a}_t)$.

Solve the **constrained** pseudo-cost minimization problem:

$$\mathbf{k}_t = \operatorname*{arg\,min}_{\mathbf{k} \in K : \overline{c}(\mathbf{k}) \le 1 - z^{(t-1)}} \overline{f}_t(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{\rho^{(t-1)}}^{\rho^{(t-1)} + \overline{c}(\mathbf{k})} \psi^{(\varepsilon)}(u)\, du \tag{5.6}$$

$$\text{such that } \mathbf{p} \leftarrow \Phi^{-1}\mathbf{k} \text{ and}$$

$$\begin{aligned} &\mathrm{SC}_{t-1} + f_t(\mathbf{p}) + \mathbb{W}^1(\mathbf{p}, \mathbf{p}_{t-1}) + \mathbb{W}^1(\mathbf{p}, \mathbf{a}_t) + \tau c(\mathbf{a}_t) \\ &+ (1 - z^{(t-1)} - c(\mathbf{p}))L + \max\{A^{(t)} - z^{(t-1)} - c(\mathbf{p}), 0\}(U - L) \\ &\le (1 + \varepsilon)[\textsc{Adv}_t + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L]. \end{aligned} \tag{5.7}$$

Update running cost $\mathrm{SC}_t \leftarrow \mathrm{SC}_{t-1} + f_t(\mathbf{p}_t) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1})$ and utilization $z^{(t)} \leftarrow z^{(t-1)} + c(\mathbf{p}_t)$.

Solve the **unconstrained** pseudo-cost minimization problem:

$$\tilde{\mathbf{k}}_t = \operatorname*{arg\,min}_{\mathbf{k} \in K : \overline{c}(\mathbf{k}) \le 1 - z^{(t-1)}} \overline{f}_t(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{\rho^{(t-1)}}^{\rho^{(t-1)} + \overline{c}(\mathbf{k})} \psi^{(\varepsilon)}(u)\, du \tag{5.8}$$

Update the *robust pseudo-utilization* $\rho^{(t)} \leftarrow \rho^{(t-1)} + \min\{\overline{c}(\tilde{\mathbf{k}}_t), c(\mathbf{p}_t)\}$.

end

Furthermore, we give a lower bound on the best achievable robustness ratio for any $(1 + \varepsilon)$-consistent algorithm, using a construction of a challenging metric space and service cost sequence. Since robustness and consistency are defined over all valid inputs (i.e., based on competitive ratio), this result characterizes the *optimal robustness-consistency tradeoff*, and implies that ST-CLIP matches the optimal up to log factors that are due to the metric embedding.

Theorem 5.4.4. *Given untrusted advice* $\textsc{Adv}$ *and* $\varepsilon \in (0, \eta - 1]$*, any* $(1 + \varepsilon)$*-consistent learning-augmented algorithm for* SOAD *is at least* $\gamma^{(\varepsilon)}$*-robust, where* $\gamma^{(\varepsilon)}$ *is defined in* (5.5).

Learning-augmentation and robustness-consistency tradeoffs have been previously considered in both MTS and OWT; we briefly review how Theorem 5.4.4 compares.

For MTS, Christianson et al. [177] show that for $\varepsilon \in (0, 1]$, any $(1 + \varepsilon)$-consistent algorithm must be $2^{\Omega(1/\varepsilon)}$-robust (see Chapter 3). While optimal tradeoffs for the minimization variant of OWT have not been studied, Sun et al. [167] show that any γ-robust algorithm must be $\theta/[\theta/\gamma + (\theta - 1)(1 - 1/\gamma \ln(\theta-1/\gamma-1))]$-consistent in the maximization case, where $\theta = U/L$ is the price bound ratio. While these bounds are not directly comparable, it is notable that the extra structure of SOAD allows it to avoid the exponential robustness of MTS.

Proof overviews

We now give proof sketches of Theorems 5.4.3 and 5.4.4, relegating the full proofs to Section 5.E.

Proof Sketch of Theorem 5.4.3. We separately consider consistency and robustness in turn.

Lemma 5.4.5. ST-CLIP *is* $(1 + \varepsilon)$*-consistent when the advice is correct, i.e.,* $\textsc{Adv}(\mathcal{I}) = \texttt{OPT}(\mathcal{I})$.

For consistency, recall that the constraint enforces that the expected cost of ST-CLIP *thus far* at time j (i.e., before mandatory allocation) satisfies (5.7). Since this constraint holds for all steps before the mandatory allocation, we must resolve the cost *during the mandatory allocation*. We characterize two worst-case scenarios based on whether ST-CLIP has completed *less* (**Case 1**, see (5.20)) or *more* (**Case 2**, see (5.21)) of the deadline constraint compared to $\textsc{Adv}$. In either of these cases, (5.20) and (5.21) show that replacing the "hedging terms" that follow SC_{j-1} and $\textsc{Adv}_{j-1}$ in the constraint with worst-case service and movement costs yields a consistency ratio that is $\leq (1 + \varepsilon)$.

Lemma 5.4.6. ST-CLIP *is* $O(\log n)\gamma^{(\varepsilon)}$*-robust, where* $\gamma^{(\varepsilon)}$ *is defined in* (5.5).

For robustness, we define two cases that characterize "bad" advice, namely "inactive" advice that forces mandatory allocation (**Case 1**, see (5.22)), and "overactive" advice that incurs sub-optimal cost (**Case 2**, see (5.24)). For each of these, we derive bounds on the portion of ST-CLIP's expected solution that is *allowed* to come from the pseudo-cost minimization without violating consistency.

In Case 1, ST-CLIP assumes that $\textsc{Adv}$ can satisfy the constraint at the best possible service cost L, so we derive an *upper bound* describing the maximum utilization achievable via the pseudo-cost minimization before the mandatory allocation (see Proposition 5.E.1). In Case 2, ST-CLIP must follow Adv to avoid violating consistency, even if Adv incurs sub-optimal cost—we derive a *lower bound* on the amount of utilization that ST-CLIP must "spend" while continually satisfying the $(1+\varepsilon)$-consistency constraint (see Proposition 5.E.2). These characterizations enable pseudo-cost proof techniques (e.g., as in Theorem 5.3.3) that show $O(\log n)\gamma^{(\varepsilon)}$-robustness in each case. □

Proof Sketch of Theorem 5.4.4. In Definition 5.E.3, we define a slight variant of the special metric and y-adversary construction from Theorem 5.3.4, denoted by $\mathcal{A}'_y$. Informally, $\mathcal{A}'_y$ presents "good" cost functions at distant points, before eventually presenting just the best service cost functions (i.e., y) at the starting point. We consider two types of advice that each capture consistency and robustness, respectively. In this setting, bad advice completes none of the deadline constraint before the mandatory allocation, while good advice makes the exact decisions that recover $\texttt{OPT}(\mathcal{A}'_y)$.

Using the proof of Theorem 5.3.4, we characterize the cost of a learning-augmented algorithm $\texttt{ALG}$ according to two arbitrary *constraint satisfaction functions* $s(y), t(y) : [L, U] \to [0, 1]$ (see (5.26)). Conditioned on the advice that $\texttt{ALG}$ receives, any α-consistent and γ-robust $\texttt{ALG}$ must satisfy two conditions, where the robustness condition follows from the proof of Theorem 5.3.4 (see (5.25)), and the consistency condition is given by $\gamma \int_{U/\gamma}^{L} \ln\left(\frac{U-u-D+2\tau}{U-U/\gamma-D-2\tau}\right) du + [2D+2\tau]\left[\gamma \ln\left(\frac{U-L-D+2\tau}{U-U/\gamma-D-2\tau}\right)\right] \leq \alpha L - L$ (see (5.27)). Substituting $\alpha := (1+\varepsilon)$ and binding the inequality above yields the result. □

5.5 Generalization to Time-Varying Metrics

Before moving to our case study, we present a generalization of the results in Sections 5.3 and 5.4 to settings with *time-varying metrics*. This is motivated by the applications of SOAD (see Section 5.2) since in practice, the distance between points in the metric (e.g., network delays, transit costs) may not be constant. The extension to time-varying metrics is straightforward, and we present corollaries for both PCM and ST-CLIP after formalizing the extension of SOAD that we consider.

SOAD with time-varying distances (SOAD-T)

In SOAD with time-varying distances, we let $d_t(\cdot, \cdot) : t \in [T]$ denote a *time-varying distance function* between points in X, and we assume that an online algorithm $\texttt{ALG}$ is always able to observe the current distance d_t at time t. Additionally, we redefine

D to be an upper bound on the normalized spatial distance between any two points in X over the entire time horizon T, namely $D = \sup_{t\in[T]} \left(\max_{u,v\in X: u\neq v} \frac{d_t(u,v)}{\min\{c^{(u)},c^{(v)}\}} \right)$. Although distances between the locations of X are time varying, SOAD-T assumes that the temporal switching cost between `ON` and `OFF` states at a single point $u \in X$ is constant (i.e., $\| \cdot \|_{\ell_1(\beta)}$ is *not* time-varying) for simplicity of presentation.

Main results

In the following results, we show that our robust algorithm PCM (see Algorithm 10), and our learning-augmented algorithm ST-CLIP (see Algorithm 11) are both sufficiently flexible to provide guarantees in SOAD-T with minimal changes. Note that the lower bounds in the time-invariant setting still apply to the time-varying setting (e.g., by setting d_t constant for all $t \in [T]$).

First, as a corollary to Theorem 5.3.3, we show that PCM retains its $O(\log n)\eta$ competitive bound in the setting of SOAD-T. We state the result here and give the full proof in Section 5.F.

Corollary 5.5.1. PCM *is* $O(\log n)\eta$*-competitive for* SOAD-T*, where* η *is given by* (5.4).

Furthermore, as a corollary to Theorem 5.4.3, we show that ST-CLIP's consistency-robustness bound also holds for the time-varying setting of SOAD-T when just one term is swapped within the consistency constraint. We state the result here and give the full proof in Section 5.F.

Corollary 5.5.2. *With a minor change to the consistency constraint,* ST-CLIP *is* $(1+\varepsilon)$*-consistent and* $O(\log n)\gamma^{(\varepsilon)}$*-robust for* SOAD-T*, where* $\gamma^{(\varepsilon)}$ *is the solution to* (5.5).

5.6 Case Study: Carbon-aware Workload Management in Datacenters

We end the chapter with a case study applying our algorithms to the problem of carbon-aware workload management on a simulated global network of datacenters.

Experimental setup

We simulate a carbon-aware scheduler that schedules a delay-tolerant batch job on a network of datacenters. We simulate a global network of datacenters based on measurements between Amazon Web Services (AWS) regions. We construct SOAD instances as follows: we generate a job with length J (in hours), an arrival time (rounded to the nearest hour), and a "data size" G, where G gives the amount of data (in GB) to be transferred while migrating the job. The task is to finish the job before the deadline T while minimizing total CO_2 emissions, which are a function of the scheduling decisions and the carbon intensity at each timestep and region.

AWS measurement data. We pick 14 AWS regions [218] based on available carbon data (see Table 5.2 in the appendix). Among these regions, we collect 72,900 pairwise measurements of latency and throughput, compute the mean and variance, and sample a latency matrix. To model migration overhead, we scale the data transferred (and corresponding latency) to match G. These values are scaled by carbon data to define a distance metric on the regions in terms of *CO_2 overhead*. To model network heterogeneity, we set a parameter $\kappa \in [0, 1]$ to adjust the simulated energy of the network. κ is a ratio—if $\kappa = 0.5$, a minute of data transfer from machine(s) in one region to machine(s) in another uses half as much energy as executing at the full allocation (i.e., $x_t^{\mathtt{ON}^{(\cdot)}} = 1$) for one minute.

Carbon data traces. We obtain hourly carbon intensity data for each region, expressed as grams of CO_2 equivalent per kilowatt-hour. In the main body, we consider *average carbon intensity* [171], which gives the average emissions of all electricity generated on a grid at a certain time; this data spans 2020-2022 and includes all regions. In Section 5.B, we also consider *marginal carbon intensity* [219]; this signal is available for 9 regions in 2022, and also includes proprietary forecasts.

We use the latency of moving data between regions to calculate a *CO_2 overhead* for the metric (X, d) (latency × energy × carbon intensity). In most cases, we approximate the network's carbon intensity by the average across regions. When specified, we introduce variation by resampling the carbon intensity of up to $\Upsilon \in [0, n^2]$ links each timestep. We henceforth call Υ a *volatility factor*; resampling assigns a new random carbon intensity (within $[L, U]$) to a link between two regions.

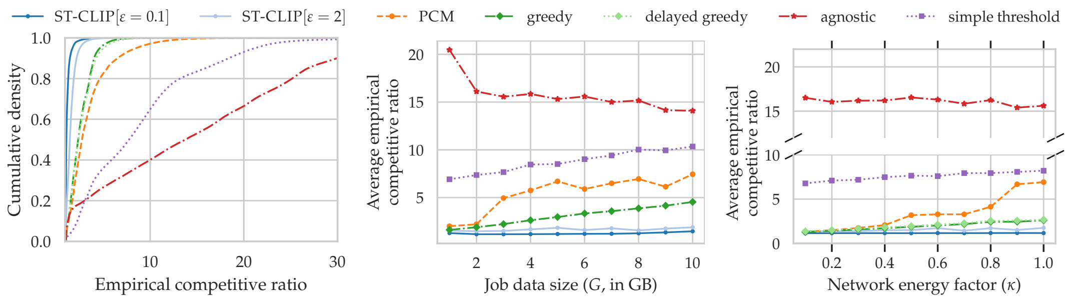
Cloud job traces. We use Google cluster traces [220] that provide a real distribution of job lengths. We normalize this distribution such that the maximum length is 12 hours—each job's length J is drawn from the distribution and rounded up to the next integer, so J falls in the range $\{1, \ldots, 12\}$.

Forecasts. We generate forecasts of the carbon intensity for each location and time. These forecasts are used to solve for a predicted optimal solution that assumes they are correct, which becomes black-box Adv for ST-CLIP. For the average carbon intensity signal, we generate synthetic forecasts by combining true data with random noise.[2] Letting $\text{Carbon}_t^{(u)}$ denote the carbon intensity at datacenter u and time t, our synthetic forecast is given by $\text{Pred}_t^{(u)} = 0.6 \cdot \text{Carbon}_t^{(u)} + 0.4 \cdot \text{Unif}(L, U)$. To test ST-CLIP's robustness, Experiment V directly manipulates Adv. We set an *adversarial factor* $\xi \in [0, 1]$, where $\xi = 0$ implies Adv is correct. We use a solver on true data to obtain two solutions, where one is given a *flipped objective* (i.e., it maximizes carbon emissions). Letting $\{\mathbf{x}_t^\star\}_{t\in[T]}$ denote the decisions of OPT and $\{\breve{\mathbf{x}}_t\}_{t\in[T]}$ denote the decisions of the maximization solution, we have $\text{Adv} := \{(1-\xi)\mathbf{x}_t^\star + \xi \breve{\mathbf{x}}_t\}_{t\in[T]}$. We note that although this is unrealistic in practice, manipulating Adv directly allows us to to quantify the sensitivity of ST-CLIP against all sources of error.

Setup details. We simulate 1,500 jobs for each configuration. Each job's arrival region and arrival time is uniformly random across all active regions and times. Each job's deadline T and data size G are either fixed or drawn from a distribution, and this is specified. To set the parameters L and U, we examine the preceding month of carbon intensities (in all regions) leading up to the arrival time and set L and U according to the minimum and maximum, respectively. We set the following defaults (i.e., unless otherwise specified): The metric covers all 14 regions. Each job's length is drawn from the Google traces as above. The temporal switching coefficient τ is set to 1, the network energy factor κ is set to 0.5, and the volatility factor Υ is set to 0 (i.e., the network is stable).

Benchmark algorithms. We compute the offline optimal solution for each instance using CVXPY [215]. We compare ST-CLIP and PCM against four baselines adapted from literature. The first is a **carbon-agnostic** approach that runs the job whenever it is submitted without migration, simulating the behavior of a non-carbon-aware scheduler. We also consider two *greedy baselines* that use simple decision rules. The first of these is a **greedy** policy that examines the current carbon intensity across all regions at the arrival time, migrates to the "greenest" region (i.e., with lowest carbon intensity), and runs the full job. This captures an observation [221] that one migration to a consistently low-carbon region yields most of the benefits of spatiotemporal shifting. We also consider a policy that we term **delayed greedy**, which examines the *full forecast* across all regions, migrating to start the job at the "best region and time" (i.e., slot with lowest predicted carbon anywhere). If there is not enough time to finish the job after the identified slot, it is scheduled to start as close to it as possible. The final baseline is a **simple threshold**-based approach

[2]We use an open-source ML model that provides carbon intensity forecasts for U.S. regions [50] to tune the magnitude of random noise such that Adv's empirical competitive ratio is slightly worse than an Adv that uses the ML forecasts.



(a) CDFs of competitive ratios for each algorithm, across all average carbon intensity experiments.

(b) Average empirical competitive ratios for varying *job data size* G, with $T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.

(c) Average empirical competitive ratios for varying *energy factor* κ, with $G = 4, T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.

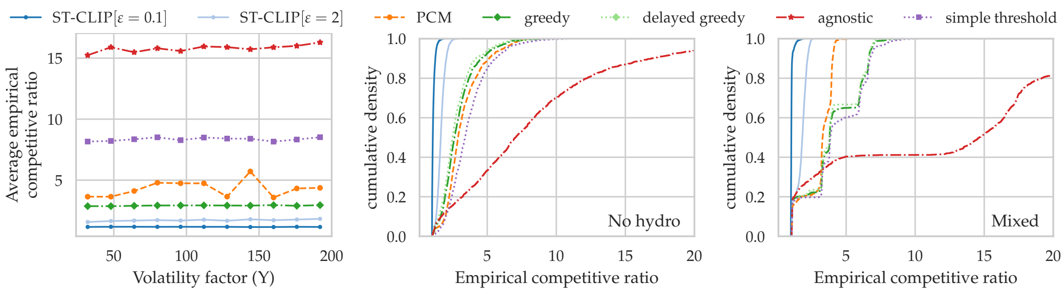
Figure 5.2: Experimental results comparing ST-CLIP against several baseline algorithms.

from temporal shifting literature [192, 222]; it sets a threshold $\sqrt{UL}$, based on prior work in online search [78]. At each timestep, it runs the job in the best region whose carbon intensity is $\leq \sqrt{UL}$, without considering migration overheads. If no regions are $\leq \sqrt{UL}$ at a particular time, the job is checkpointed in place, and a *mandatory allocation* happens when approaching the deadline if the job is not finished.

Experimental results

We highlight several experiments here, referring to Section 5.B for the extended set. A summary is given in Figure 5.2a, where we plot a cumulative distribution function (CDF) of the empirical competitive ratio for all tested algorithms in Expts. I-IV and VI-VIII. Given imperfect advice, ST-CLIP[$\varepsilon = 2$] significantly outperforms the baselines, improving on greedy, delayed greedy, simple threshold, and carbon-agnostic by averages of 32.1%, 33.5%, 79.4%, and 88.7%, respectively. In Expts. I-III, each job's deadline is a random integer between 12 and 48 (denoted by $T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$). In these experiments, both greedy policies outperform our robust algorithm, PCM. This result aligns with prior findings [221]; since these experiments consider all 14 regions, there are consistent low-carbon grids in the mix that give an advantage to the greedy policies. In Expt. IV, we examine this further, showing that the performance of greedy policies can degrade in realistic situations.

Experiment I: *Effect of job data size G.* In Figure 5.2b, we plot the average empirical competitive ratio for job data sizes $G \in \{1, \ldots, 10\}$. Recall that parameter D depends on the diameter of the metric space (i.e., the worst migration overhead between regions); as G increases, this maximum overhead grows. As predicted by the theoretical bounds, PCM's performance degrades as G grows; we observe



(a) Average empirical competitive ratios for varying *volatility factor* Υ. $G = 4$, $T \sim \mathrm{Unif}_{\mathbb{Z}}(12, 48)$.

(b) CDFs of empirical competitive ratios for each tested algorithm on the "no hydro" region subset.

(c) CDFs of empirical competitive ratios for each tested algorithm on the "mixed" region subset.

Figure 5.3: Further experimental results comparing ST-CLIP against several baseline algorithms.

the same effect for the greedy policies and simple threshold. Since it can leverage advice, ST-CLIP maintains consistent performance for many settings of G.

Experiment II: *Effect of network energy scale* κ. Figure 5.2c plots the average empirical competitive ratio for $\kappa \in [0.1, 1]$, fixing $G = 4$. As in Expt. I, κ affects the parameter D—thus, the performance of PCM degrades slightly as κ grows. When κ is small, greedy policies perform nearly as well as ST-CLIP, though they degrade as κ increases; ST-CLIP's usage of advice yields consistent performance.

Experiment III: *Effect of volatility factor* Υ. In Figure 5.3a, we plot the average empirical competitive ratio for $\Upsilon \in [28, 196]$, fixing $G = 4$. Aligning with the theoretical results (Corollary 5.5.1 & 5.5.2), we find that PCM and ST-CLIP's performance is robust to this volatility. Both of the greedy policies do not consider the migration overhead and only migrate once, so their performance is consistent.

Experiment IV: *Effect of electric grids and datacenter availability.* Greedy policies do well in Expts. I-III, where some regions have consistently low-carbon grids.[3] In practice, a greedy policy may face obstacles if it is unable to migrate to low-carbon regions. For instance, such regions might reach capacity, *removing them* as migration options. Jobs may also be *restricted* from leaving a region due to regulations [223]. Further, consistently low-carbon grids often leverage hydroelectric or nuclear sources that are difficult to build at scale compared to cheaper renewables [224]. This is important because it suggests future grids will moreso resemble those marked by renewable intermittency.

[3] Figure 5.5 in the appendix plots a sample of carbon intensity data for all 14 regions to motivate this visually.

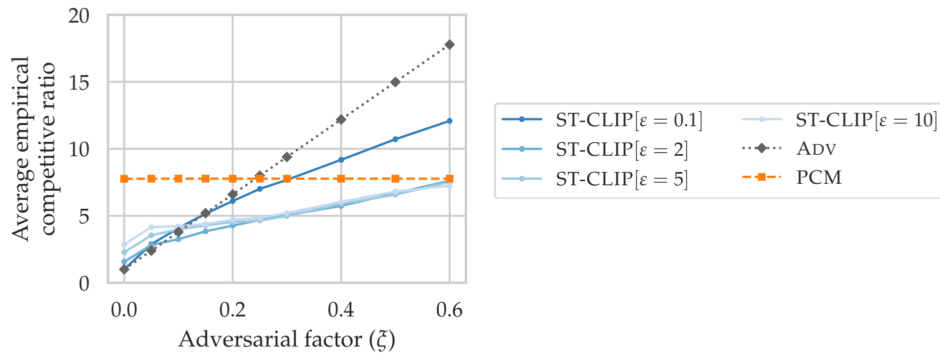


Figure 5.4: Average empirical competitive ratio for varying *adversarial factor* ξ, with $G = 4, T = 12$. $\xi \to 0$ implies that ADV $\to$ OPT, and as ξ grows, ADV degrades.

In Figure 5.3b and 5.3c, we present results where the metric is a *subset* of the 14 regions, giving results on more subsets in Section 5.B. By considering these subsets, we approximate issues discussed above (e.g., datacenter congestion, grid characteristics). Figure 5.3b considers a "no hydroelectric" subset that omits Sweden and Quebec. Under this subset, PCM closes the gap with the greedy policies, with an average competitive ratio that is within 4.24% of both. Figure 5.3c considers a smaller subset of 5 regions: South Korea, Virginia, Sydney, Quebec, and France. On this mixed set of grids, PCM outperforms greedy and delayed greedy by 30.91% and 28.79%, respectively. The results above highlight that situations do arise where greedy policies perform worse than both ST-CLIP and PCM. However, such situations are not a majority—out of the 14 random subsets that we tested, PCM outperformed the greedy policy in *four* subsets, including the "mixed" (Figure 5.3c) and "mixed 2" (Figure 5.7c) subsets. PCM is relatively conservative in its decisions, being optimized for worst-case (adversarial) inputs; this advantages greedy policies on the majority of instances that do not benefit from the "worst-case hedging" behavior that PCM exhibits.

Experiment V: *Effect of bad black-box advice ξ.* Figure 5.4 plots the effect of bad black-box advice on ST-CLIP's performance. We test values of $\xi \in [0, 0.6]$, generating ADV according to the technique discussed in Section 5.6. ST-CLIP is initialized with $\varepsilon \in [0.1, 2, 5, 10]$, where $\varepsilon \to 0$ implies that it follows ADV more closely. We also plot the empirical competitive ratio of PCM and ADV as opposing baselines. We find that ST-CLIP nearly matches ADV when it is correct, while degrading more gracefully as ADV's cost increases. This result shows that ST-CLIP is empirically robust to even adversarial black-box advice.

5.7 Conclusion

Motivated by sustainability applications, we introduce and study spatiotemporal online allocation with deadlines (SOAD), the first online problem that combines general metrics with deadline constraints, bridging the gap between existing literature on metrical task systems and online search. Our main results present PCM as a competitive algorithm for SOAD, and ST-CLIP, a learning-augmented algorithm that achieves a near-optimal robustness-consistency tradeoff. We evaluate our proposed algorithms in a case study of carbon-aware workload management in datacenters. A number of questions remain for future work, including natural extensions motivated by applications. For computing applications, SOAD may be extended to model resource contention and/or delayed access to resources, particularly after moving the allocation to a new point. Similarly, an extension to model *multiple* workloads with different deadlines would be natural (e.g., scheduling multiple batch jobs, dynamic job arrivals/departures).

Our theoretical results contend with substantial generality (i.e., in the metric); it would be interesting to explore whether improved results can be obtained under a more structured setting.

Appendix

In these appendix sections, we describe additional example applications of the SOAD framework, we provide additional experimental results on our proposed algorithms, and we give full proofs of the theoretical results in the main body of the chapter.

5.A Deferred Examples

In this section, we detail two more examples of applications that motivate the SOAD problem introduced in the main body, picking up from Section 5.2.

Carbon-aware or cost-aware autonomous electric vehicle charging. Consider an autonomous electric vehicle taxi (AEV) servicing a city [225] with multiple charging stations. Suppose that by the end of a day (i.e., deadline T), the AEV must replenish the charge that it will have used throughout the day. Service costs $f_t^{(u)}$ can represent either the carbon emissions of charging at location u during time slot t, or the charging cost plus opportunity cost of charging at location u during time slot t. We note that even within a single city, the *locational marginal emissions* (i.e., the carbon intensity of electricity at a specific location) may vary significantly [226], and charging prices can be similarly variable based on, e.g., time-of-use and/or zonal energy pricing [227]. The metric space (X, d) and the spatial movement

Table 5.1: A summary of key notation.

Notation	Description
$t \in [T]$	Timestep index
$\mathcal{X}$	Feasible set for vector allocation decisions
$\mathbf{x}_t \in \mathcal{X}$	Allocation decision at time t
$f_t(\cdot) : \mathcal{X} \to \mathbb{R}$	(*Online input*) Service cost function revealed to the player at time t
$c(\cdot) : \mathcal{X} \to [0,1]$	Constraint function; describes the fraction satisfied by an allocation
$d(u,v) : u,v \in X \to \mathbb{R}$	Spatial distance in the metric (X, d)
$\lVert\mathbf{x} - \mathbf{x}'\rVert_{\ell_1(\beta)} : \mathbf{x}, \mathbf{x}' \in \mathcal{X} \to \mathbb{R}$	Switching costs between ON and OFF allocations
$g(\cdot, \cdot) \coloneqq d(\cdot, \cdot) + \lVert \cdot \rVert_{\ell_1(\beta)}$	Combined movement & switching cost between points and allocations
$u \in X$	Point u in an n-point metric space (X, d)
$\mathtt{ON}^{(u)}, \mathtt{OFF}^{(u)}$	ON state and OFF state at point u, respectively
$x_t^{\mathtt{ON}^{(u)}}, x_t^{\mathtt{OFF}^{(u)}} \in [0,1]$	Fractional allocations to ON / OFF states at point u at time t
$c^{(u)} \in (0,1]$	Throughput coefficient; describes constraint satisfied by $x^{\mathtt{ON}^{(u)}} = 1$
$f_t^{(u)} \in [c^{(u)}L, c^{(u)}U]$	Service cost coefficient at $\mathtt{ON}^{(u)}$ & time t; proportional to $L > 0$ and U.
$\beta^{(u)} > 0$	Switching coefficient; describes switching cost between $\mathtt{ON}^{(u)} \leftrightarrow \mathtt{OFF}^{(u)}$
$\tau : \beta^{(u)} \leq \tau c^{(u)} \ \forall u \in X$	Upper bound on normalized switching coefficient
$D = \sup_{u,v \in X} \frac{d(u,v)}{\min\{c^{(u)}, c^{(v)}\}}$	Upper bound on normalized spatial distance between any two points
$\mathcal{S}$	Discrete set of all ON and OFF states
$\Delta_{\mathcal{S}}$	Probability measure over $\mathcal{S}$
$\mathbf{p}_t \in \Delta_{\mathcal{S}}$	Probability distribution (& corresponding random allocation) at time t
$\mathbb{W}_1(\mathbf{p}, \mathbf{p}') : \mathbf{p}, \mathbf{p}' \in \Delta_{\mathcal{S}} \to \mathbb{R}$	Optimal transport distance between distributions (in terms of $g(\cdot, \cdot)$)
$\delta_s \in \Delta_{\mathcal{S}}$	Dirac measure supported at $\mathtt{OFF}^{(s)}$
$\mathcal{T} = (V, E)$	Hierarchically separated tree (HST) constructed by Definition 5.2.1
$K \subset \mathbb{R}^{\lvert V \rvert}$	Vector space corresponding to $\mathcal{T}$ (see Definition 5.2.2)
$\mathbf{k}_t \in K$	Vector decision (& corresponding prob. distribution) at time t
$\Phi \in \mathbb{R}^{\lvert V \rvert \times 2n}$	Linear map such that $\Phi\mathbf{p} \in K$ and $\Phi^{-1}\mathbf{k} \in \Delta_{\mathcal{S}}$
$\lVert \cdot \rVert_{\ell_1(\mathbf{w})} : K \to \mathbb{R}$	Weighted ℓ_1 norm that recovers optimal transport distances in $\mathcal{T}$
$\overline{f}_t(\mathbf{k}) = f_t(\Phi^{-1}\mathbf{k})$ $\overline{c}(\mathbf{k}) = c(\Phi^{-1}\mathbf{k})$	Notation shorthand for functions defined on vector space K
$z^{(t)} \in [0,1]$	Utilization; fraction of constraint satisfied in expectation up to time t
$\textsc{Adv}(\mathcal{I}) \coloneqq \{\mathbf{a}_t \in \Delta_{\mathcal{S}}\}_{t \in [T]}$	Black-box advice provided to ST-CLIP (see Definition 5.4.1)
$A^{(t)} \in [0,1]$	$\textsc{Adv}$ utilization; fraction of constraint satisfied by $c(\mathbf{a}_1) + \cdots + c(\mathbf{a}_t)$

cost $d(u_{t-1}, u_t)$ capture either the carbon overhead (in terms of "wasted" electricity) or the opportunity/time cost of moving to a different location for charging. The temporal switching cost $\lVert\mathbf{x}_t - \mathbf{x}_{t-1}\rVert_{\ell_1(\boldsymbol{\beta})}$ captures the small overhead of stopping or restarting charging at a single location, due to extra energy or time spent connecting or disconnecting from the charger at location u. Finally, the constraint function $c(\mathbf{x}_t)$ captures how much charge is delivered during time t according to decision $\mathbf{x}_t$, where an $\mathbf{x}_t$ that places a full allocation in the $\mathtt{OFF}^{(u)}$ state indicates that the AEV is serving customers (i.e., not charging). Since the AEV may move around in the city while serving customers, $\mathbf{x}_t$ should be updated exogenously to reflect the true state. We note that SOAD may not capture the case where the distance between charging stations is large (i.e., moving to a different location incurs substantial discharge) or the case where the AEV is at risk of fully discharging the battery before time T, which would require immediate charging.

Table 5.2: Summary of CO_2 data sets for each tested AWS region in our case study experiments, including the minimum and maximum carbon intensities, duration, granularity, and data availability.

Location	AWS Region	Average Carbon Intensity *(in gCO_2eq/kWh)* [171]			Marginal Carbon Intensity *(in gCO_2eq/kWh)* [219]		
		Duration	Min.	Max.	Duration	Min.	Max.
Virginia, U.S.	`us-east-1`	01/01/2020 - 12/31/2022 Hourly granularity 26,304 data points	293	567	01/01/2022 - 12/31/2022 Hourly granularity 8,760 data points	48	1436
California, U.S.	`us-west-1`		83	451		67	1100
Oregon, U.S.	`us-west-2`		42	682		427	2000
Quebec, Canada	`ca-central-1`		26	109		887	1123
London, U.K.	`eu-west-2`		56	403		706	1082
France	`eu-west-3`		18	199		549	1099
Sweden	`eu-north-1`		12	59		438	2556
Germany	`eu-central-1`		130	765		11	1877
Sydney, Australia	`ap-southeast-2`		267	817		12	1950
Brazil	`sa-east-1`		46	292	*Data not available*		
South Africa	`af-south-1`		586	785			
Israel	`il-central-1`		514	589			
Hyderabad, India	`ap-south-2`		552	758			
South Korea	`ap-northeast-2`		453	503			

Allocating tasks to volunteers. Consider a non-profit that has a task to complete before some short-term deadline T, with several locations and scheduled time slots for volunteer efforts (e.g., stores, community centers) throughout a region. Using platforms such as VolunteerMatch [228], volunteers can signal their interests in tasks (e.g., via a ranking) and availability—in assigning this task, the non-profit may want to maximize the engagement of their assigned volunteer(s). Service costs $f_t^{(u)}$ can represent the aggregate rankings of the volunteers present at location u during time slot t, where a lower number means that they are more interested in a given task. The metric space (X, d) and the spatial movement cost $d(u_{t-1}, u_t)$ can capture the cost of, e.g., moving supplies that must be present at the location to work on the task. The temporal switching cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\beta)}$ may capture the cost of, e.g., setting up or breaking down the setup required to work on the task at a given location u. Finally, the constraint function $c(\mathbf{x}_t)$ captures how much of the task can be completed during time t according to assignment decision $\mathbf{x}_t$. We note that the fractional allocation to `ON` / `OFF` states specified by SOAD may not be useful in this setting because, e.g., groups of volunteers may not be fractionally divisible.

5.B Supplemental Experiments

In this section, we present additional results and figures to complement those in the main body. In the first few results, we present additional experiments manipulating parameters using the *average carbon intensity signal*. Then, in Section 5.B, we present a supplemental slate of experiments using the *marginal carbon intensity signal* obtained from WattTime [219].

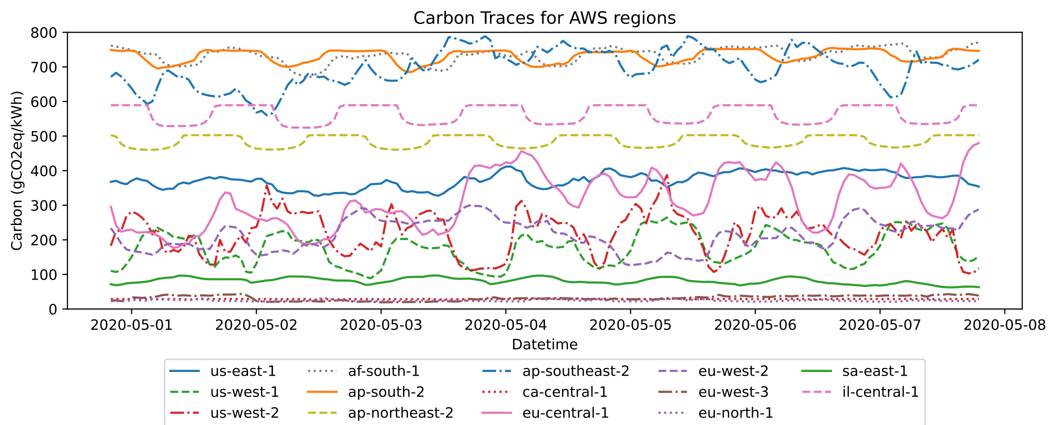
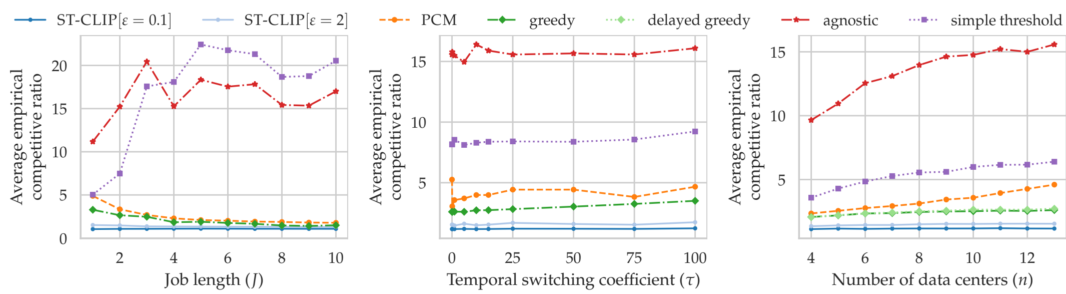


Figure 5.5: Average carbon intensity traces [171] for all 14 AWS regions, over a week-long period in 2020.



(a) Average empirical competitive ratios for varying *job length* J, with $G = 4, T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.

(b) Average empirical competitive ratios for varying *temporal switching coefficient* τ, with $G = 4, T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.

(c) Average empirical competitive ratios for varying *number of datacenters* n, with $G = 4, T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.

Figure 5.6: Supplementary experimental results comparing ST-CLIP against several baseline algorithms in varied settings.

Experiment VI: *Effect of job length* J. Figure 5.6a plots the average empirical competitive ratio for fixed job lengths $J \in \{1, \dots, 10\}$, where J is the length of the job in hours. In this experiment, we fix each job's data size to $G = 4$. As J increases, the empirical competitive ratio of PCM improves, and ST-CLIP remains consistent. We note that the simple threshold technique is able to achieve good performance in the case when $J = 1$; intuitively, since this simple threshold is agnostic to the switching overhead, its performance degrades when it uses more than one opportunity to migrate between regions (i.e., when the job takes more than one time slot).

As is characteristic of realistic cloud traces, the Google cluster traces we use in Experiments I-V are mostly composed of shorter jobs between 1 and 2 hours long.

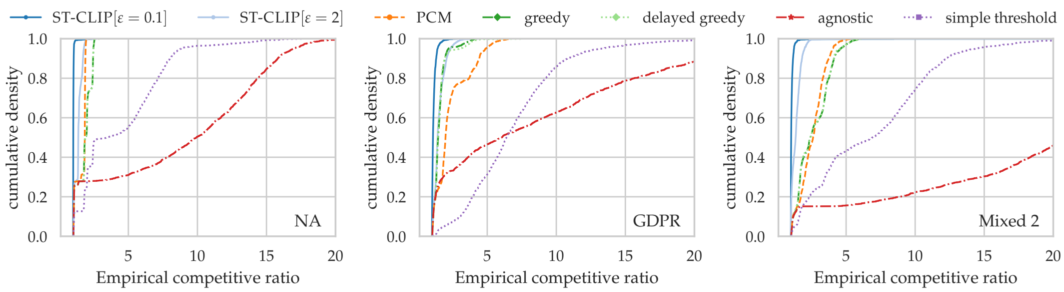
These results for fixed job lengths highlight that ST-CLIP and PCM do even better when given lengthy jobs—such jobs are less frequent but take up a disproportionate amount of compute cycles (and thus contribute disproportionately to the carbon footprint of) a typical datacenter.

Experiment VII: *Effect of temporal switching coefficient τ.*
Recall that increasing or decreasing τ simulates jobs that have more or less time-consuming checkpoint and resume overheads, respectively. In Figure 5.6b, we plot the average empirical competitive ratio for varying $\tau \in \{0, \dots, 100\}$. In this experiment, we fix $J = 4$ and $G = 4$. Compared to varying G, τ has a smaller impact across the board, although as predicted by the theoretical bounds, the performance of PCM degrades slightly as τ grows, and the greedy policies are similarly affected.

Experiment VIII: *Effect of number of datacenters n.*
Building off of the idea in Experiment IV, in Figure 5.6c we plot the average empirical competitive ratio for varying $n \in [4, 13]$, where a random subset (of size n) is sampled from the base 14 regions for each batch job instance. For each job, we fix $G = 4$, a deadline T that is a random integer between 12 and 48, and use the average carbon intensity signal. We find that most algorithms' performance degrades as the size of the subset increases. This is likely because the expected range of carbon intensities expands as more diverse electric grids are included in the subset. As in previous experiments, ST-CLIP's performance with black-box advice is consistent as n increases.

Experiment IV (continued): *Effect of electric grids and datacenter availability.*
Continuing from Experiment IV in the main body, we present results where the metric space is constructed on four additional *subsets* of the 14 regions. By considering these region subsets, we approximate issues of datacenter availability, and electric grid characteristics that might face a deployment in practice.

Figure 5.7a considers a North American subset of 4 regions: California, Oregon, Virginia, and Quebec. Under this subset, PCM slightly outperforms greedy and delayed greedy by 8.31% and 8.24%, respectively. Figure 5.7b considers an "EU / GDPR" subset that includes 4 regions: France, Germany, London, and Sweden. On this subset, with a large proportion of consistent low-carbon grids (i.e., both Sweden and France), the greedy and delayed greedy policies outperform PCM by 21.54% and 20.25%, respectively. These results highlight a "best case" situation where the



(a) CDFs of empirical competitive ratios for each tested algorithm on the North America subset.

(b) CDFs of empirical competitive ratios for each tested algorithm on the EU (GDPR) region subset.

(c) CDFs of empirical competitive ratios for each tested algorithm on the second "mixed" subset.

Figure 5.7: Supplementary experimental results comparing ST-CLIP against several baseline algorithms on various subsets of regions.

greedy policies are able to outperform PCM and nearly match the performance of ST-CLIP. Figure 5.7c considers a second "mixed" subset of 7 regions: California, South Korea, Germany, Hyderabad, Israel, Sweden, and South Africa. On this subset, with a geographically distributed mix of high and low-carbon grids, PCM outperforms greedy and delayed greedy by 3.74% and 4.73%, respectively.

We briefly note that since the greedy policies exhibit fairly good performance across many of these experiments, for the intended application of carbon-aware workload management in datacenters it may be worthwhile to evaluate the performance of ST-CLIP when given black-box advice ADV that simply encodes the decisions of the greedy policy. Since the black-box advice model can accommodate any arbitrary sequence of decisions, including heuristics, such a composition may achieve a favorable tradeoff between average-case performance and worst-case guarantees if, e.g., machine-learned forecasts are not available.

Experiment IX: *Runtime (wall clock) overhead measurements.*
In this experiment, we measure the wall clock runtime of each tested algorithm. Our experiment implementations are in Python, using NumPy [229], SciPy [230], and CVXPY [215]—we use the `time.perf_counter()` module in Python to calculate the total runtime (in milliseconds) for each algorithm on each instance, and report this value *normalized by the deadline* to give the *per-slot* (i.e., hourly) overhead of each algorithm.

Figure 5.8 reports these measurements for batch jobs with deadlines $T \in \{6, \ldots, 48\}$, fixed $G = 4$, and job lengths from the Google trace are truncated to $T/2$ if necessary.

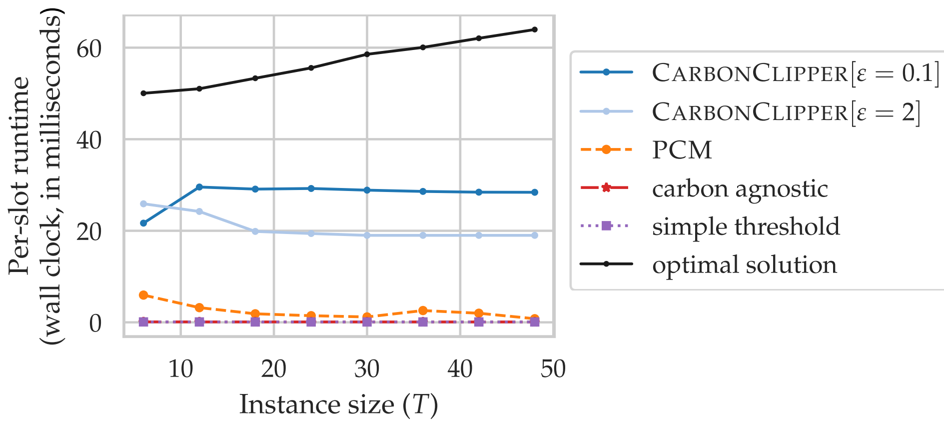


Figure 5.8: Average per-slot wall clock runtime for *instance sizes (i.e., deadlines)* $T \in \{6, \ldots, 48\}$, with $G = 4$.

For this experiment, we run each algorithm and each instance in a single thread on a MacBook Pro with M1 Pro processor and 32 GB of RAM.

We find that the average per-slot (i.e., once per hour) runtimes of PCM and ST-CLIP are 2.3 milliseconds and 24.3 milliseconds, respectively—this reflects the relative complexity of the optimization problems being solved in each, and the runtime is steady in the size of the instance, as is expected from an online algorithm. The optimal solution takes, on average, 173.8 milliseconds per slot to solve, although note that it finds the solution for all time slots at once. As the size of the instance (T) grows, this per-slot time slightly increases. Intuitively, the decision rule-based algorithms (carbon agnostic, simple threshold, and the greedy policies (not included in the plot)) have the lowest, and functionally negligible, runtime. This last result suggests that one effective way to reduce the impact of PCM and ST-CLIP's runtime overhead would be to develop approximations that avoid computing the exact solution to the minimization problem. However, for carbon intensity signals that are updated every 5 minutes to one hour, the overhead of PCM and ST-CLIP is likely reasonable in practice.

Marginal Carbon Intensity

In contrast to average carbon intensity, the marginal carbon intensity signal calculates the emissions of the generator(s) that are responding to changes in load on a grid at a certain time. From WattTime [219], we obtain data for 9 of the 14 regions we consider, spanning all of 2022. This data also includes carbon intensity forecasts published by WattTime, and we use these forecasts directly instead of generating

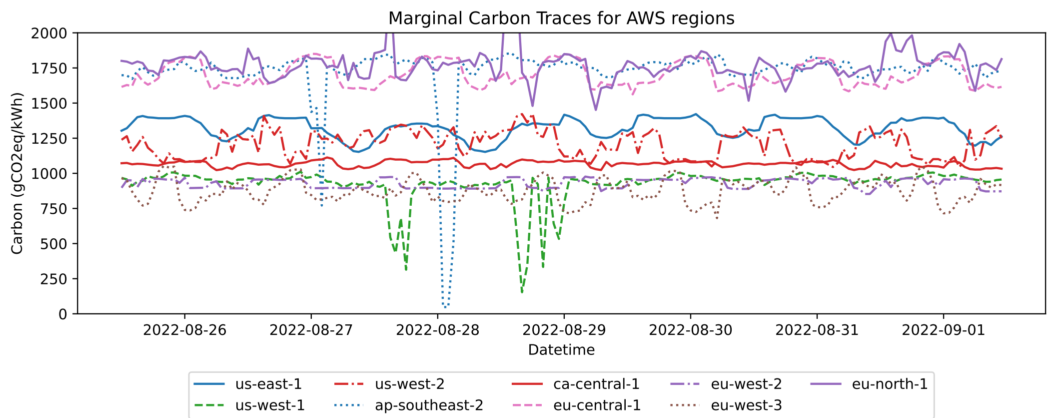


Figure 5.9: Marginal carbon intensity traces [171] for 9 AWS regions, over a week-long period in 2022.

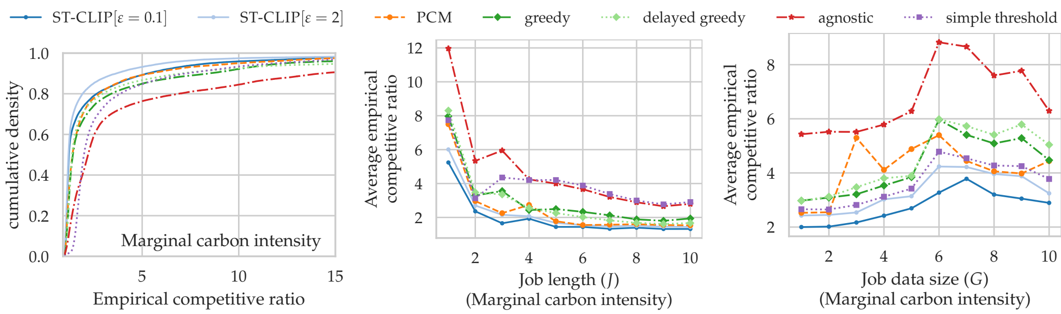
synthetic forecasts as in Section 5.6. In Figure 5.9, we plot a one-week sample of carbon intensity data to motivate this visually.

A high-level summary of these experiments is given in Figure 5.10a—this plot gives a cumulative distribution function (CDF) of the empirical competitive ratios for all tested algorithms, aggregating over all experiments that use the marginal carbon intensity signal. In these experiments, we consider all of the 9 regions for which the marginal data is available, and each job's deadline T is a random integer between 12 and 48 (henceforth denoted by $T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$).

In these experiments, we observe differences that are likely attributable to the characteristic behavior of the marginal carbon intensity signal, which generally represents the high emissions rate of a quick-to-respond generator (e.g., a gas turbine) unless the supply of renewables on the grid exceeds the current demand. However, the relative ordering of performance has been largely preserved.

Interestingly, we note that the real forecasts are worse in the marginal setting compared to the average carbon intensity signal, again likely because of the characteristics of marginal carbon. Rather than predicting, e.g., the diurnal patterns of an average signal, predicting marginal carbon requires a model to pick out specific time slots where curtailment is expected to occur.

Despite these challenges of forecasting the marginal signal, ST-CLIP with $\varepsilon = 2$ outperforms the baselines in both average and worst-case performance, improving on the closest greedy policy by an average of 12.04%, and outperforming delayed greedy, simple threshold, and carbon-agnostic by averages of 11.02%, 37.12%, and 85.67%, respectively.



(a) CDFs of competitive ratios for each algorithm, across all marginal carbon intensity experiments.

(b) Average competitive ratios using marginal carbon data, with varying *job length* J and $G = 4$, $T \sim \mathrm{Unif}_{\mathbb{Z}}(12, 48)$.

(c) Average competitive ratios using marginal carbon data, with varying *job data size* G and $T \sim \mathrm{Unif}_{\mathbb{Z}}(12, 48)$.

Figure 5.10: Supplementary experimental results comparing ST-CLIP against several baseline algorithms when using marginal carbon intensity.

Marginal Experiment I: *Effect of job length J.*

In Figure 5.10b, we plot the average competitive ratio for different job lengths $J \in \{1, \ldots, 10\}$. Each job has $G = 4$ and an integer deadline T randomly sampled between 12 and 48. Compared to the greedy policies, our robust baseline PCM performs favorably in these experiments, which is likely due to a combination of differences in the marginal setting and the less performant forecasts. Similar to the average carbon setting, the simple threshold technique performs well when the job length is short, but performance suffers when it has more opportunities for migration.

Marginal Experiment II: *Effect of job data size G.*

In Figure 5.10c, we plot the average empirical competitive ratio for different job data sizes $G \in \{1, \ldots, 10\}$ using the marginal carbon signal. As predicted by the theoretical bounds, and observed in the main body for the average carbon experiments, the performance of PCM degrades as G grows—we observe the same effect for the greedy policies and the simple threshold algorithm. The performance of ST-CLIP also grows, which suggests that larger migration overhead does have an impact when the advice suffers from a lack of precision due to the more challenging setting for forecasts posed by marginal carbon intensity.

5.C Proofs for Section 5.2

In the following, we prove Theorem 5.2.3, which shows that the expected cost of any randomized SOAD decision $\mathbf{p}_t \in \Delta_{\mathcal{S}}$ is equivalent to that of a decision which chooses a point in X probabilistically according to the distribution of $\mathbf{p}_t$ and then interprets the ON / OFF probabilities at that point as deterministic allocations in $\mathcal{X}$.

Proof of Theorem 5.2.3. Suppose that $\mathbf{p}_t, \mathbf{p}_{t-1}$ are probability distributions over the randomized state space $\Delta_{\mathcal{S}}$, and let $\mathbb{E}\left[\text{Cost}(\mathbf{p}_t, \mathbf{p}_{t-1})\right]$ denote the expected cost of decision $\mathbf{p}_t$. This cost is defined as follows:

$$\mathbb{E}\left[\text{Cost}(\mathbf{p}_t, \mathbf{p}_{t-1})\right] = \mathbb{E}\left[f_t(\mathbf{p}_t) + g(\mathbf{p}_t, \mathbf{p}_{t-1})\right].$$

Recall that because the cost function f_t is linear and separable, the expectation can be written as:

$$\mathbb{E}\left[f_t(\mathbf{p}_t)\right] = \sum_{u \in X} f_t^{(u)} p_t^{\mathtt{ON}^{(u)}}.$$

For any $\mathbf{p}_t \in \Delta_{\mathcal{S}}$, let $\mathbf{r}_t := \{r_t^{(u)} \leftarrow p_t^{\mathtt{ON}^{(u)}} + p_t^{\mathtt{OFF}^{(u)}} : u \in X\} \in \Delta_X$, i.e., a vector that aggregates the total probabilities across the states space $\mathcal{S}$ at each point of X.

We note that by disaggregating the spatial and temporal switching costs, we have that the expectation $\mathbb{E}\left[g(\mathbf{p}_t, \mathbf{p}_{t-1})\right]$ can be written in terms of the Wasserstein-1 distance with respect to the underlying metric X and a linear temporal term that depends on the probability assigned to the OFF state. This is the case because the optimal transport plan $\mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1})$ must always involve first moving probability mass to/from the OFF and ON states at each point of X, and then within the spatial metric—this follows since SOAD defines that movement within the spatial metric X can only be made between ON states—i.e., a player moving from $\mathtt{OFF}^{(u)}$ to $\mathtt{OFF}^{(v)}$ must first traverse to $\mathtt{ON}^{(u)}$, then through the metric X, and finally through $\mathtt{ON}^{(v)}$.

$$\mathbb{E}\left[g(\mathbf{p}_t, \mathbf{p}_{t-1})\right] = \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \sum_{u \in X} \beta^{(u)} |p_t^{\mathtt{OFF}^{(u)}} - p_{t-1}^{\mathtt{OFF}^{(u)}}|.$$

Thus, the expected cost of $\mathbf{p}_t$ can be written as:

$$\mathbb{E}\left[\text{Cost}(\mathbf{p}_t, \mathbf{p}_{t-1})\right] = \sum_{u \in X} f_t^{(u)} p_t^{\mathtt{ON}^{(u)}} + \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \sum_{u \in X} \beta^{(u)} |p_t^{\mathtt{OFF}^{(u)}} - p_{t-1}^{\mathtt{OFF}^{(u)}}|.$$

In the mixed probabilistic/deterministic setting, the true allocation to $\mathtt{ON}^{(u)}$ (denoted by $\tilde{p}_t^{\mathtt{ON}^{(u)}}$) for any point $u \in X$ is defined as $\tilde{p}_t^{\mathtt{ON}^{(u)}} = p_t^{\mathtt{ON}^{(u)}} / r_t^{(u)}$ (conversely, we have $\tilde{p}_t^{\mathtt{OFF}^{(u)}} = p_t^{\mathtt{OFF}^{(u)}} / r_t^{(u)}$). Letting $\mathbb{L}_t^{(u)} \in \{0, 1\}$ denote an indicator variable that encodes the player's location (i.e., point) at time t (0 if player is not at u, 1 if player is at u), the expected cost of $\tilde{\mathbf{p}}_t$ can be written as follows:

$$\begin{aligned}
&\mathbb{E}\left[\text{Cost}(\tilde{\mathbf{p}}_t, \tilde{\mathbf{p}}_{t-1})\right] \\
&= \mathbb{E}\left[\sum_{u \in X} \mathbb{L}_t^{(u)} f_t^{(u)} \tilde{p}_t^{\mathtt{ON}^{(u)}}\right] + \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \mathbb{E}\left[\sum_{u \in X} \left|\mathbb{L}_t^{(u)} \beta^{(u)} \tilde{p}_t^{\mathtt{OFF}^{(u)}} - \mathbb{L}_{t-1}^{(u)} \beta^{(u)} \tilde{p}_{t-1}^{\mathtt{OFF}^{(u)}}\right|\right].
\end{aligned}$$

Noting that $\mathbb{E}\left[\mathbb{L}_t^{(u)}\right] = r_t^{(u)}$ by linearity of expectation we have:

$$\begin{aligned}&\mathbb{E}\left[\mathrm{Cost}(\tilde{\mathbf{p}}_t, \tilde{\mathbf{p}}_{t-1})\right] \\ &= \sum_{u \in X} r_t^{(u)} f_t^{(u)} \tilde{p}_t^{\texttt{ON}^{(u)}} + \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \sum_{u \in X} |r_t^{(u)} \beta^{(u)} \tilde{p}_t^{\texttt{OFF}^{(u)}} - r_{t-1}^{(u)} \beta^{(u)} \tilde{p}_{t-1}^{\texttt{OFF}^{(u)}}|.\end{aligned}$$

Furthermore, recalling the definitions of $\tilde{p}_t^{\texttt{ON}^{(u)}}$ and $\tilde{p}_t^{\texttt{OFF}^{(u)}}$, we have the following:

$$\mathbb{E}\left[\mathrm{Cost}(\tilde{\mathbf{p}}_t, \tilde{\mathbf{p}}_{t-1})\right] = \sum_{u \in X} f_t^{(u)} p_t^{\texttt{ON}^{(u)}} + \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \sum_{u \in X} \beta^{(u)} |p_t^{\texttt{OFF}^{(u)}} - p_{t-1}^{\texttt{OFF}^{(u)}}|.$$

Recalling that $\mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) = \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \sum_{u \in X} \beta^{(u)} |p_t^{\texttt{OFF}^{(u)}} - p_{t-1}^{\texttt{OFF}^{(u)}}|$ by the structure of the spatial and temporal switching costs completes the proof, since $\mathbb{E}\left[\mathrm{Cost}(\mathbf{p}_t, \mathbf{p}_{t-1})\right] = \mathbb{E}\left[\mathrm{Cost}(\tilde{\mathbf{p}}_t, \tilde{\mathbf{p}}_{t-1})\right]$, and thus the expected cost is equivalent if a point (location) is first chosen probabilistically and the `ON` / `OFF` probabilities at that point are then interpreted as deterministic (fractional) allocations.

□

5.D Proofs for Section 5.3

Convexity of the pseudo-cost minimization problem in PCM

In this section, we prove Theorem 5.3.2, which states that the pseudo-cost minimization problem central to the design of PCM is a convex minimization problem, implying that it can be solved efficiently.

For convenience, let $h_t(\mathbf{k}) : t \in [T]$ represent the pseudo-cost minimization problem's objective for a single arbitrary timestep:

$$h_t(\mathbf{k}) = f_t(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)} + \overline{c}(\mathbf{k})} \psi(u) du. \tag{5.9}$$

Proof of Theorem 5.3.2. We prove the statement by contradiction. By definition, the sum of two convex functions gives a convex function. Since $\|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})}$ is a norm and $\mathbf{k}_{t-1}$ is fixed, by definition it is convex. We have also assumed as part of the problem setting that each $f_t(\mathbf{k})$ is linear. Thus, $f_t(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})}$ must be convex. The remaining term is the negation of $\int_{z^{(t-1)}}^{z^{(t-1)} + \overline{c}(\mathbf{k})} \psi(u) du$. Let $w(\overline{c}(\mathbf{k})) = \int_{z^{(t-1)}}^{z^{(t-1)} + \overline{c}(\mathbf{k})} \psi(u) du$. By the fundamental theorem of calculus, we have

$$\nabla w(\overline{c}(\mathbf{k})) = \psi(z^{(t-1)} + \overline{c}(\mathbf{k})) \nabla \overline{c}(\mathbf{k}).$$

Let $b(\overline{c}(\mathbf{k})) = \psi(z^{(t-1)} + \overline{c}(\mathbf{k}))$. Then we have

$$\nabla^2 w(\overline{c}(\mathbf{k})) = \nabla^2 \overline{c}(\mathbf{k}) w(\overline{c}(\mathbf{k})) + \nabla \overline{c}(\mathbf{k}) b'(\overline{c}(\mathbf{k})) \nabla \overline{c}(\mathbf{k})^{\intercal}.$$

Since $\overline{c}(\mathbf{k})$ is piecewise linear by the definition of SOAD, we know that $\nabla^2 \overline{c}(\mathbf{k}) w(\overline{c}(\mathbf{k})) = 0$. Since ψ is monotonically decreasing on the interval $[0, 1]$, we know that $b'(\overline{c}(\mathbf{k})) < 0$, and thus $\nabla \overline{c}(\mathbf{k}) b'(\overline{c}(\mathbf{k})) \nabla \overline{c}(\mathbf{k})^\intercal$ is negative semidefinite. This implies that $w(\overline{c}(\mathbf{k}))$ is concave in $\mathbf{k}$.

Since the negation of a concave function is convex, this causes a contradiction, because the sum of two convex functions gives a convex function. Thus, $h_t(\cdot) = f_t(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)}+\overline{c}(\mathbf{k})} \psi(u)du$ is always convex under the assumptions of SOAD. □

By showing that $h_t(\cdot)$ is convex, it follows that the pseudo-cost minimization (5.2) in PCM is a convex minimization problem (i.e., it can be solved efficiently using numerical methods).

Proof of Theorem 5.3.3

In the following, we prove Theorem 5.3.3. In what follows, we let $\mathcal{I} \in \Omega$ denote an arbitrary valid SOAD instance. Let $z^{(j)} = \sum_{t \in [T]} \overline{c}(\mathbf{k}_t)$ denote the final utilization before the mandatory allocation. Also note that $z^{(t)} = \sum_{m \in [t]} \overline{c}(\mathbf{k}_m)$ is non-decreasing over t.

In what follows, we let η be defined as the solution to $\ln\left(\frac{U-L-D-2\tau}{U-U/\eta-D}\right) = \frac{1}{\eta}$, which has a closed form given by:

$$\eta := \left[W\left(\frac{(D+L-U+2\tau)\exp\left(\frac{D-U}{U}\right)}{U} \right) + \frac{U-D}{U} \right]^{-1}. \tag{5.10}$$

Note that setting η as above satisfies the following equality within the pseudo-cost function ψ (defined in Definition 5.3.1):

$$\psi(1) = U - \tau + (U/\eta - U + D + \tau)\exp(1/\eta) = L + D.$$

We start by proving Lemma 5.3.5, which states that $\texttt{OPT}$ is lower bounded by

$$\texttt{OPT}(\mathcal{I}) \geq \frac{\max\left\{\psi(z^{(j)}) - D, L\right\}}{O(\log n)}.$$

Proof of Lemma 5.3.5. Without loss of generality, denote the minimum gradient of any cost function (excluding $\texttt{OFF}$ states) by $\nabla_{\min}$. Suppose that a cost function f_m with $\nabla_{\min}$ gradient (at a dimension corresponding to any $\texttt{ON}$ state) arrives at timestep m.

Recall that PCM solves the following pseudo-cost minimization problem at time m:

$$\mathbf{k}_m = \operatorname*{arg\,min}_{\mathbf{k}\in K:\overline{c}(\mathbf{k})\leq 1-z^{(t-1)}} f_m(\mathbf{k}) + \|\mathbf{k}-\mathbf{k}_{m-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(m-1)}}^{z^{(m-1)}+\overline{c}(\mathbf{k})} \psi(u)du.$$

By assumption, since $f_m(\cdot)$ is linear and satisfies $\nabla f_m < \psi(z^{(j)}) - D$, there must exist a dimension in f_m (i.e., a service cost associated with an $\texttt{ON}$ state) that satisfies the following. Let $\texttt{ON}^{[d]} \subset [d]$ denote the index set (i.e., the dimensions in $\mathbf{k}$) that correspond to allocations in $\texttt{ON}$ states.

$$\exists i \in \texttt{ON}^{[d]} : f_m(\mathbf{k})_i \leq [\nabla_{\min} \cdot \overline{c}(\mathbf{k})]_i \,.$$

Also note that $\|\mathbf{k}-\mathbf{k}_{m-1}\|_{\ell_1(\mathbf{w})}$ is upper bounded by $(D+\tau)\overline{c}(\mathbf{k})$, since in the worst-case, PCM must pay the max movement and switching cost to move the allocation to the "furthest" point and make a decision $\mathbf{k}$.

Since ψ is monotone decreasing on the interval $z \in [0,1]$, by definition we have that $\mathbf{k}_m$ solving the true pseudo-cost minimization problem is lower-bounded by the $\breve{\mathbf{k}}_m$ solving the following minimization problem (specifically, the constraint satisfaction satisfies $\overline{c}(\breve{\mathbf{k}}_m) \leq \overline{c}(\mathbf{k}_m)$):

$$\breve{\mathbf{k}}_m = \operatorname*{arg\,min}_{\mathbf{k}\in K:\overline{c}(\mathbf{k})\leq 1-z^{(t-1)}} \nabla_{\min}\cdot \overline{c}(\mathbf{k}) + D\overline{c}(\mathbf{k}) + \tau\overline{c}(\mathbf{k}) - \int_{z^{(m-1)}}^{z^{(m-1)}+\overline{c}(\mathbf{k})} \psi(u)du. \quad (5.11)$$

By expanding the right hand side, we have:

$$\begin{aligned}
&\nabla_{\min}\cdot \overline{c}(\mathbf{k}) + D\overline{c}(\mathbf{k}) + \tau\overline{c}(\mathbf{k}) - \int_{z^{(m-1)}}^{z^{(m-1)}+\overline{c}(\mathbf{k})} \psi(u)du \\
&= \nabla_{\min}\cdot \overline{c}(\mathbf{k}) + D\overline{c}(\mathbf{k}) + \tau\overline{c}(\mathbf{k}) - \int_{z^{(m-1)}}^{z^{(m-1)}+\overline{c}(\mathbf{k})} \left[U - \tau + (U/\eta - U + D + \tau)\exp(u/\eta)\right] du \\
&= (\nabla_{\min} - U + D + \tau)\overline{c}(\mathbf{k}) - [(\tau - U + D)\eta + U]\left(\exp\left(\frac{z^{(m-1)} + \overline{c}(\mathbf{k})}{\eta}\right) - \exp\left(\frac{z^{(m-1)}}{\eta}\right)\right).
\end{aligned}$$

Letting $\overline{c}(\mathbf{k})$ be some scalar y (which is valid since we assume there is at least one dimension $i \in [d]$ where the growth rate of $f_m(\cdot)$ is at most $\nabla_{\min}$), the pseudo-cost minimization problem finds the value y that minimizes the following quantity:

$$(\nabla_{\min} - U + D + \tau)y - [(\tau - U + D)\eta + U]\left(\exp\left(\frac{z^{(m-1)} + y}{\eta}\right) - \exp\left(\frac{z^{(m-1)}}{\eta}\right)\right).$$

Taking the derivative of the above with respect to y yields the following:

$$\begin{aligned}
&\nabla_{\min} + D + \tau - U - \frac{[(\tau - U + D)\eta + U]\exp\left(\frac{z^{(m-1)}+y}{\eta}\right)}{\eta} \\
&= \nabla_{\min} + D - \psi(z^{(m-1)} + y). \quad (5.12)
\end{aligned}$$

Note that since the minimization problem is convex by Theorem 5.3.2, the unique solution to the above coincides with a point y where the derivative is zero. This implies that PCM will increase $\overline{c}(\mathbf{k})$ until $\nabla_{\min} = \psi(z^{(m-1)} + y) - D$, which further implies that $\psi(z^{(m)}) = \nabla_{\min} + D$.

Since this minimization $\breve{\mathbf{k}}_m$ is a lower bound on the true value of $\mathbf{k}_m$, this implies that $\psi(z^{(t)}) - D$ is a lower bound on the minimum service cost coefficient (excluding `OFF` states) seen *so far* at time t. Further, the *final utilization* $z^{(j)}$ gives that the minimum service cost coefficient over any cost function over the entire sequence is lower bounded by $\psi(z^{(j)}) - D$.

Note that the best choice for `OPT` is to service the entire workload at the minimum service cost if it is feasible. Since the vector space $(K, \|\cdot\|_{\ell_1(\mathbf{w})})$ used by PCM has at most $O(\log n)$ distortion with respect to the underlying metric used by `OPT` (see Definition 5.2.1), this implies that $\texttt{OPT}(\mathcal{I}) \geq \frac{\max\{\psi(z^{(j)})-D,\, L\}}{O(\log n)}$. □

Next, we prove Lemma 5.3.6, which states that the expected cost of PCM$(\mathcal{I})$ is upper bounded by $\mathbb{E}[\text{PCM}(\mathcal{I})] \leq \int_0^{z^{(j)}} \psi(u)du + (1 - z^{(j)})U + \tau z^{(j)}$.

Proof of Lemma 5.3.6. Recall that $z^{(t)} = \sum_{m \in [t]} \overline{c}(\mathbf{k}_m)$ is non-decreasing over t.

Observe that whenever $\overline{c}(\mathbf{k}_t) > 0$, we have that $f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} < \int_{z^{(t-1)}}^{z^{(t-1)}+\overline{c}(\mathbf{k}_t)} \psi(u)du$. Then, if $\overline{c}(\mathbf{k}_t) = 0$, which corresponds to the case when $\mathbf{k}_t$ allocates all of the marginal probability mass to `OFF` states, we have the following:

$$f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)}+\overline{c}(\mathbf{k}_t)} \psi(u)du = 0 + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - 0$$
$$= \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})}. \qquad (5.13)$$

This gives that for any timestep where $\overline{c}(\mathbf{k}_t) = 0$, we have the following inequality, which follows by observing that from Assumption 5.2, any marginal probability mass assigned to `ON` states in the previous timestep can be moved to `OFF` states at a cost of at most $\tau\overline{c}(\mathbf{k}_{t-1})$.

$$f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} \leq \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})}$$
$$\leq \tau\overline{c}(\mathbf{k}_{t-1}), \forall t \in [T] : \overline{c}(\mathbf{k}_t) = 0.$$

Since any timestep where $\overline{c}(\mathbf{k}_t) > 0$ implies that $f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} < \int_{z^{(t-1)}}^{z^{(t-1)}+\overline{c}(\mathbf{k}_t)} \psi(u)du$, we have the following inequality across all timesteps (*i.e., an upper bound on the excess cost not accounted for by the pseudo-cost*):

$$f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)}+\overline{c}(\mathbf{k}_t)} \psi(u)du \leq \tau\overline{c}(\mathbf{k}_{t-1}), \forall t \in [T].$$

Thus, we have

$$\begin{aligned}
\tau z^{(j)} &= \sum_{t\in[j]} \tau \bar{c}(\mathbf{k}_{t-1}) \\
&\geq \sum_{t\in[j]} \left[f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)}+\bar{c}(\mathbf{k}_t)} \psi(u)du \right] \\
&= \sum_{t\in[j]} \left[f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} \right] - \int_0^{z^{(j)}} \psi(u)du \\
&= \text{PCM}(\mathcal{I}) - (1 - z^{(j)})U - \int_0^{z^{(j)}} \psi(u)du.
\end{aligned}$$

Combining Lemma 5.3.5 and Lemma 5.3.6 gives

$$\mathsf{CR} \leq \frac{\mathbb{E}[\text{PCM}(\mathcal{I})]}{\mathtt{OPT}(\mathcal{I})} \leq \frac{\int_0^{z^{(j)}} \psi(u)du + (1 - z^{(j)})U + \tau z^{(j)}}{\max\{\psi(z^{(j)}) - D, L\}} \leq \eta,$$

where the last inequality holds since for any $z \in [0, 1]$:

$$\begin{aligned}
&\int_0^z \psi(u)du + \tau z + (1 - z)U \\
&= \int_0^z \left[U - \tau + (U/\eta - U + D + \tau) \exp(z/\eta) \right] du + (1 - z)U + \tau z \\
&= \left[((\tau - U + D)\eta + U)\, \mathrm{e}^{\frac{u}{\eta}} + Uu - \tau u \right]_0^z + (1 - z)U + \tau z \\
&= ((\tau - U + D)\eta + U)\, \mathrm{e}^{\frac{z}{\eta}} - (\tau - U + D)\eta \\
&= \eta \left[\left((\tau - U + D) + \frac{U}{\eta} \right) \mathrm{e}^{\frac{z}{\eta}} - \tau + U - D \right] \\
&= \eta[\psi(z) - D]. \qquad (5.14)
\end{aligned}$$

This completes the proof of Theorem 5.3.3. □

Proof of Theorem 5.3.4

In this section, we prove Theorem 5.3.4, which states that η (as defined in (5.4)) is the optimal competitive ratio for SOAD. To show this lower bound, we first define a family of special two-stage adversaries, a corresponding metric space X, and then show that the competitive ratio for any algorithm is lower bounded under the instances provided by these adversaries.

Prior work has shown that difficult instances for online search problems with a minimization objective occur when inputs arrive at the algorithm in an decreasing order of cost [78, 79, 176, 192]. For SOAD, we extend this idea and additionally consider

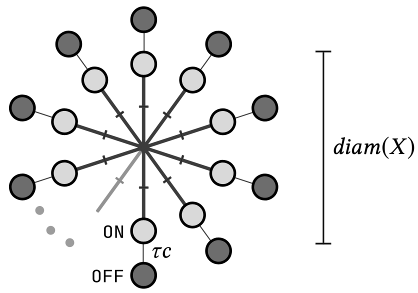


Figure 5.11: A motivating illustration of the lower bound star metric considered in Definition 5.D. Light gray circles represent the points of the metric space (X, d), and darker circles represent the OFF states of SOAD. Note that the distance between any two points in the metric is $\text{diam}(X) = Dc$.

how adaptive adversaries can strategically present good service cost functions at distant points in the metric first, followed by good service costs at the starting point (e.g., "at home"), to create a family of sequences that simultaneously penalize the online player for moving "too much" and for not moving enough.

We now formalize two such families of adversaries, namely $\{\mathcal{G}_y\}_{y\in[L,U]}$ and $\{\mathcal{A}_y\}_{y\in[L,U]}$, where $\mathcal{A}_y$ and $\mathcal{G}_y$ are both called y-adversaries.

Definition 5.D.1 (y-adversaries for SOAD)**.** *Let $m \in \mathbb{N}$ be sufficiently large, and $\sigma := {}^{(U-L)}/_{m}$. The metric space X is a weighted star metric with n points, each with an* ON *and* OFF *state. For the constraint function $c(\cdot)$, we set one throughput constant for all* ON *states such that $c^{(u)} \ll 1 : u \in X$. This value is henceforth simply denoted by c. See Figure 5.11 for an illustration.*

The movement cost can be represented by a weighted ℓ_1 norm $\|\cdot\|_{\ell_1(\mathbf{w})}$ that combines the spatial distances given by the metric with the temporal switching cost. Recall Assumption 5.2—at any single point u, the switching cost between the ON *and* OFF *states is given by $\tau c \cdot |x_t^{\mathtt{ON}^{(u)}} - x_{t-1}^{\mathtt{ON}^{(b)}}|$ (i.e., for two arbitrary allocations $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$) for $\tau > 0$. Furthermore, for any two disjoint points in the metric $u, v : u \neq v$, the distance between $\mathtt{ON}^{(u)}$ and $\mathtt{ON}^{(v)}$ is exactly $Dc = \text{diam}(X)$.*

Let y denote a value on the interval $[L, U]$, which represents the "best service cost function" presented by the y-adversary in their sequence. We define two distinct stages of the input.

*In **Stage 1**, the adversary presents two types of cost functions such that "good cost functions" are always at points that are distant to the online player. These cost functions are denoted (for convenience) as $Up(x) = Ux$, and $Down^i(x) = (U - i\sigma)x$, where one such cost function is delivered at each point's* ON *state, at each timestep.*

Without loss of generality, let the starting point be $s \in X$ and the start state be $\mathtt{OFF}^{(s)}$ (for both ALG *and* OPT*). In the first timestep, the adversary presents cost function*

$Up(x) = Ux$ at the starting point's ON *state (i.e.,* $\mathtt{ON}^{(s)}$*), and $Down^1(x)$ at all of the other $(n-1)$* ON *states.*

If ALG ever *moves a non-zero fractional allocation to an* ON *state other than the starting point, that point becomes* inactive*, meaning that the adversary will present $Up(x)$ at that location in the next timestep and for the rest of the sequence.*

In the second timestep, the adversary presents cost function $Up(x) = Ux$ at the starting point and any inactive points, and $Down^1(x)$ at all of the other ON *states. The adversary continues to sequentially present $Down^1(x)$ in this manner until it has presented it at least μ times (where $\mu \coloneqq 1/c$). It then moves on to present $Down^2(x)$, $Down^3(x)$, and so forth. The adversary follows the above pattern, presenting "good cost functions" to a shrinking subset of* ON *states until they present $Down^{m_y}(x) = yx$ up to μ times at any remaining active states. In the timestep after the last $Down^{m_y}(x) = yx$ is presented, the adversary presents $Up(x)$ everywhere, and **Stage 1** ends.*

*In **Stage 2**, the adversary presents $Up(x)$ and $Down^i(x)$ cost functions at the starting point s, in an alternating fashion. All other* ON *states are considered inactive in this stage, so they only receive $Up(x)$. In the first timestep, the adversary presents $Down^1(x)$ at the starting point, followed by $Up(x)$ in the following timestep. In the third timestep, the adversary presents $Down^2(x)$ at the starting point, followed by $Up(x)$ in the subsequent timestep. The adversary continues alternating in this manner until they present $Down^{m_y}(x) = yx$ at the starting point. In the $\mu - 1$ timesteps after $Down^{m_y}(x) = yx$ is presented, the adversary presents $Down^{m_y}(x) = yx$ at the starting point (allowing* OPT *to reduce their switching cost). Finally, the adversary presents $Up(x)$ everywhere for the final μ timesteps, and **Stage 2** ends.*

*The first family of y-adversaries, $\{\mathcal{G}_y\}_{y\in[L,U]}$, only uses **Stage 1**—the sequence ends when **Stage 1** ends. The second family, $\{\mathcal{A}_y\}_{y\in[L,U]}$, sequentially uses both stages—cost functions are presented using the full **Stage 1** sequence first, which is then followed by the full **Stage 2** sequence. This concatenated sequence ends when **Stage 2** ends.*

Note that the final cost function for any point in any y-adversary instance is always $Up(x)$.

Proof of Theorem 5.3.4. Let $s(y)$ and $t(y)$ denote *constraint satisfaction functions* mapping $[L, U] \to [0, 1]$, that fully describe ALG's expected deadline constraint satisfaction (i.e., $\mathbb{E}\left[\sum_{t\in[j]} \overline{c}(\mathbf{k})\right]$ before the mandatory allocation) during **Stage 1** and **Stage 2**, respectively.

Note that for large m, **Stage 1** and **Stage 2** for $y = z - \sigma$ are equivalent to first processing **Stage 1** and **Stage 2** for $y = z$, and then processing an additional batch of cost functions such that the best cost function observed is $\text{Down}^{m_{z-\sigma}}(x) = (z-\sigma)x$.

As the expected deadline constraint satisfactions at each timestep are unidirectional (irrevocable), we must have that $s(y)$ and $t(y)$ both satisfy $s(y-\sigma) \geq s(y)$ and $t(y-\sigma) \geq t(y)$, i.e., $s(y)$ and $t(y)$ are non-decreasing on $[L, U]$. Note that the optimal solutions for adversaries $\mathcal{G}_y$ and $\mathcal{A}_y$ are not the same.

In particular, we have that $\texttt{OPT}(\mathcal{G}_y) = \min\{y + D + \tau, U\}$. For relatively large y, the optimal solution may choose to satisfy the constraint at the starting point, but for sufficiently small y, the optimal solution on $\mathcal{G}_y$ may choose to move to a distant point. For $\mathcal{A}_y$, since the "good cost functions" arrive at the starting point, we have $\texttt{OPT}(\mathcal{A}_y) = y$ for any $y \in [L, U]$.

Due to the adaptive nature of the y-adversary, any $\texttt{ALG}$ incurs expected movement cost proportional to $s(y)$ during **Stage 1**. Furthermore, since $c \ll 1$ for all $\texttt{ON}$ states, note that as soon as $s(y) > c$, in expectation, $\texttt{ALG}$ has *moved away from* the starting point. Thus, during **Stage 2**, $\texttt{ALG}$ must also incur expected movement cost proportional to $t(y)$ as it "moves back" to the starting point. Let $\mathbf{l} = {}^{U}\!/\!{\eta^\star} - 2\tau$ denote the worst marginal cost that an $\eta^\star$-competitive $\texttt{ALG}$ should be willing to accept in either stage. The total expected cost incurred by an $\eta^\star$-competitive online algorithm $\texttt{ALG}$ on adversaries $\mathcal{A}_y$ and $\mathcal{G}_y$ can be expressed as follows:

$$\mathbb{E}[\texttt{ALG}(\mathcal{G}_y)] = s(\mathbf{l})\mathbf{l} - \int_{\mathbf{l}}^{y} u ds(u) + Ds(y) + (1 - s(y))U + 2\tau s(y) \tag{5.15}$$

$$\begin{aligned} \mathbb{E}[\texttt{ALG}(\mathcal{A}_y)] = s(\mathbf{l})\mathbf{l} - \int_{\mathbf{l}}^{y} u ds(u) + Ds(y) + t(\mathbf{l})\mathbf{l} \\ - \int_{\mathbf{l}}^{y} u dt(u) + Dt(y) \\ + (1 - s(y) - t(y))U + 2\tau\left[s(y) + t(y)\right]. \end{aligned} \tag{5.16}$$

In the above expressions, $uds(u)$ is the expected cost of buying $ds(u)$ constraint satisfaction at cost u. The same convention extends to $udt(u)$. $Ds(y)$ and $Dt(y)$ represent the movement cost paid by $\texttt{ALG}$ during **Stage 1** and **Stage 2**, respectively, given that "good service cost functions" arrive at distant points (i.e., at distance D). $(1 - s(y))U$ and $(1 - s(y) - t(y))U$ give the expected cost of the mandatory allocation on adversary $\mathcal{G}_y$ and $\mathcal{A}_y$, respectively. Similarly, 2τ gives the expected temporal switching cost due to allocation decisions made (excepting the mandatory allocation).

For any $\eta^\star$-competitive algorithm, the constraint satisfaction functions $s(\cdot)$ and $t(\cdot)$ must simultaneously satisfy $\mathbb{E}[\texttt{ALG}(\mathcal{G}_y)] \leq \eta^\star \texttt{OPT}(\mathcal{G}_y)$ and $\mathbb{E}[\texttt{ALG}(\mathcal{A}_y)] \leq \eta^\star \texttt{OPT}(\mathcal{A}_y)$ for all $y \in [L, U]$. This gives a necessary condition that the functions must satisfy as follows:

$$s(\mathbf{l})\mathbf{l} - \int_{\mathbf{l}}^{y} u ds(u) + Ds(y) + (1 - s(y))U + 2\tau s(y) \leq \eta^\star \left[y + D + 2\tau \right]$$

$$\begin{aligned} s(\mathbf{l})\mathbf{l} - \int_{\mathbf{l}}^{y} u ds(u) + t(\mathbf{l})\mathbf{l} - \int_{\mathbf{l}}^{y} u dt(u) \\ + (1 - s(y) - t(y))U + \left[D + 2\tau \right] (s(y) + t(y)) \leq \eta^\star \left[y \right]. \end{aligned}$$

By integration by parts, the above expressions imply that the constraint satisfaction functions $s(y)$ and $t(y)$ must satisfy the following conditions:

$$\begin{aligned} s(y) &\geq \frac{U - \eta^\star y - \eta^\star D - 2\eta^\star \tau}{U - y - D - 2\tau} - \frac{1}{U - y - D - 2\tau} \int_{\mathbf{l}}^{y} s(u) du \\ t(y) &\geq \frac{(y + D - U + 2\tau) s(y)}{U - y - D - 2\tau} + \frac{U - \eta^\star y}{U - y - D - 2\tau} \\ &\quad - \frac{1}{U - y - D - 2\tau} \int_{\mathbf{l}}^{y} s(u) + t(u) du \\ &\geq \frac{\eta^\star D + \eta^\star 2\tau}{U - y - D - 2\tau} - \frac{1}{U - y - D - 2\tau} \int_{\mathbf{l}}^{y} t(u) du. \end{aligned} \tag{5.17}$$

In what follows, we substitute $\mathbf{l} = \frac{U}{\eta} - 2\tau$. By Grönwall's Inequality [217, Theorem 1, p. 356], we have the following:

$$\begin{aligned} s(y) &\geq \frac{U - \eta^\star y - \eta^\star D - 2\eta^\star \tau}{U - y - D - 2\tau} - \int_{\mathbf{l}}^{y} \frac{U - \eta^\star u - \eta^\star D - 2\eta^\star \tau}{(U - u - D - 2\tau)^2} du \\ &\geq \frac{U - \eta^\star y - \eta^\star D - 2\eta^\star \tau}{U - y - D - 2\tau} \\ &\quad - \left[\frac{2\tau\eta^\star + U\eta^\star - 2\tau - U}{u - U + D + 2\tau} - \eta^\star \ln\left(U - u - D - 2\tau\right) \right]_{\mathbf{l}}^{y} \\ &\geq \eta^\star \ln\left(U - y - D - 2\tau\right) \\ &\quad - \eta^\star \ln\left(U - \mathbf{l} - D - 2\tau\right) - \frac{\eta^\star D + \eta^\star 2\tau}{\frac{U}{\eta^\star} - U + D} \qquad \forall y \in [L, U]; \\ t(y) &\geq \frac{\eta^\star D + \eta^\star 2\tau}{U - y - D - 2\tau} - \int_{\mathbf{l}}^{y} \frac{\eta^\star D + \eta^\star 2\tau}{(U - u - D - 2\tau)^2} du \end{aligned}$$

$$\geq \frac{\eta^\star D + \eta^\star 2\tau}{U - y - D - 2\tau} - \left[\frac{-\eta^\star D - \eta^\star 2\tau}{u - U + D + 2\tau} \right]_{\mathbf{l}}^{y}$$
$$\geq \frac{\eta^\star D + \eta^\star 2\tau}{\mathbf{l} - U + D + 2\tau}.$$

By the definition of the adversaries $\mathcal{G}_y$ and $\mathcal{A}_y$, we have that $t(L) \leq 1 - s(L)$, and thus $s(L) \leq 1 - t(L)$. We combine this inequality with the above bounds to give the following condition for any $\eta^\star$-competitive online algorithm:

$$\eta^\star \ln (U - L - D - 2\tau) - \eta^\star \ln (U - \mathbf{l} - D - 2\tau) - \frac{\eta^\star D + \eta^\star 2\tau}{\frac{U}{\eta^\star} - U + D}$$
$$\leq s(L) \leq 1 - t(L) \leq 1 - \frac{\eta^\star D + \eta^\star 2\tau}{\mathbf{l} - U + D + 2\tau}.$$

The optimal $\eta^\star$ is obtained when the above inequality is binding, and is given by the solution to the following transcendental equation (after substituting $\frac{U}{\eta} - 2\tau$ for $\mathbf{l}$):

$$\ln \left[\frac{U - L - D - 2\tau}{U - {}^{U}\!/_{\eta^\star} - D} \right] = \frac{1}{\eta^\star}. \tag{5.18}$$

The solution to the above is given by the following (note that $W(\cdot)$ denotes the Lambert W function):

$$\eta^\star \to \left[W \left(\frac{(D + L - U + 2\tau) \exp \left(\frac{D-U}{U} \right)}{U} \right) + \frac{U - D}{U} \right]^{-1}. \tag{5.19}$$

□

5.E Proofs for Section 5.4

Proof of Theorem 5.4.3

In this section, we prove Theorem 5.4.3, which states that ST-CLIP is $(1 + \varepsilon)$-consistent for any $\varepsilon \in (0, \eta - 1]$, and $O(\log n)\gamma^{(\varepsilon)}$-robust, where $\gamma^{(\varepsilon)}$ is the solution to (5.5).

Proof. We show the result by separately considering consistency (the competitive ratio when advice is correct) and robustness (the competitive ratio when advice is not correct) in turn.

Recall that the black-box advice Adv is denoted by a decision $\mathbf{a}_t$ at each time t. Throughout the proof, we use shorthand notation SC_t to denote the expected cost of ST-CLIP up to time t, and Adv_t to denote the cost of Adv up to time t. We start by proving Lemma 5.4.5 to show that ST-CLIP is $(1 + \varepsilon)$-consistent.

Proof of Lemma 5.4.5. First, we note that the constrained optimization enforces that the expected cost of ST-CLIP so far plus a term that forecasts the mandatory allocation is always within $(1+\varepsilon)$ of the advice. There is always a feasible $\mathbf{p}_t$ that satisfies the constraint, because setting $\mathbf{k}_t = \Phi \mathbf{a}_t$ is always within the feasible set. Formally, if timestep $j \in [T]$ denotes the timestep marking the start of the mandatory allocation, the constraint in (5.7) holds for every timestep $t \in [j]$.

Thus, to show $(1+\varepsilon)$ consistency, we must resolve the cost of any actions during the *mandatory allocation* and show that the final expected cost of ST-CLIP is upper bounded by $(1+\varepsilon)\textsc{Adv}$.

Let $\mathcal{I} \in \Omega$ be an arbitrary valid SOAD sequence. If the mandatory allocation begins at timestep $j < T$, both ST-CLIP and Adv must greedily satisfy the constraint during the last m timesteps $[j, T]$. This is assumed to be feasible, and the cost due to switching in and out of `ON / OFF` states is assumed to be negligible as long as m is sufficiently large.

Let $(1 - z^{(j-1)})$ denote the remaining deadline constraint that must be satisfied by ST-CLIP in expectation at these final m timesteps, and let $(1 - A^{(j-1)})$ denote the remaining deadline constraint to be satisfied by Adv. We consider two cases, corresponding to the cases where ST-CLIP has *underprovisioned* with respect to Adv (i.e., it has completed less of the deadline constraint in expectation) and *overprovisioned* (i.e., completed more of the deadline constraint), respectively.

Case 1: ST-CLIP$(\mathcal{I})$ has "underprovisioned" $((1 - z^{(j-1)}) > (1 - A^{(j-1)}))$

In this case, ST-CLIP must satisfy *more* of the deadline constraint (in expectation) during the mandatory allocation compared to Adv. From the previous timestep, we know that the following constraint holds:

$$\begin{aligned} \mathrm{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L + (A^{(j-1)} - z^{(j-1)})U \\ \leq (1+\varepsilon)\left[\textsc{Adv}_{j-1} + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L\right]. \end{aligned}$$

Let $\{\mathbf{p}_t\}_{t \in [j,T]}$ and $\{\mathbf{a}_t\}_{t \in [j,T]}$ denote the decisions made by ST-CLIP and Adv during the mandatory allocation, respectively. Conditioned on the fact that ST-CLIP has completed $z^{(j-1)}$ fraction of the deadline constraint in expectation, we have that $\mathbb{E}\left[\sum_{t=j}^T c(\mathbf{p}_t)\right] = (1 - z^{(j-1)})$ and $\sum_{t=j}^T c(\mathbf{a}_t) = (1 - A^{(j-1)})$.

Considering $\{f_t(\cdot)\}_{t \in [j,T]}$, by assumption we have a lower bound based on L, namely $\sum_{t=j}^T f_t(\mathbf{a}_t) \geq L \sum_{t=j}^T c(\mathbf{a}_t)$. For the service costs that ST-CLIP must incur *over and* *above* what Adv incurs, we have a upper bound based on U, so $\mathbb{E}\left[\sum_{t=j}^T f_t(\mathbf{p}_t)\right] \leq \sum_{t=j}^T f_t(\mathbf{a}_t) + U(\sum_{t=j}^T c(\mathbf{p}_t) - \sum_{t=j}^T c(\mathbf{a}_t))$.

Note that the worst case for ST-CLIP occurs when $\sum_{t=j}^T f_t(\mathbf{a}_t)$ exactly matches this lower bound, i.e., $\sum_{t=j}^T f_t(\mathbf{a}_t) = L\sum_{t=j}^T c(\mathbf{a}_t)$, as Adv is able to satisfy the rest of the deadline constraint at the best possible marginal price. Furthermore, note that if ST-CLIP and Adv are in different points of the metric at time j, the term $\mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1})$ in the left-hand-side of the constraint allows ST-CLIP to "move back" and follow Adv just before the mandatory allocation begins, thus leveraging the same cost functions as Adv. By the constraint in the previous timestep, we have the following:

$$\begin{aligned}
&\mathrm{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L + (A^{(j-1)} - z^{(j-1)})U \\
&\qquad \leq (1+\varepsilon)[\mathrm{Adv}_{j-1} + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L], \\
&\mathrm{SC}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L\sum_{t=j}^T c(\mathbf{a}_t) + U\left(\sum_{t=j}^T c(\mathbf{p}_t) - \sum_{t=j}^T c(\mathbf{a}_t)\right) \\
&\qquad \leq (1+\varepsilon)\left[\mathrm{Adv}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L\sum_{t=j}^T c(\mathbf{a}_t)\right] \leq (1+\varepsilon)\mathrm{Adv}(\mathcal{I}). \\
&\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right] \leq (1+\varepsilon)\mathrm{Adv}(\mathcal{I}).
\end{aligned} \tag{5.20}$$

Case 2: ST-CLIP($\mathcal{I}$) has "overprovisioned" ($(1 - z^{(j-1)}) \leq (1 - A^{(j-1)})$)

In this case, ST-CLIP must satisfy *less* of the deadline constraint (in expectation) during the mandatory allocation compared to Adv.

From the previous timestep, we know that the following constraint holds:

$$\begin{aligned}
&\mathrm{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + (1 - z^{(j-1)})L \\
&\qquad \leq (1+\varepsilon)\left[\mathrm{Adv}_{j-1} + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L\right].
\end{aligned}$$

Let $\{\mathbf{p}_t\}_{t\in[j,T]}$ and $\{\mathbf{a}_t\}_{t\in[j,T]}$ denote the decisions made by ST-CLIP and Adv during the mandatory allocation, respectively. As previously, we have that $\mathbb{E}\left[\sum_{t=j}^T c(\mathbf{p}_t)\right] = (1 - z^{(j-1)})$ and $\sum_{t=j}^T c(\mathbf{a}_t) = (1 - A^{(j-1)})$.

Considering $\{f_t(\cdot)\}_{t\in[j,T]}$, we have a lower bound on $\sum_{t=j}^T f_t(\cdot)$ based on L, namely $\sum_{t=j}^T f_t(\mathbf{p}_t) \geq L\sum_{t=j}^T c(\mathbf{p}_t)$ and $\sum_{t=j}^T f_t(\mathbf{a}_t) \geq L\sum_{t=j}^T c(\mathbf{a}_t)$. Since ST-CLIP has "overprovisioned," we know $\mathbb{E}\left[\sum_{t=j}^T c(\mathbf{p}_t)\right] \leq \sum_{t=j}^T c(\mathbf{a}_t)$, and thus it follows that $\mathbb{E}\left[\sum_{t=j}^T f_t(\mathbf{p}_t)\right] \leq \sum_{t=j}^T f_t(\mathbf{a}_t)$.

By the constraint in the previous timestep, we have:

$$\begin{aligned}\frac{\mathrm{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + (1 - z^{(j-1)})L}{\textsc{Adv}_{j-1} + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L} \\ = \frac{\mathrm{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + L\sum_{t=j}^T c(\mathbf{p}_t)}{\textsc{Adv}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L\sum_{t=j}^T c(\mathbf{a}_t)} \\ \leq (1+\varepsilon).\end{aligned}$$

Let $y = \mathbb{E}\left[\sum_{t=j}^T f_t(\mathbf{p}_t)\right] - L\sum_{t=j}^T c(\mathbf{p}_t)$, and let $y' = \sum_{t=j}^T f_t(\mathbf{a}_t) - L\sum_{t=j}^T c(\mathbf{a}_t)$. By definition, $y \geq 0$ and $y' \geq 0$. Note that by resolving the mandatory allocation and by definition, we have that the final expected cost $\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right] \leq \mathrm{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + L\sum_{t=j}^T c(\mathbf{p}_t) + y$, and $\textsc{Adv}(\mathcal{I}) \geq \textsc{Adv}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L\sum_{t=j}^T c(\mathbf{a}_t) + y'$.

Furthermore, since ST-CLIP has "overprovisioned" and by the linearity of the cost functions $f_t(\cdot)$, we have that $y \leq y'$. Combined with the constraint from the previous timestep, we have the following bound:

$$\begin{aligned}\frac{\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right]}{\textsc{Adv}(\mathcal{I})} &\leq \frac{\mathrm{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + L\sum_{t=j}^T c(\mathbf{p}_t) + y}{\textsc{Adv}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L\sum_{t=j}^T c(\mathbf{a}_t) + y'} \\ &\leq \frac{\mathrm{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + L\sum_{t=j}^T c(\mathbf{p}_t)}{\textsc{Adv}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L\sum_{t=j}^T c(\mathbf{a}_t)} \\ &\leq (1+\varepsilon). \qquad (5.21)\end{aligned}$$

Thus, by combining the bounds in each of the above two cases, the result follows, and we conclude that ST-CLIP is $(1+\varepsilon)$-consistent with accurate advice. □

Having proved consistency, we next prove Lemma 5.4.5 to show that ST-CLIP is $O(\log n)\gamma^{(\varepsilon)}$-robust.

Proof of Lemma 5.4.6. Let $\varepsilon \in (0, \eta^\star - 1]$ be the target consistency (recalling that ST-CLIP is $(1+\varepsilon)$-consistent), and let $\mathcal{I} \in \Omega$ denote an arbitrary valid SOAD sequence. To prove the robustness of ST-CLIP, we consider two "bad cases" for the advice $\textsc{Adv}(\mathcal{I})$, and show that in the worst-case, ST-CLIP's competitive ratio is bounded by $O(\log n)\gamma^{(\varepsilon)}$.

Case 1: $\textsc{Adv}(\mathcal{I})$ is "inactive"

Consider the case where ADV accepts nothing during the main sequence and instead satisfies the entire deadline constraint at the end of the sequence immediately before the mandatory allocation, incurring the worst possible movement & switching cost in the process. In the worst-case, this gives that $\text{ADV}(\mathcal{I}) = U + D + \tau$.

Based on the consistency constraint (and using the fact that ST-CLIP will always be "overprovisioning" with respect to ADV throughout the main sequence), we can derive an upper bound on the constraint satisfaction that ST-CLIP is "allowed to accept" from the robust pseudo-cost minimization. Recall the following constraint:

$$\begin{aligned} &\text{SC}_{t-1} + f_t(\mathbf{p}_t) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + \tau c(\mathbf{a}_t) + (1 - z^{(t-1)} - c(\mathbf{p}_t))L \\ &\quad \leq (1+\varepsilon)\left[\text{ADV}_t + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L\right]. \end{aligned}$$

Proposition 5.E.1. *Under "inactive" advice, z_{PCM} is an upper bound on the amount that* ST-CLIP *can accept from the pseudo-cost minimization in expectation without violating $(1+\varepsilon)$-consistency, and is defined as:*

$$z_{PCM} = \gamma^{(\varepsilon)} \ln\left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right].$$

Proof. Consider an arbitrary timestep t. If ST-CLIP is *not* allowed to make a decision that makes progress towards the constraint (i.e., it cannot accept anything more from the robust pseudo-cost minimization), we have that $c(\mathbf{p}_t)$ is restricted to be 0. Recall that $\mathbf{a}_t = \delta_s$ (where δ_s is the Dirac measure at the starting `OFF` state) for any timesteps before the mandatory allocation, because the advice is assumed to be inactive. By definition, since any cost functions accepted so far in expectation (i.e., in SC_{t-1}) can be attributed to the robust pseudo-cost minimization, we have the following in the worst-case, using the same techniques used in the proof of Theorem 5.3.3:

$$\text{SC}_{t-1} = \int_0^{z^{(t-1)}} \psi^{(\varepsilon)}(u)du + \tau z^{(t-1)}.$$

Combining the above with the left-hand side of the consistency constraint, we have the following. Observe that $\mathbf{p}_t$ is in an `OFF` state, $\mathbf{a}_t = \delta_s$, and any prior movement costs to make progress towards the constraint can be absorbed into the pseudo-cost ψ since $\|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} \geq \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1})$. Furthermore, in the worst-case, $\mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) = Dz^{(t-1)}$ (i.e., the pseudo-cost chooses to move to a point in the metric that is a distance D away from ADV).

$$\begin{aligned}\mathrm{SC}_{t-1} &+ \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + (1 - z^{(t-1)})L \\ &= \int_0^{z^{(t-1)}} \psi^{(\varepsilon)}(u)du + \tau z^{(t-1)} + Dz^{(t-1)} + (1 - z^{(t-1)})L.\end{aligned}$$

As stated, let $z^{(t-1)} = z_{\mathrm{PCM}}$. Then by properties of the pseudo-cost,

$$\begin{aligned}\mathrm{SC}_{t-1} &+ \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + (1 - z_{\mathrm{PCM}})L \\ &= \int_0^{z_{\mathrm{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\mathrm{PCM}} + (1 - z_{\mathrm{PCM}})U \\ &\quad + (1 - z_{\mathrm{PCM}})L + Dz_{\mathrm{PCM}} - (1 - z_{\mathrm{PCM}})U \\ &= \gamma^{(\varepsilon)} \left[\psi^{(\varepsilon)}(z_{\mathrm{PCM}}) - D\right] + (1 - z_{\mathrm{PCM}})L + Dz_{\mathrm{PCM}} - (1 - z_{\mathrm{PCM}})U \\ &= \gamma^{(\varepsilon)}L + (L - U)\left(1 - \gamma^{(\varepsilon)} \ln\left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right]\right) + Dz_{\mathrm{PCM}} \\ &= \gamma^{(\varepsilon)}L + L - U + (U - L + D)\,\gamma^{(\varepsilon)} \ln\left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right].\end{aligned}$$

Substituting for the definition of $\gamma^{(\varepsilon)}$, we obtain:

$$\begin{aligned}\mathrm{SC}_{t-1} &+ \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + (1 - z_{\mathrm{PCM}})L \\ &= \gamma^{(\varepsilon)}L + L - U + (U - L + D)\,\gamma^{(\varepsilon)} \ln\left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right] \\ &= \left[\varepsilon L + U - \gamma^{(\varepsilon)}(U - L + D) \ln\left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right]\right] \\ &\quad + L - U + (U - L + D)\,\gamma^{(\varepsilon)} \ln\left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right] \\ &= \varepsilon L + L = (1 + \varepsilon)L.\end{aligned}$$

This completes the proposition, since $(1 + \varepsilon)L$ is exactly the right-hand side of the consistency constraint (note that $(1 + \varepsilon)\left[\mathrm{ADV}_t + \tau c(\mathbf{a}_t) + (1 - A_t)L\right] = (1 + \varepsilon)L$). □

If ST-CLIP is constrained to use at most z_{PCM} of its utilization to be robust, the remaining $(1 - z_{\mathrm{PCM}})$ utilization must be used for the mandatory allocation and/or to follow ADV. Note that if ST-CLIP has moved away from ADV's point in the metric, and ADV turns out to be "inactive" bad advice that incurs sub-optimal service cost late in the sequence, the consistency constraint will become non-binding and ST-CLIP will not have to move back to follow ADV in the metric. Thus, we have the following worst-case competitive ratio for ST-CLIP, specifically for Case 1, where we assume $\texttt{OPT}(\mathcal{I}) \to {}^{\psi^{(\varepsilon)}(z_{\text{PCM}})}\!/\!{}_{O(\log n)} = {}^{L}\!/\!{}_{O(\log n)}$, as in, e.g., Lemma 5.3.5:

$$\frac{\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right]}{\texttt{OPT}(\mathcal{I})} \leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{PCM}})U}{{}^{L}\!/\!{}_{O(\log n)}} \tag{5.22}$$

$$\leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{PCM}})D + (1 - z_{\text{PCM}})U}{{}^{L}\!/\!{}_{O(\log n)}}. \tag{5.23}$$

By the definition of $\psi^{(\varepsilon)}(\cdot)$, we have the following:

$$\begin{aligned}
\frac{\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right]}{\texttt{OPT}(\mathcal{I})} &\leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{PCM}})D + (1 - z_{\text{PCM}})U}{{}^{L}\!/\!{}_{O(\log n)}} \\
&\leq \frac{\gamma^{(\varepsilon)}\left[\psi^{(\varepsilon)}(z_{\text{PCM}}) - 2D\right]}{{}^{L}\!/\!{}_{O(\log n)}} \\
&\leq \frac{\gamma^{(\varepsilon)}\left[L + 2D - 2D\right]}{{}^{L}\!/\!{}_{O(\log n)}} \\
&\leq O(\log n)\gamma^{(\varepsilon)}.
\end{aligned}$$

Case 2: $\textsc{Adv}(\mathcal{I})$ is "overactive"

We now consider the case where $\textsc{Adv}$ incurs bad service cost due to "accepting" cost functions which it "should not" (i.e., $\textsc{Adv}(\mathcal{I}) \gg \texttt{OPT}(\mathcal{I})$). Let $\textsc{Adv}(\mathcal{I}) = \mathbb{V} \gg \texttt{OPT}(\mathcal{I})$ (i.e., the final total cost of $\textsc{Adv}$ is $\mathbb{V}$ for some $\mathbb{V} \in [L, U]$, and $\mathbb{V}$ is much greater than the optimal solution).

This is without loss of generality, since we can assume that $\mathbb{V}$ is the "best marginal service and movement cost" incurred by $\textsc{Adv}$ at a particular timestep and the consistency ratio changes strictly in favor of $\textsc{Adv}$. Based on the consistency constraint, we can derive a lower bound on the amount that ST-CLIP *must* accept from $\textsc{Adv}$ in expectation to stay $(1 + \varepsilon)$-consistent. To do this, we consider the following sub-cases:

- **Sub-case 2.1**: Let $\mathbb{V} \geq \frac{U+D}{1+\varepsilon}$.

In this sub-case, ST-CLIP can fully ignore the advice, because the following consistency constraint is never binding (note that $\textsc{Adv}_t \geq \frac{U+D}{1+\varepsilon} A^{(t)}$):

$$
\begin{aligned}
&\mathrm{SC}_{t-1} + f_t(\mathbf{p}_t) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) \\
&\qquad + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L + (A^{(t)} - z^{(t-1)} - c(\mathbf{p}_t))U \\
&\qquad \leq (1+\varepsilon)\left[\textsc{Adv}_t + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L\right], \\
&(D+\tau)c(\mathbf{a}_t) + (1 - A^{(t)})L + (A^{(t)})U \leq (1+\varepsilon)\left[\mathbb{V}c(\mathbf{a}_t) + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L\right], \\
&(D+\tau)A^{(t)} + (1 - A^{(t)})L + UA^{(t)} \leq (1+\varepsilon)\left[\frac{U+D}{1+\varepsilon}A^{(t)} + \tau A^{(t)} + (1 - A^{(t)})L\right].
\end{aligned}
$$

• **Sub-case 2.2**: Let $\mathbb{V} \in (L, \frac{U+D}{1+\varepsilon})$.

In this case, in order to remain $(1+\varepsilon)$-consistent, ST-CLIP must follow Adv and incur some "bad cost," denoted by $\mathbb{V}$. We derive a lower bound that describes the minimum amount that ST-CLIP must follow Adv in order to always satisfy the consistency constraint.

Proposition 5.E.2. *Under "overactive" advice, $z_{\textsc{Adv}}$ is a lower bound on the amount that* ST-CLIP *must accept from the advice in order to always satisfy the consistency constraint, and is defined as:*

$$
z_{\textsc{Adv}} \geq 1 - \frac{\mathbb{V}\varepsilon}{U + D - \mathbb{V}}.
$$

Proof. For the purposes of showing this lower bound, we assume there are no marginal service costs in the instance that would otherwise be accepted by the robust pseudo-cost minimization.

Based on the consistency constraint, we have the following:

$$
\begin{aligned}
&\mathrm{SC}_{t-1} + f_t(\mathbf{p}_t) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) \\
&\qquad + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L + (A^{(t)} - z^{(t-1)} - c(\mathbf{p}_t))U \\
&\qquad \leq (1+\varepsilon)\left[\textsc{Adv}_t + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L\right].
\end{aligned}
$$

We let $f_t(\mathbf{p}_t) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + \tau c(\mathbf{a}_t) \leq v c(\mathbf{p}_t)$ for any $\mathbf{p}_t : c(\mathbf{p}_t) < c(\mathbf{a}_t)$, which holds by linearity of the cost functions $f_t(\cdot)$ and a prevailing condition that $c(\mathbf{p}_t) \leq c(\mathbf{a}_t)$ for the "bad service costs" accepted by Adv. Note that ST-CLIP must "follow" Adv to distant points in the metric to avoid violating consistency, and recall that $\mathbf{p}_t = \mathbf{a}_t$ is always in the feasible set. Under this condition that ST-CLIP follows Adv, $\mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) + \tau c(\mathbf{a}_t)$ is upper bounded by the movement cost of Adv and absorbed into $\mathbb{V}$. The term $\mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t)$ is upper bounded by $D(A^{(t)} - c(\mathbf{p}_t))$ by Assumption 5.2 of the metric.

$$
\begin{aligned}
&\mathrm{SC}_{t-1} + \mathbb{V}c(\mathbf{p}_t) + L - LA^{(t)} + UA^{(t)} + DA^{(t)} - Uz^{(t-1)} - Uc(\mathbf{p}_t) - Dc(\mathbf{p}_t) \\
&\qquad \leq (1+\varepsilon)\left[\mathbb{V}A^{(t-1)} + \mathbb{V}c(\mathbf{a}_t) + L - LA^{(t)}\right], \\
&\mathbb{V}c(\mathbf{p}_t) - Dc(\mathbf{p}_t) - Uc(\mathbf{p}_t) \\
&\qquad \leq (1+\varepsilon)\left[\mathbb{V}A^{(t-1)} + \mathbb{V}c(\mathbf{a}_t) + L - LA^{(t)}\right] \\
&\qquad\qquad - \mathrm{SC}_{t-1} - L + LA^{(t)} - UA^{(t)} - DA^{(t)} + Uz^{(t-1)}, \\
&\mathbb{V}c(\mathbf{p}_t) - Dc(\mathbf{p}_t) - Uc(\mathbf{p}_t) \\
&\qquad \leq \mathbb{V}A^{(t)} - DA^{(t)} - UA^{(t)} - \mathrm{SC}_{t-1} + Uz^{(t-1)} \\
&\qquad\qquad + \varepsilon\left[\mathbb{V}A^{(t-1)} + \mathbb{V}c(\mathbf{a}_t) + L - LA^{(t)}\right], \\
&c(\mathbf{p}_t) \geq \frac{\mathbb{V}A^{(t)} - DA^{(t)} - UA^{(t)} - \mathrm{SC}_{t-1} + Uz^{(t-1)} + \varepsilon\left[\mathbb{V}A^{(t)} + L - LA^{(t)}\right]}{\mathbb{V} - D - U}.
\end{aligned}
$$

In the event that $A^{(t-1)} = 0$ (i.e., nothing has been accepted so far by either Adv or ST-CLIP), we have:

$$
\begin{aligned}
c(\mathbf{p}_t) &\geq \frac{\mathbb{V}c(\mathbf{a}_t) - Dc(\mathbf{a}_t) - Uc(\mathbf{a}_t) + \varepsilon\left[\mathbb{V}c(\mathbf{a}_t) + L - Lc(\mathbf{a}_t)\right]}{\mathbb{V} - D - U}, \\
c(\mathbf{p}_t) &\geq c(\mathbf{a}_t) - \frac{\varepsilon\left[\mathbb{V}c(\mathbf{a}_t) + L - Lc(\mathbf{a}_t)\right]}{U + D - \mathbb{V}}.
\end{aligned}
$$

Through a recursive definition, we can show that for any $A^{(t)}$, given that ST-CLIP has satisfied $z^{(t-1)}$ of the deadline constraint by following Adv so far, it must set $\mathbf{p}_t$ such that:

$$
z^{(t)} \geq z^{(t-1)} + c(\mathbf{a}_t) - \frac{\varepsilon\left[\mathbb{V}c(\mathbf{a}_t) + L - Lc(\mathbf{a}_t)\right]}{U + D - \mathbb{V}}.
$$

Continuing the assumption that $\mathbb{V}$ is constant, if ST-CLIP has accepted $z^{(t-1)}$ thus far, we have the following if we assume that all of the constraint satisfaction up to this point happened in a single previous timestep m:

$$\begin{aligned}
c(\mathbf{p}_t) &\geq A^{(t)} - \frac{Uc(\mathbf{p}_m) + Dc(\mathbf{p}_m) - \mathrm{SC}_{t-1} + \varepsilon\left[\mathbb{V}A^{(t)} + L - LA^{(t)}\right]}{U + D - \mathbb{V}}, \\
c(\mathbf{p}_t) &\geq c(\mathbf{a}_t) + c(\mathbf{a}_m) - c(\mathbf{p}_m) \\
&\quad - \frac{\varepsilon\left[\mathbb{V}(c(\mathbf{a}_t) + c(\mathbf{a}_m)) + L - L(c(\mathbf{a}_t) + c(\mathbf{a}_m))\right]}{U + D - \mathbb{V}}, \\
c(\mathbf{p}_t) + c(\mathbf{p}_m) &\geq c(\mathbf{a}_t) + c(\mathbf{a}_m) - \frac{\varepsilon\left[\mathbb{V}(c(\mathbf{a}_t) + c(\mathbf{a}_m)) + L - L(c(\mathbf{a}_t) + c(\mathbf{a}_m))\right]}{U + D - \mathbb{V}}, \\
z^{(t)} &\geq A^{(t)} - \frac{\varepsilon\left[\mathbb{V}A^{(t)} + L - LA^{(t)}\right]}{U + D - \mathbb{V}}.
\end{aligned}$$

This gives intuition into the desired z_{ADV} bound. The above motivates that the *aggregate* expected constraint satisfaction by ST-CLIP at any given timestep t must satisfy a lower bound. Consider that the worst case for Sub-case 2.2 occurs when all of the v prices accepted by ADV arrive first, before any prices that would be considered by the pseudo-cost minimization. Then let $A^{(t)} = 1$ for some arbitrary timestep t, and we have the stated lower bound on z_{ADV}. □

If ST-CLIP is forced to use z_{ADV} of its utilization to be $(1 + \varepsilon)$ consistent against ADV, that leaves at most $(1 - z_{\text{ADV}})$ utilization for robustness. We define $z' = \min(1 - z_{\text{ADV}}, z_{\text{PCM}})$ and consider the following two cases.

- **Sub-case 2.2.1**: if $z' = z_{\text{PCM}}$, the worst-case competitive ratio is bounded by the following. Note that if $z' = z_{\text{PCM}}$, the amount of utilization that ST-CLIP can use to "be robust" is exactly the same as in **Case 1**, and we again have that $\mathtt{OPT}(\mathcal{I}) \to \psi^{(\varepsilon)}(z_{\text{PCM}})/O(\log n) = L/O(\log n)$:

$$\begin{aligned}
&\frac{\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right]}{\mathtt{OPT}(\mathcal{I})} \\
&\leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{ADV}} - z_{\text{PCM}})U + z_{\text{ADV}}\mathbb{V}}{L/O(\log n)}, \\
&\leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{PCM}})D + (1 - z_{\text{PCM}})U}{L/O(\log n)}, \\
&\leq O(\log n)\gamma^{(\varepsilon)}. \qquad (5.24)
\end{aligned}$$

- **Sub-case 2.2.2**: if $z' = 1 - z_{\text{ADV}}$, the worst-case competitive ratio is bounded by the following. Note that ST-CLIP *cannot* use z_{PCM} of its utilization for robustness, so the following bound assumes that the "robust service costs" accepted by ST-CLIP are bounded by the *worst* $(1 - z_{\text{Adv}})$ *fraction* of the pseudo-cost function $\psi^{(\varepsilon)}$ (note that $\psi^{(\varepsilon)}$ is non-increasing on $z \in [0, 1]$):

$$\frac{\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right]}{\texttt{OPT}(\mathcal{I})} \leq \frac{\int_0^{1-z_{\text{Adv}}} \psi^{(\varepsilon)}(u)du + \tau(1 - z_{\text{Adv}}) + z_{\text{Adv}}\mathbb{V}}{L/O(\log n)}.$$

Note that if $z' = 1 - z_{\text{Adv}}$, we know that $1 - z_{\text{Adv}} < z_{\text{PCM}}$, which further gives the following by definition of z_{Adv}:

$$\begin{aligned} 1 - z_{\text{PCM}} < 1 - \frac{\mathbb{V}\varepsilon}{U + D - \mathbb{V}}, \\ \mathbb{V}\varepsilon < (U + D - \mathbb{V})z_{\text{PCM}}, \\ \mathbb{V} < \frac{U + D}{(1 + \frac{\varepsilon}{z_{\text{PCM}}})}. \end{aligned}$$

By plugging $\mathbb{V}$ back into the definition of z_{Adv}, we have that $z_{\text{Adv}}\mathbb{V} \leq \left(\frac{(1-z_{\text{PCM}})(U+D)}{1+\frac{\varepsilon}{z_{\text{PCM}}}}\right)$, giving the following:

$$\begin{aligned} &\frac{\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right]}{\texttt{OPT}(\mathcal{I})} \\ &\leq \frac{\int_0^{1-z_{\text{Adv}}} \psi^{(\varepsilon)}(u)du + \tau(1 - z_{\text{Adv}}) + \left(\frac{(1-z_{\text{PCM}})(U+D)}{1+\frac{\varepsilon}{z_{\text{PCM}}}}\right)}{L/O(\log n)}, \\ &\leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{PCM}})D + (1 - z_{\text{PCM}})U}{L/O(\log n)}, \\ &\leq O(\log n)\gamma^{(\varepsilon)}. \end{aligned}$$

Thus, by combining the bounds in each of the above two cases, the result follows, and we conclude that ST-CLIP is $O(\log n)\gamma^{(\varepsilon)}$-robust. □

Having proven Lemma 5.4.5 (consistency) and Lemma 5.4.6 (robustness), the statement of Theorem 5.4.3 follows: ST-CLIP is $(1 + \varepsilon)$-consistent and $O(\log n)\gamma^{(\varepsilon)}$-robust given any advice for SOAD. □

Proof of Theorem 5.4.4

In this section, we prove Theorem 5.4.4, which states that $\gamma^{(\varepsilon)}$ (as defined in (5.5)) is the optimal robustness for any $(1 + \varepsilon)$-consistent learning-augmented SOAD algorithm.

Proof. To show this result, we build off the same y-adversaries for SOAD defined in Definition 5.D, where $y \in [L, U]$. For the purposes of showing consistency, we define a slightly tweaked adversary $\mathcal{A}'_y$:

Definition 5.E.3 ($\mathcal{A}'_y$ adversary for learning-augmented SOAD)**.** *Recall the $\mathcal{A}_y$ adversary defined in Definition 5.D. During **Stage 1** of the adversary's sequence, $\mathcal{A}_y$ and $\mathcal{A}'_y$ are identical. In **Stage 2**, $\mathcal{A}'_y$ presents Up(x) at the starting point's $\mathtt{ON}$ state $\mathtt{ON}^{(s)}$ once, followed by Down$^{m_y}(x) = yx$ at $\mathtt{ON}^{(s)}$. All other $\mathtt{ON}$ states are considered inactive in this stage, so they only receive Up(x). In the $\mu - 1$ timesteps after Down$^{m_y}(x) = yx$ is presented, the adversary presents Down$^{m_y}(x) = yx$ at the starting point in the metric (allowing $\mathtt{OPT}$ and* Adv *to reduce their switching cost). Finally, the adversary presents Up(x) everywhere for the final μ timesteps, and **Stage 2** ends.*

As in the proof of Theorem 5.3.4, for adversary $\mathcal{A}'_y$, the optimal offline objective is $\mathtt{OPT}(\mathcal{A}'_y) \to y$. Against these adversaries, we consider two types of advice—the first is *bad* advice, which stays with their full allocation at the starting $\mathtt{OFF}$ state (i.e., $\mathbf{a}_t = \delta_s$) for all timesteps $t < j$ before the mandatory allocation, incurring a final cost of $U + 2\tau$.

On the other hand, *good* advice sets $\mathbf{a}_t = \delta_s$ for all timesteps up to the first timestep when y is revealed at the starting point in the metric, after which it sets $a_t^{\mathtt{ON}^{(s)}} = 1/\mu$ to achieve final cost $\text{Adv}(\mathcal{A}'_y) = \mathtt{OPT}(\mathcal{A}'_y) = y + 2\tau/\mu$.

We let $(s(y) + t(y))$ denote a *robust constraint satisfaction function* $[L, U] \to [0, 1]$, that fully quantifies the actions of a learning-augmented algorithm $\mathtt{ALG}$ playing against adaptive adversary $\mathcal{A}'_y$, where $(s(y) + t(y))$ gives the progress towards the deadline constraint under the instance $\mathcal{A}'_y$ before (either) the mandatory allocation or the black-box advice sets $a_t^{\mathtt{ON}^{(s)}} > 0$. Since the conversion is unidirectional (irrevocable), we must have that $s(y - \sigma) + t(y - \sigma) \geq (s(y) + t(y))$, i.e., $(s(y) + t(y))$ is non-increasing in $[L, U]$.

As in the proof of Theorem 5.3.4, the adaptive nature of $\mathcal{A}'_y$ forces any algorithm to incur a movement and switching cost proportional to $(s(y) + t(y))$ during the robust phase, specifically denoted by $(D + 2\tau)(s(y) + t(y))$. Recall that by the proof of Theorem 5.3.4, for any γ-competitive online algorithm $\mathtt{ALG}$, we have the following condition on $(s(y) + t(y))$ for all $y \in [L, U]$:

$$(s(y) + t(y)) \geq \gamma \ln (U - y - D - 2\tau) - \gamma \ln (U - \mathbf{l} - D - 2\tau) . \tag{5.25}$$

Furthermore, we have that the expected cost of ALG on adversary $\mathcal{A}'_y$ is given by:

$$\mathbb{E}\left[\texttt{ALG}(\mathcal{A}'_y)\right] = s(\mathbf{l})\mathbf{l} - \int_{\mathbf{l}}^{y} u ds(u) + Ds(y) + t(\mathbf{l})\mathbf{l} - \int_{\mathbf{l}}^{y} u dt(u) + Dt(y) \\ + (1 - s(y) - t(y))U + 2\tau\left[s(y) + t(y)\right]. \tag{5.26}$$

To simultaneously be α-consistent when the advice *is* correct, ALG must satisfy $\mathbb{E}\left[\texttt{ALG}(\mathcal{A}'_L)\right] \leq \alpha\texttt{OPT}(\mathcal{A}'_L) = \alpha L$. If the advice is correct, ALG must pay an additional factor of D to move back and follow Adv in the worst case—but can satisfy the rest of the deadline constraint at the best cost functions L. It must also still pay for switching incurred by the robust algorithm (recall that OPT pays no switching cost). Using integration by parts, we have:

$$\int_{\mathbf{l}}^{L} s(u) + t(u) du + [2D + 2\tau]\left(s(L) + t(L)\right) \\ +(1 - s(y) - t(y))L + L\left(s(y) + t(y)\right) \leq \alpha L,$$

$$\int_{\mathbf{l}}^{L} s(u) + t(u) du + [2D + 2\tau]\left(s(L) + t(L)\right) \leq \alpha L - L. \tag{5.27}$$

By combining equations (5.25) and (5.27), and substituting $\mathbf{l} = {}^{U}\!/\gamma$, the robust constraint satisfaction function $(s(y) + t(y))$ of any γ-robust and α-consistent online algorithm must satisfy:

$$\gamma \int_{\mathbf{l}}^{L} \ln\left(\frac{U - u - D - 2\tau}{U - {}^{U}\!/\gamma - D - 2\tau}\right) du + [2D + 2\tau]\left[\gamma \ln\left(\frac{U - L - D - 2\tau}{U - {}^{U}\!/\gamma - D - 2\tau}\right)\right] \\ \leq \alpha L - L.$$

When all inequalities are binding, this equivalently gives that

$$\alpha \geq \gamma + 1 - \frac{U}{L} + \frac{\gamma(U - L + D)}{L} \ln\left(\frac{U - L - D - 2\tau}{U - {}^{U}\!/\gamma - D - 2\tau}\right). \tag{5.28}$$

We define α such that $\alpha := (1 + \varepsilon)$. By substituting for α into (5.28), we recover the definition of $\gamma^{(\varepsilon)}$ as given by (5.5), which subsequently completes the proof. Thus, we conclude that any $(1 + \varepsilon)$-consistent algorithm for SOAD is at least $\gamma^{(\varepsilon)}$-robust. □

5.F Proofs for Section 5.5

Proof of Corollary 5.5.1

In this section, we prove Corollary 5.5.1, which states that PCM is $O(\log n)\eta$-competitive for SOAD-T, the variant of SOAD where distances in the metric (X, d) are allowed to be time-varying (see Section 5.5).

We recall a few mild assumptions on the SOAD-T problem that directly imply the result. Let $d_t(\cdot, \cdot)$ denote the distance between points in X at time $t \in [T]$. We redefine D as $D = \sup_{t\in[T]} \left(\max_{u,v\in X: u\neq v} \frac{d_t(u, v)}{\min\{c^{(u)},c^{(v)}\}} \right)$, i.e., it is an upper bound on distance between any two points in the metric at any time over the horizon T. Recall that the temporal switching cost between `ON` and `OFF` states at a single point is defined as *non-time-varying*, so $\| \cdot \|_{\ell_1(\beta)}$ gives the temporal switching cost for all $t \in [T]$.

We also assume that the tree embedding-based vector space $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$ defined by Definition 5.2.1 is appropriately reconstructed at each step, and that PCM has knowledge of the current distances (i.e., $\| \cdot \|_{\ell_1(\mathbf{w})}$ accurately reflects $d_t(\cdot, \cdot)$ at time t).

Under these assumptions, every step in the proof of Theorem 5.3.3 holds. In particular, note that in Lemma 5.3.5, the only fact about the distance function that is used is the fact that the distance between two `ON` states is upper bounded by D, in (5.11), and that the vector space $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$ has expected $O(\log n)$ distortion with respect to the underlying metric, which follows by definition. In Lemma 5.3.6, most of PCM's movement cost is absorbed into the integral over the pseudo-cost function ψ, and the only other fact about the distance that is used is that the distance between `ON` and `OFF` states at a single point $u \in X$ is fixed and bounded by $\tau c^{(u)}$, which follows by definition. Thus, we conclude that PCM is $O(\log n)\eta$-competitive for SOAD-T. □

Proof of Corollary 5.5.2

In this section, we prove Corollary 5.5.2, which states that a minor change to the consistency constraint enables ST-CLIP to be $(1 + \varepsilon)$-consistent and $O(\log n)\gamma^{(\varepsilon)}$-robust for SOAD-T, given any $\varepsilon \in (0, \eta - 1]$.

We start by recalling assumptions on the SOAD-T problem that inform the result. Recall that we redefine D as $D = \sup_{t\in[T]} \left(\max_{u,v\in X: u\neq v} \frac{d_t(u, v)}{\min\{c^{(u)},c^{(v)}\}} \right)$, i.e., it is an upper bound on distance between any two points at any time over the horizon T, and the temporal switching cost between `ON` and `OFF` states at a single point is defined as *non-time-varying*, so $\| \cdot \|_{\ell_1(\beta)}$ gives the temporal switching cost for all $t \in [T]$.

We also assume that the tree embedding-based vector space $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$ defined by Definition 5.2.1 is appropriately reconstructed at each step, and that ST-CLIP has knowledge of the current distances (i.e., $\| \cdot \|_{\ell_1(\mathbf{w})}$ and $\mathbb{W}^1(\cdot, \cdot)$ accurately reflect $d_t(\cdot, \cdot)$ at time t).

For the consistency constraint, we define a modified Wasserstein-1 distance function $\overline{\mathbb{W}}^1$ that will be used in the consistency constraint to hedge against the time-varying properties of the metric. This distance computes the optimal transport between two distributions on Δ_S while assuming that the underlying distances are given by $\overline{d}(\cdot,\ \cdot)$, which is itself defined such that $\overline{d}(u,\ v) = D \min\{c^{(u)}, c^{(v)}\} : u, v \in X : u \neq v$.

$$\overline{\mathbb{W}}^1(\mathbf{p}, \mathbf{p}') := \min_{\pi \in \Pi(\mathbf{p}, \mathbf{p}')} \mathbb{E}\left[\overline{d}(\mathbf{x}, \mathbf{x}')\right], \tag{5.29}$$

where $(\mathbf{x}, \mathbf{x}') \sim \pi_t$ and $\Pi(\mathbf{p}, \mathbf{p}')$ is the set of distributions over X^2 with marginals $\mathbf{p}$ and $\mathbf{p}'$.

Intuitively, the purpose of $\overline{\mathbb{W}}^1$ is to leverage the D upper bound between points in the metric to give a "worst-case optimal transport" distance between distributions, assuming that the time-varying distances increase in future timesteps.

To this end, within the definition of the consistency constraint (5.7), ST-CLIP for SOAD-T replaces the term $\mathbb{W}^1(\mathbf{p}, \mathbf{a}_t)$ with $\overline{\mathbb{W}}^1(\mathbf{p}, \mathbf{a}_t)$; this term hedges against the case where ST-CLIP must move to follow ADV in a future timestep, and in the time-varying distances case, we charge ST-CLIP an extra amount to further hedge against the case where the underlying distances between ST-CLIP and ADV grow in future timesteps.

Paralleling the proof of Theorem 5.4.3, we consider consistency and robustness independently.

Consistency. For consistency, according to the proof of Lemma 5.4.5, we show that resolving the mandatory allocation remains feasible in the time-varying case. First, note that there is always a feasible $\mathbf{p}_t$ that satisfies the consistency constraint, since even in the time-varying case, setting $\mathbf{k}_t = \Phi \mathbf{a}_t$ is always within the feasible set—this follows by observing that at a given time t, if ST-CLIP has moved away from ADV, it has already "prepaid" a worst-case movement cost of $\overline{\mathbb{W}}^1(\mathbf{p}_m, \mathbf{a}_m)$ (for some previous timestep m) in order to move back and follow ADV. The remainder of the proof of Lemma 5.4.5 only uses the fact that at the beginning of the mandatory allocation (at time $j \in [T]$), $\overline{\mathbb{W}}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1})$ is an upper bound on the movement cost paid by ST-CLIP (if necessary) while migrating to ADV's points in the metric to take advantage of the same service cost functions. By definition of $\overline{\mathbb{W}}^1$, this follows for any time-varying distance $d_t(\cdot,\ \cdot)$, and the remaining steps in the proof hold. □

Robustness. For robustness, following the proof of Lemma 5.4.6, we show that a certain amount of ST-CLIP's utilization can be "set aside" for robustness. First, note that in Case 1 (i.e., "inactive" advice), the proof of Proposition 5.E.1 only requires that $\mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t)$ is bounded by D, which follows immediately by the definition of $\overline{\mathbb{W}}^1$—the remaining steps for Case 1 follow. In Case 2 (i.e., "overactive" advice), note that the proof of Sub-case 2.1 similarly only requires that $\mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t)$ is bounded by D, which follows because $\mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) \leq \overline{\mathbb{W}}^1(\mathbf{p}_t, \mathbf{a}_t) \leq D \max\{c(\mathbf{p}_t), c(\mathbf{a}_t)\}$. Likewise, the remaining steps for Case 2 follow. □

Thus, we conclude that ST-CLIP is $(1 + \varepsilon)$-consistent and $O(\log n)\gamma^{(\varepsilon)}$-robust for SOAD-T, given any $\varepsilon \in (0, \eta - 1]$.

Part II

Beyond the Black Box: New Frontiers in Uncertainty, Risk, and Reliability

Chapter 6

RISK-SENSITIVE ONLINE ALGORITHMS

In the first part of this thesis, we focused on the design of learning-augmented algorithms for online problems, where a decision-maker seeks to obtain *consistency* relative to the performance of a black-box AI/ML advice algorithm, and worst-case *robustness* in case of poor AI/ML advice performance. In Chapters 3 and 5, however, the primary algorithms we designed were *randomized*, with robustness bounds only holding in expectation. This could potentially expose decision-makers to poor empirical performance, depending on the randomness inherent in the algorithm. This is potentially problematic for *risk-sensitive* decision-makers in high-stakes application domains, who may seek to minimize their likelihood of incurring costs of a particular magnitude.

Inspired by this challenge, we study the design of *risk-sensitive online algorithms*, in which risk measures are used in the competitive analysis of randomized online algorithms. We introduce the CVaR_δ-competitive ratio (δ-CR) using the conditional value-at-risk of an algorithm's cost, which measures the expectation of the $(1-\delta)$-fraction of worst outcomes against the offline optimal cost, and use this measure to study three online optimization problems: continuous-time ski rental, discrete-time ski rental, and one-max search. These problems are prototypical problems in the literature on online algorithms and online optimization, and have connections to applications such as peak-aware economic dispatch in microgrids [95] and energy trading [96]. The structure of the optimal δ-CR and algorithm varies significantly between problems: we prove that the optimal δ-CR for continuous-time ski rental is $2 - 2^{-\Theta(\frac{1}{1-\delta})}$, obtained by an algorithm described by a delay differential equation. In contrast, in discrete-time ski rental with buying cost B, there is an abrupt phase transition at $\delta = 1 - \Theta(\frac{1}{\log B})$, after which the classic deterministic strategy is optimal. Similarly, one-max search exhibits a phase transition at $\delta = \frac{1}{2}$, after which the classic deterministic strategy is optimal; we also obtain an algorithm that is asymptotically optimal as $\delta \downarrow 0$ that arises as the solution to a delay differential equation.

This chapter is primarily based on the following paper:

[1] N. Christianson, B. Sun, S. Low, and A. Wierman. "Risk-Sensitive Online Algorithms." Accepted for Presentation at the Conference on Learn-

ing Theory 2024: https://proceedings.mlr.press/v247/christianson24a.html. arXiv: 2405.09859 [cs], [Online]. Available: http://arxiv.org/abs/2405.09859.

6.1 Introduction

Randomness can improve decision-making performance in many online problems; for instance, randomization improves the competitive ratio of online ski rental from 2 to $\frac{e}{e-1}$ [231], of metrical task systems (MTS) from linear to polylogarithmic in number of states [76, 142], and of online search from polynomial to logarithmic in the fluctuation ratio [78, 79]. However, this improved performance can only be obtained on average over multiple problem instances, as a randomized algorithm may vary wildly in its performance on any particular run. While this may not pose a concern for decision-making agents facing a large number of problem instances, such variability may be undesirable if an agent has only a small number of instances to solve, or if they are sensitive to risks of a particular magnitude or likelihood.

Numerous fields, including economics, finance, and decision science, have fielded research on risk aversion and alternative *risk measures* that enable modifying decision-making objectives to accommodate these risk preferences (e.g., [232–236]). One of the most well-studied risk measures in recent years, due to its nice properties (as a *coherent* risk measure) and computational tractability, is the *conditional value-at-risk* (CVaR_δ), which measures the expectation of a random loss/reward on its $(1-\delta)$-fraction of worst outcomes [237–239]. CVaR_δ and other risk measures have been applied to problems spanning finance and insurance [240, 241], energy systems [48, 49], and robotic control [89, 242], and have been studied as an objective in place of the expectation in MDPs [85, 87, 243], bandits [244, 245], and online learning [83, 246, 247].

Despite the significant extent of literature on risk-sensitive algorithms for online learning with the conditional value-at-risk, there has been no work on the design and analysis of *competitive* algorithms for online optimization problems like ski rental, online search, knapsack, function chasing, or MTS with risk-sensitive objectives. These types of online optimization problems have deep connections with online learning [116, 130, 131], but also substantial qualitative differences due varied problem structures and the competitive analysis framework. Coupled with their practical applications to problems like TCP acknowledgement [248], online matching [249], dynamic power management [94], peak-aware economic dispatch in microgrids [95], and energy trading [96], we are thus motivated to ask: how can we design competitive online algorithms when we care about the CVaR_δ of the cost/reward, and what are the optimal competitive ratios for different problems?

In this work, we begin to work toward answering this question, studying risk sensitivity in the design of competitive online algorithms for online optimization. In particular, we focus on two of the prototypical problems in online optimization: *ski rental*, which, as a special case of MTS, encapsulates the fundamental "rent vs. buy" tradeoff inherent in online optimization with switching costs [94, 120], and *one-max search*, which exhibits a complementary "accept vs. wait" tradeoff fundamental to constrained online optimization [250, 251]. While both of these problems are simple to pose, they both reflect crucial components of the difficulty of more complicated online optimization problems, and thus serve as ideal analytic testbeds for investigating the design of risk-sensitive algorithms in online optimization.

Contributions

In this work, we define a novel version of the competitive ratio that penalizes a randomized algorithm's cost via the conditional value at risk (CVaR_δ), which we call the CVaR_δ-competitive ratio (δ-CR). We then study the design of algorithms for several online problems with the δ-CR objective. We make contributions along three fronts:

(1) Optimal Risk-Sensitive Online Algorithms We find the *optimal* CVaR_δ-competitive algorithm for continuous-time ski rental with any δ and characterize its δ-CR as $2 - 2^{-\Theta(\frac{1}{1-\delta})}$. For discrete-time ski rental, we analytically characterize the optimal CVaR_δ-competitive algorithm when $\delta = O(\frac{1}{B})$, where B is the buying cost, and we prove that there is a phase transition at $\delta = 1 - \Theta(\frac{1}{\log B})$, after which the optimal δ-CR coincides with the deterministic optimal $2 - \frac{1}{B}$. Finally, we propose an algorithm for one-max search whose δ-CR is asymptotically optimal for small δ, and we prove that one-max search exhibits a phase transition at $\delta = \frac{1}{2}$, after which the optimal δ-CR coincides with the deterministic optimal $\sqrt{\theta}$, where θ is the so-called "fluctuation ratio" of the problem.

(2) Techniques For continuous-time ski rental and one-max search, we show that the conditional value-at-risk of an algorithm's cost can be written as an integral expression of its inverse cumulative distribution function. This parametrization is useful both for proving analytic bounds on algorithms' δ-CR, and as a source for optimal algorithms for these problems: it is through this formulation that we obtain the delay differential equation describing the optimal algorithm for continuous-time ski rental, and similarly how we obtain our algorithm for one-max search, which is asymptotically optimal when δ is small. For both versions of ski rental, our results rely on structural characterizations of the optimal algorithm which, while evocative of similar results from the ski rental literature, require significantly more care due to the complicated behavior of the conditional value-at-risk.

(3) Insights We gain several new insights from our results. The phase transitions in the discrete-time ski rental and one-max search problems, where δ sufficiently large implies that the optimal δ-CR is the deterministic optimal competitive ratio, suggests that there is a sharp limit to the benefit that randomization can yield in certain risk-sensitive online problems. Moreover, the qualitative difference between the continuous- and discrete-time ski rental problems—namely, the fact that the latter has a phase transition while the former does not—indicates that continuous and discrete problems may, in general, behave differently when risk sensitivity is introduced.

Related Work

Risk-aware online algorithms As mentioned earlier, while numerous problems in MDPs, bandits, and online learning have been studied with the conditional value-at-risk and other risk measures penalizing the objective, we are not aware of existing work in the literature designing competitive online algorithms for online optimization with such objectives. Even-Dar et al. [83] consider the related problem of online learning with expert advice and rewards depending on the Sharpe ratio and mean-variance risk measures; they prove lower bounds precluding the possibility of obtaining sublinear regret in this setting as well as upper bounds for several relaxed objectives. While their work also considers a notion of competitive ratio against the best fixed expert in its lower bounds, their upper bounds focus on regret-style results, and their problem setting is markedly different from those we consider. A related problem is the demonstration of *high-probability guarantees* on the competitive ratio of randomized online algorithms, which was studied by Komm et al. [252] for a general class of online problems. However, their work is concerned with proving that existing algorithms have performance close to some nominal value with high probability, rather than designing new algorithms that are provably optimal given an agent's particular risk preferences and the distribution over algorithm performance.

Closest to our current work is the recent paper of Dinitz et al. [253] on risk-constrained algorithms for ski rental, where the objective remains to minimize the competitive ratio defined in terms of expected cost, but algorithms must satisfy additional constraints on the likelihood that their cost will exceed a specified value. This amounts to imposing constraints on the *value-at-risk (*VaR*)*, or quantiles, of the algorithm's competitive ratio; in contrast, we focus on algorithms that are optimal for a *risk-sensitive objective* involving the conditional value at risk, which considers not just the likelihood of exceeding a certain value, but the expectation over the resulting tail of the distribution. VaR and CVaR have been exhaustively compared in the financial literature (e.g., [241, 254]), and CVaR, which is a so-called *coherent* risk measure, often exhibits more favorable robustness and handling of tail events than VaR, which is not coherent [235]. Indeed, VaR is very sensitive to problem structure and parameter selection, leading to the interesting non-continuous behavior in solution structure observed in [253]. CVaR does not beget such sensitivity, but influences the solution structure in its own unique way: for continuous-time ski rental and one-max search, we obtain algorithms that result from the solution of delay differential equations.

Beyond worst-case analysis of algorithms The strengthening and weakening of the adversary as δ is varied in the CVaR_δ-competitive ratio is similar in spirit to beyond worst-case analysis, where the adversary is weakened or additional information is provided to enable improved bounds over the pessimistic and unrealistic adversarial setting. There is a significant breadth of work in the literature applying these ideas to online optimization and other online problems. Some notable directions on this subject are smoothed analysis [255, 256], in which an adversary's decision is tempered with stochastic noise; algorithms with advice [257, 258], in which an algorithm receives a small number of accurate bits of information about the problem instance in advance; and algorithms with predictions [69, 70, 167, 177, 193], in which algorithms are augmented with potentially unreliable predictions about the problem instance, and algorithms seek to exploit these predictions when they are accurate while maintaining worst-case guarantees when they are not.

Notation

Throughout, capital letters (e.g., X) refer to random variables on $\mathbb{R}$, which we interchangeably refer to via their measures (e.g., $\mu \in \mathscr{B}(\mathbb{R})$) or their cumulative distribution functions (e.g., $F_X(x) = \mu(-\infty, x])$. Given a random variable X with support bounded in the interval $[a, b]$, we define its inverse CDF as $F_X^{-1}(p) = \inf\{x \in [a, b] : F_X(x) \geq p\}$; note that, given the bounded support, this definition agrees with the standard definition (where the infimum is taken over all of $\mathbb{R}$) for all $p \in (0, 1]$, with the only disagreement being at $p = 0$, where $F_X^{-1}(0) = \operatorname{ess\,inf} X$, whereas the standard definition yields $-\infty$; this variant is well-established in the literature (e.g., [259, Definition 1.16]). We use this definition to ensure finiteness of F_X^{-1} on $[0, 1]$, and the bounded support of X will be clear by context whenever the inverse CDF is discussed. $\mathbb{R}_+$ denotes the nonnegative reals and $\mathbb{R}_{++}$ the strictly positive reals, and $\mathbb{N}$ denotes the natural numbers. The notation $[\cdot]^+$ refers to the $\max\{\cdot, 0\}$ function, and for any $N \in \mathbb{N}$, we write $[N] = \{1, \ldots, N\}$ and denote by Δ_N the N-dimensional probability simplex. For a vector $\mathbf{x} \in \mathbb{R}^n$, we denote its ith entry x_i. The function $W_k(x)$ refers to the kth branch of the Lambert W function, which is defined as a solution to $W_k(x)e^{W_k(x)} = x$ (see, e.g., [216]).

6.2 Background & Preliminaries

In this section, we introduce risk measures and the conditional value-at-risk, and give overviews of the three online problems we study in this work.

Risk Measures and the Conditional Value-at-Risk

A *risk measure* is a mapping from the set of $\mathbb{R}$-valued random variables to $\mathbb{R}$ that gives a deterministic valuation of the *risk* associated with a particular random loss. As risk preferences can vary by decision-making agent and application, many different risk measures have been introduced and studied in the literature (see, e.g., [260, Chapter 6] for several examples). A prominent class of measures that has emerged in practice due to its favorable properties is the set of *coherent risk measures* [235]. Perhaps one of the most well-studied coherent risk measures in recent years is the *conditional value-at-risk* (CVaR): the CVaR at probability level δ of a random variable X, written $\mathrm{CVaR}_\delta[X]$, is the expectation of X on the δ-tail of its distribution, i.e., its $(1-\delta)$-fraction of worst outcomes. It can be defined in several ways:

Definition 6.2.1 (Conditional Value-at-Risk)**.** *Let X be a real-valued random variable with CDF F_X. If X has a density f, then for $\delta \in [0, 1)$ the conditional value-at risk at level δ of X is defined as the expectation of X, conditional on its outcome lying in the δ-tail of its distribution [237]:*

$$\mathrm{CVaR}_\delta[X] = \mathbb{E}[X | X \geq F_X^{-1}(\delta)].$$

For a general random loss X with probability measure μ, $\mathrm{CVaR}_\delta[X]$ *can be defined in several equivalent ways [238, 239, 261]:*

$$\begin{aligned}\mathrm{CVaR}_\delta[X] &= \inf_{t\in\mathbb{R}}\left\{t + \frac{1}{1-\delta}\mathbb{E}[X-t]^+\right\} \\ &= \frac{1}{1-\delta}\int_\delta^1 F_X^{-1}(p)\,\mathrm{d}p \\ &= \sup_{\nu\in\mathcal{Q}} \mathbb{E}_{Y\sim\nu}[Y],\end{aligned} \tag{6.1}$$

where in the final expression, $\mathcal{Q}$ is an uncertainty set of probability measures defined as

$$\mathcal{Q} = \{\nu : \mu = \beta\nu + (1-\beta)\rho \textit{ for some measure } \rho \textit{ and } \beta \in [1-\delta, 1]\}.$$

The first expression in (6.1) is a variational form of CVaR_δ, and is useful for tractable formulations of risk-sensitive optimization problems. The latter two expressions highlight the intuition that $\mathrm{CVaR}_\delta[X]$ computes the expected loss of X on the worst $(1-\delta)$-fraction of outcomes in its distribution, or, in the parlance of [261] which we sometimes adopt, on the "worst $(1-\delta)$-sized subpopulation."

From the above definition it is clear that $\mathrm{CVaR}_0[X] = \mathbb{E}[X]$ and $\lim_{\delta\uparrow 1}\mathrm{CVaR}_\delta[X] \to \operatorname{ess\,sup} X$, the largest value that X can take [262]; we thus define $\mathrm{CVaR}_1[X] := \operatorname{ess\,sup} X$, so that CVaR_δ is defined for all $\delta \in [0, 1]$.

Online Algorithms and Competitive Analysis

In the study of online algorithms, algorithm performance is typically measured via the *competitive ratio*, or the worst case ratio in (expected) cost between an algorithm and the offline optimal strategy that knows all uncertainty in advance.

Definition 6.2.2 (Competitive ratio)**.** *Consider an online problem with uncertainty drawn adversarially from a set of instances $\mathcal{I}$. Let* Alg *be a deterministic online algorithm for the problem, and let* Opt *be the offline optimal algorithm.* Alg*'s* ***competitive ratio* (CR)** *is the worst-case ratio in cost between* Alg *and* Opt *over all problem instances:*

$$\mathrm{CR}(\textsc{Alg}) := \sup_{I\in\mathcal{I}} \frac{\textsc{Alg}(I)}{\textsc{Opt}(I)}.$$

If Alg *has competitive ratio C, it is also called* **C*-competitive***. *If* Alg *is a randomized algorithm, then the competitive ratio is defined with its expected cost:*

$$\mathrm{CR}(\textsc{Alg}) := \sup_{I\in\mathcal{I}} \frac{\mathbb{E}[\textsc{Alg}(I)]}{\textsc{Opt}(I)},$$

where the expectation is taken over ALG*'s randomness.*

In our work, we introduce a new version of the competitive ratio for randomized algorithms that goes beyond expected performance: instead, we penalize a randomized algorithm via the ratio between the conditional value-at-risk of its cost and the offline optimal's cost, terming this metric the CVaR_δ-*competitive ratio* (abbreviated δ-CR).

Definition 6.2.3 (CVaR_δ-Competitive Ratio)**.** *Let* ALG *be a randomized algorithm, and let* OPT *be the offline optimal algorithm. The* ***CVaR_δ-Competitive Ratio (δ-CR)*** *is defined as the worst-case ratio between the* CVaR_δ *of* ALG*'s cost and the offline optimal cost:*

$$\delta\text{-CR}(\textsc{Alg}) := \sup_{I \in \mathcal{I}} \frac{\mathrm{CVaR}_\delta[\textsc{Alg}(I)]}{\textsc{Opt}(I)},$$

where the CVaR_δ *is taken over* ALG*'s randomness.*

It is immediately clear that any deterministic algorithm has δ-CR $=$ CR for all $\delta \in [0, 1]$, while for randomized algorithms these metrics will generally differ for $\delta > 0$. Note that, given the definition of CVaR_δ as focusing on the *worst* $(1 - \delta)$-fraction of a distribution, the δ-CR may also be interpreted as a metric that gives the adversary additional power to shift the distribution of the algorithm's randomness. Under this interpretation, the δ-CR may be viewed as an interpolation between the classic randomized case where the adversary has no power over ALG's randomness ($\delta = 0$), and the case where the adversary has full control over ALG's randomness and determinism is optimal ($\delta = 1$). This model can also be seen as a complement to the oblivious adversary, which knows ALG but cannot see the realization of its randomness, and the adaptive adversary, which sees all random outcomes; in the δ-CR case, while the adversary does not see ALG's random outcome directly, it has the ability to control this outcome in a way limited by the CVaR.

Online Problems Studied

We now provide a brief introduction for each of the three problems we study in this work.

Continuous-Time Ski Rental In the *continuous-time ski rental (CSR)* problem, a player faces a ski season of unknown and adversarially-chosen duration $s \in \mathbb{R}_{++}$, and must choose how long to rent skis before purchasing them. In particular, the player pays cost equal to the duration of renting, and cost B for purchasing the skis. Deterministic algorithms for ski rental are wholly determined by the day $x \in \mathbb{R}_{++}$ on which the player stops renting and purchases the skis: an algorithm that rents until day x and then purchases pays cost $s \cdot \mathbb{1}_{x>s} + (x+B) \cdot \mathbb{1}_{x \leq s}$. Randomized algorithms can be described by a random variable X over purchase days, in which case the algorithm pays (random) cost $s \cdot \mathbb{1}_{X>s} + (X+B) \cdot \mathbb{1}_{X \leq s}$. Given knowledge of the total number of skiing days s, the offline optimal strategy is to rent for the entire season if $s < B$, incurring cost s, and to buy immediately otherwise, yielding cost B. Defining $\alpha_\delta^{\mathrm{CSR},\mu}$ as the δ-CR of a strategy $X \sim \mu$, we have

$$\alpha_\delta^{\mathrm{CSR},\mu} := \sup_{s \in \mathbb{R}_{++}} \alpha_\delta^{\mathrm{CSR},\mu}(s) := \sup_{s \in \mathbb{R}_{++}} \frac{\mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X>s} + (X+B) \cdot \mathbb{1}_{X \leq s}]}{\min\{s, B\}},$$

where $\alpha_\delta^{\mathrm{CSR},\mu}(s)$ denotes the competitive ratio of the strategy μ when the adversary's decision is s. We denote by $\alpha_\delta^{\mathrm{CSR},*}$ the smallest δ-CR of any strategy. We will omit the "CSR" in the superscript when it is clear through context that we are discussing the continuous-time ski rental problem.

We will assume without loss of generality that $B = 1$. It is well known that $\alpha_1^{\mathrm{CSR},*} = 2$, which is achieved by purchasing skis deterministically at time 1, and $\alpha_0^{\mathrm{CSR},*} = \frac{e}{e-1}$, which is achieved by a probability density supported on the interval $[0,1]$ [231, 263]. In the following lemma, which is proved in Section 6.A, we show that when considering δ-CR as a performance metric with general $\delta \in [0,1]$, we may similarly restrict our focus to probability measures with support on $[0,1]$.

Lemma 6.2.4. *Let μ_1 be a distribution on $\mathbb{R}_+$. There is a distribution μ_2 with support in $[0,1]$ such that, for any $\delta \in [0,1]$, μ_2 has no worse δ-CR than μ_1: $\alpha_\delta^{CSR,\mu_2} \leq \alpha_\delta^{CSR,\mu_1}$.*

An important consequence of the preceding lemma is that we can restrict the adversary's decisions to $s \in (0,1]$, since choosing $s > 1$ will not change the δ-CR for any random strategy supported on $[0,1]$. Thus for μ supported in $[0,1]$, we have $\alpha_\delta^{\mathrm{CSR},\mu} = \sup_{s \in (0,1]} \alpha_\delta^{\mathrm{CSR},\mu}(s)$.

Discrete-Time Ski Rental In the *discrete-time ski rental (DSR)* problem, a player faces a ski season of unknown and adversarially-chosen duration $s \in \mathbb{N}$ and must choose an integer number of days to rent skis before purchasing them; renting for a day costs 1, and purchasing skis has an integer cost $B \geq 2$. The cost structure is essentially identical to the continuous-time case, except the algorithm's and adversary's decisions are restricted to lie in $\mathbb{N}$: if a player buys skis at the start of day $x \in \mathbb{N}$ and the true season duration is $s \in \mathbb{N}$, their cost will be $s \cdot \mathbb{1}_{x>s} + (B+x-1) \cdot \mathbb{1}_{x \le s}$. Thus for a random strategy $X \sim \mu$ with support on $\mathbb{N}$, the δ-CR is defined as follows:

$$\alpha_\delta^{\mathrm{DSR}(B),\mu} := \sup_{s \in \mathbb{N}} \alpha_\delta^{\mathrm{DSR}(B),\mu}(s) := \sup_{s \in \mathbb{N}} \frac{\mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X>s} + (B+X-1) \cdot \mathbb{1}_{X \le s}]}{\min\{s, B\}}.$$

As in the continuous-time setting, we denote by $\alpha_\delta^{\mathrm{DSR}(B),*}$ the smallest δ-CR of any strategy, and will omit the "DSR" from the superscript when it is clear from context, instead writing just $\alpha_\delta^{B,\mu}$. It is well known that $\alpha_1^{B,*} = 2 - \frac{1}{B}$, achieved by deterministically purchasing skis at the start of day B, and $\alpha_0^{B,*} = \frac{1}{1-\left(1-B^{-1}\right)^B}$, which approaches $\alpha_0^{\mathrm{CSR},*} = \frac{e}{e-1}$ as $B \to \infty$. Following identical reasoning as in Lemma 6.2.4 for the continuous-time setting, we may without loss of generality restrict our focus to strategies μ with support on $[B]$, and likewise to adversary decisions in $[B]$. Finally, note that the discrete problem is easier than the continuous-time problem, i.e., $\alpha_\delta^{\mathrm{DSR}(B),*} \le \alpha_\delta^{\mathrm{CSR},*}$ for all $\delta \in [0,1]$ and $B \in \mathbb{N}$; this is because we can embed DSR into the continuous setting by restricting the continuous-time adversary to choose season durations $\{\frac{1}{B}, \dots, \frac{B-1}{B}, 1\}$ and reducing the player's buying cost by $\frac{1}{B}$.

One-Max Search In the *one-max search (OMS)* problem, a player faces a sequence of prices $v_t \in [L, U]$ arriving online, with $U \ge L > 0$ known upper and lower bounds on the price sequence; we define the *fluctuation ratio* $\theta = \frac{U}{L}$ as the ratio between these bounds. The player's goal is to sell an indivisible item for the greatest possible price: after observing a price v_t, the player can choose to either accept the price and earn profit v_t, or to wait and observe the next price. The duration $T \in \mathbb{N}$ of the sequence is *a priori* unknown to the player, and if T elapses and the player has not yet sold the item, they sell it for the smallest possible price L in a compulsory trade. In the deterministic setting, the player aims to minimize their competitive ratio, defined as the worst-case ratio between the price accepted by the player and the optimal price $v_{\max} = \max_t v_t$:

$$\mathrm{CR}(\textsc{Alg}) := \sup_{(v_1,\dots,v_T) \in [L,U]^T} \frac{\textsc{Opt}(v_1,\dots,v_T)}{\textsc{Alg}(v_1,\dots,v_T)} = \sup_{(v_1,\dots,v_T) \in [L,U]^T} \frac{v_{\max}}{\textsc{Alg}(v_1,\dots,v_T)},$$

with an expectation around $\textsc{Alg}$ in the denominator if the algorithm is randomized. Note that this definition of competitive ratio differs from that in Definition 6.2.2 because this is a reward maximization, rather than a loss minimization, problem. Likewise, when discussing the conditional value-at-risk and δ-CR in this setting, we will use the reward formulation, which is the expected reward on the worst (i.e., smallest) $(1-\delta)$-fraction of outcomes in the reward distribution [239]:

$$\mathrm{CVaR}_\delta[X] = \sup_{t\in\mathbb{R}} \left\{ t - \frac{1}{1-\delta}\mathbb{E}[t-X]^+ \right\} = \frac{1}{1-\delta}\int_0^{1-\delta} F_X^{-1}(p)\,\mathrm{d}p. \tag{6.2}$$

While these definitions of CVaR_δ and CR differ from those employed in discussion of the ski rental problem, we will generally not distinguish which version we are using throughout this chapter, as it will be clear from context which problem (and hence which version) we are concerned with.

The one-max search problem was first studied in [78], which found that the optimal deterministic competitive ratio is $\sqrt{\theta}$, achieved by a "reservation price" or "threshold" [167] algorithm that accepts the first price above $\sqrt{LU}$. Randomization improves the competitive ratio exponentially: the optimal randomized competitive ratio is $1 + W_0\left(\frac{\theta-1}{e}\right) = \Theta(\log\theta)$, where W_0 is the principal branch of the Lambert W function [78, 79]. In this work, we restrict our focus to the class of *random threshold algorithms* without loss of generality;[1] such algorithms fix a distribution μ with support on $[L, U]$, draw a threshold $X \sim \mu$ at random, and accept the first price above X, earning profit $L\cdot\mathbb{1}_{X>v_{\max}} + X\cdot\mathbb{1}_{X\le v_{\max}}$.[2] Thus the δ-CR of a threshold algorithm is defined:

$$\alpha_\delta^{\mathrm{OMS}(\theta),\mu} := \sup_{v\in[L,U]} \alpha_\delta^{\mathrm{OMS}(\theta),\mu}(v) := \sup_{v\in[L,U]} \frac{v}{\mathrm{CVaR}_\delta[L\cdot\mathbb{1}_{X>v} + X\cdot\mathbb{1}_{X\le v}]}, \tag{6.3}$$

where we denote by $\alpha_\delta^{\mathrm{OMS}(\theta),\mu}(v)$ the δ-CR of one-max search with fluctuation ratio θ restricted to price sequences with maximal price v, which is wholly determined by the distribution of the random threshold X. As in the ski rental problems, we denote by $\alpha_\delta^{\mathrm{OMS}(\theta),*}$ the optimal δ-CR for the problem, and we omit "OMS" from the superscript when the problem is clear from context.

6.3 CVaR_δ-Competitive Continuous-Time Ski Rental: Optimal Algorithm and Lower Bound

As noted in the previous section, the optimal deterministic competitive ratio for continuous-time ski rental is $\alpha_1^* = 2$, and the optimal randomized competitive ratio

[1] This restriction is made without loss of generality, in the sense that any randomized algorithm for OMS with δ-CR α can be approximated by a random threshold policy with δ-CR $\alpha+\epsilon$, with ϵ arbitrarily small; see Section 6.A for a full explanation.

[2] When the player uses a random threshold algorithm, we may assume that they earn profit exactly X whenever $X \le v_{\max}$, since if the adversary (who is unaware of X) plays a sequence of prices that increases by ϵ at every time until reaching $v_{\max}$, then the player will accept the first price above X, which will be at most $X+\epsilon$; sending $\epsilon \to 0$, the player's profit is exactly X.

is $\alpha_0^* = \frac{e}{e-1}$. This immediately motivates the question of what the optimal δ-CR is, for arbitrary $\delta \in (0,1)$: how does α_δ^* grow as $\delta \uparrow 1$? And does α_δ^* strictly improve on the deterministic worst case of 2 whenever $\delta < 1$?

The classical approach for obtaining the optimal randomized algorithm for continuous-time ski rental is to assume that the optimal purchase distribution has a probability density p supported on $[0,1]$, use this to express the expected cost of the algorithm given any adversary decision, and write out the inequalities that must be satisfied for the algorithm to be α-competitive for some constant α [231]:

$$\int_0^s (t+1)p(t)\,\mathrm{d}t + s\int_s^1 p(t)\,\mathrm{d}t \le \alpha s \quad \text{for all } s \in [0,1].$$

The optimal p is found by setting these inequalities to equalities, differentiating with respect to s, solving the resulting differential equations, and choosing α to ensure p integrates to 1. If we attempt to apply this methodology to the problem with the δ-CR objective, we are met with two challenges: first, while the assumption that the optimal strategy has a density and the trick of setting the above inequalities to equalities works in the expected cost setting, there is no guarantee that these assumptions can be imposed without loss of generality when the expectation is replaced with CVaR_δ. Second, and more formidably, even if we can restrict to densities, the limits of integration in the CVaR_δ case will depend on the particular quantile structure induced by p. If $X \sim p$ and $F_X(s) \ge 1-\delta$, using the definition of CVaR_δ as the expected cost on the worst $(1-\delta)$-sized subpopulation of the loss, one can compute

$$\mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X>s} + (X+1)\cdot \mathbb{1}_{X\le s}] = \int_{F_X^{-1}(F_X(s)-(1-\delta))}^{s} (t+1)p(t)\,\mathrm{d}t,$$

whose lower limit of integration depends on p's quantile structure in a nontrivial way (i.e., it is the smallest point with CDF value equal to $F_X(s)-(1-\delta)$), significantly complicating the formulation of any differential equation we could construct using this expression.

Not all is lost, however: if we instead take inspiration from the formulation of CVaR_δ in terms of the inverse CDF of the loss distribution, it is possible to formulate the CVaR_δ of the loss of an *arbitrary* strategy X (i.e., not necessarily one with a density) in terms of the inverse CDF of X. We state this result formally in the following lemma, which is proved in Section 6.B.

Lemma 6.3.1. *Let X be a random variable supported in $[0,1]$, and fix an adversary decision $s \in (0,1]$. Then the* CVaR_δ *of the cost incurred by the algorithm playing*

X is

$$\mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X>s} + (X+1) \cdot \mathbb{1}_{X\le s}] \\ = \begin{cases} \frac{1}{1-\delta}\Big[(1-\delta-F_X(s))s + \int_0^{F_X(s)}(1+F_X^{-1}(t))\,\mathrm{d}t\Big] & \text{if } F_X(s) \le 1-\delta \\ \frac{1}{1-\delta}\int_{F_X(s)-(1-\delta)}^{F_X(s)}(1+F_X^{-1}(t))\,\mathrm{d}t & \text{otherwise.} \end{cases}$$

While the integral representation of the algorithm's cost given in Lemma 6.3.1 depends on both the CDF F_X and the inverse CDF F_X^{-1}, it is possible to remove the CDF when it is continuous and strictly increasing on $[0,1]$; in this case, for any $s \in [0,1]$, we may define a corresponding $y = F_X(s)$ and replace $F_X(s)$ with y and s with $F_X^{-1}(y)$ in Lemma 6.3.1's representation. We will show later that the optimal strategy indeed has such a continuous and strictly increasing F_X (see Lemmas 6.B.7 and 6.B.8 in Section 6.B).

As a first application of the representation for the CVaR_δ-cost in Lemma 6.3.1, we construct in the following theorem a family of densities parametrized by δ whose δ-CR we can compute analytically, giving an upper bound on α_δ^*, and in particular showing that $\alpha_\delta^* < 2$ for all $\delta \in [0,1)$.

Theorem 6.3.2. *Let $p_\delta(x)$ be a probability density defined on the unit interval $[0,1]$ as*

$$p_\delta(x) = \frac{(1-\delta)(1-e^{\frac{c}{1-\delta}})}{c(e^{\frac{c}{1-\delta}}(x-1)-x)},$$

with constant $c = -\frac{1+2W_{-1}(-1/2\sqrt{e})}{2} \approx 1.25643$, where W_{-1} is the -1 branch of the Lambert W function. Then the strategy that buys on day $X \sim p_\delta$ achieves competitive ratio

$$\alpha_\delta^{p_\delta} = 2 - \frac{1}{e^{\frac{c}{1-\delta}} - 1}.$$

In particular, $\alpha_\delta^{p_\delta} < 2$ for all $\delta \in [0,1)$.

We present a proof of this theorem in Section 6.B. Our approach is to compute the inverse CDF corresponding to the proposed density and reformulate the inequalities defining $\alpha_\delta^{p_\delta}$-competitiveness using Lemma 6.3.1; the rest of the work is concerned with computing $\alpha_\delta^{p_\delta}$.

Intuitively, the strategy p_δ in Theorem 6.3.2 behaves like one might expect a good algorithm for ski rental with the δ-CR metric should: it assigns less probability mass to earlier times and more to later times, and as δ increases, it shifts mass from earlier times to later times. However, the algorithm cannot be optimal, since

$\alpha_0^{p_0} = 2 - \frac{1}{e^c - 1} \approx 1.60$, which is larger—though only slightly—than the randomized optimal $\frac{e}{e-1} \approx 1.58$. This motivates the question: is it possible to leverage the representation in Lemma 6.3.1 to obtain the optimal algorithm for continuous-time ski rental with the δ-CR objective? In the following theorem, we answer this question in the affirmative: in particular, the optimal algorithm's inverse CDF is the solution to a delay differential equation defined on the interval $[0, 1]$.

Theorem 6.3.3. *For any $\delta \in [0, 1)$, let $\phi : [0, 1] \to [0, 1]$ be the solution to the delay differential equation*

$$\phi'(t) = \frac{1}{\alpha(1-\delta)} \left[\phi(t) - \phi(t - (1-\delta))\right] \qquad \text{for } t \in [1-\delta, 1],$$

*with initial condition $\phi(t) = \log\left(1 + \frac{t}{(\alpha-1)(1-\delta)}\right)$ on $t \in [0, 1-\delta]$. Then when $\alpha = \alpha_\delta^*$, ϕ is the inverse CDF of the **unique optimal** strategy for continuous-time ski rental with the δ-*CR *metric.*

We prove this theorem in Section 6.B. The crux of the proof is a pair of structural lemmas (Lemmas 6.B.3 and 6.B.6) which establish that, for any $\delta \in [0, 1)$, the optimal algorithm μ^* is indifferent to the adversary's decision, i.e., $\alpha_\delta^{\mu^*}(s) = \alpha_\delta^*$ for all $s \in (0, 1]$. This is analogous to the trick of "setting the inequalities to equalities" in the classical version of ski rental [231], but requires a great deal more care in the continuous-time CVaR_δ setting to make rigorous. In addition, this result depends on the fact that $\alpha_\delta^* < 2$, which we showed in Theorem 6.3.2. With this property established, we can apply Lemma 6.3.1 to pose a family of integral equations constraining the optimal inverse CDF, which can be transformed to obtain the delay differential equation in Theorem 6.3.3.

Note, however, that the delay differential equation yielding the optimal inverse CDF depends on the optimal δ-CR α_δ^*, which we have no analytic form for. Fortunately, the solution ϕ to the delay differential equation in Theorem 6.3.3 has the property that $\phi(t)$ is strictly decreasing in α for each $t \in (0, 1]$ (see Section 6.B); since the optimal inverse CDF must have $\phi(1) = 1$ (see Lemma 6.B.8 in the appendix), α_δ^* is equivalently defined as the unique choice of α for which the solution to the above delay differential equation satisfies $\phi(1) = 1$. We may thus determine α_δ^* via binary search: given some α, we solve the delay differential equation numerically and evaluate $\phi(1)$; if $\phi(1) > 1$, then we decrease α, and if $\phi(1) < 1$, we increase α. We plot the optimal δ-CR obtained via this binary search methodology (with

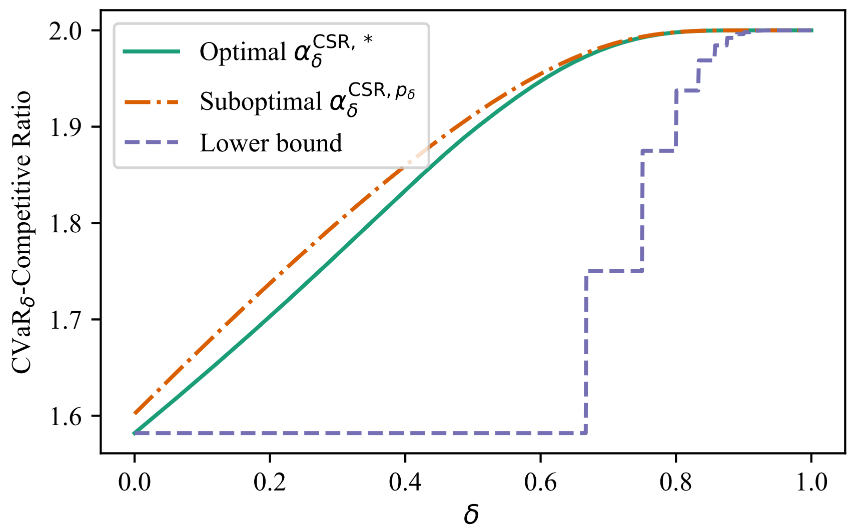


Figure 6.1: CVaR$_\delta$-competitive ratios from Theorems 6.3.2 (Suboptimal) and 6.3.3 (Optimal) and lower bound from Theorem 6.3.4 for continuous-time ski rental.

delay differential equations solved numerically in Mathematica) alongside the upper bound from Theorem 6.3.2 in Figure 6.1.

While Theorem 6.3.3 gives us a method for computing the optimal strategy and δ-CR for continuous-time ski rental, it does not give an analytic form of this solution or metric. An analytic form of $\phi(t)$ can be obtained when $\delta \leq \frac{1}{2}$, though its form is complicated and does not facilitate analysis of the optimal δ-CR in this regime (see Section 6.B). We thus conclude this section by providing a lower bound on α_δ^*, which we prove in Section 6.B.

Theorem 6.3.4. *For any $\delta \in [0, 1)$, the optimal δ-CR α_δ^* has the lower bound*

$$\alpha_\delta^* \geq \max\left\{\frac{e}{e-1}, 2 - \frac{1}{2^{\lfloor \frac{1}{1-\delta} \rfloor - 1}}\right\}.$$

We plot this lower bound in Figure 6.1 alongside the upper bound from Theorem 6.3.2 and the optimal competitive ratio. While this lower bound is vacuous for $\delta < \frac{2}{3}$, in which case it is exactly the expected cost lower bound of $\frac{e}{e-1}$, it has the same asymptotic form as the upper bound in Theorem 6.3.2 as δ approaches 1. Thus, Theorems 6.3.2 and 6.3.4 together give us that $\alpha_\delta^* = 2 - \frac{1}{2^{\Theta\left(\frac{1}{1-\delta}\right)}}$, as $\delta \uparrow 1$.

6.4 CVaR$_\delta$-Competitive Discrete-Time Ski Rental: Phase Transition and Analytic Optimal Algorithm

Having characterized the optimal algorithm and δ-CR for continuous-time ski rental in the previous section, we now turn to the discrete-time version of the problem, and ask: are there any qualitative differences between the optimal algorithm or δ-CR for the discrete problem and the continuous-time problem? And are there any regimes of δ for which we can analytically characterize the optimal algorithm? It turns out that the answer to both of these questions is yes; we begin by showing, in the following theorem, that the optimal δ-CR for the discrete-time ski rental problem exhibits a phase transition at $\delta = 1 - \Theta(\frac{1}{\log B})$ such that, for δ beyond this transition, the optimal algorithm is exactly the deterministic algorithm that buys at time B.

Theorem 6.4.1. *Let $\alpha_\delta^{B,*}$ be the optimal δ-*CR *for discrete-time ski rental with buying cost $B \in \mathbb{N}$. Then $\alpha_\delta^{B,*}$ exhibits a **phase transition** at $\delta = 1 - \Theta(\frac{1}{\log B})$, whereby before this transition, $\alpha_\delta^{B,*}$ strictly improves on the deterministic optimal δ-*CR *of $2 - \frac{1}{B}$, whereas after this transition, $\alpha_\delta^{B,*} = 2 - \frac{1}{B}$. Specifically:*

*(i) For all $\delta < 1 - \frac{c}{\log(B+1)}$, the optimal δ-*CR *is strictly bounded above by the deterministic optimal* CR*: $\alpha_\delta^{B,*} < 2 - \frac{1}{B}$ (where $c = -\frac{1+2W_{-1}(-1/2\sqrt{e})}{2} \approx 1.25643$ as in Theorem 6.3.2).*

*(ii) For all $\delta \geq 1 - \frac{1}{2\lfloor \log_2 B \rfloor + 1}$, the optimal δ-*CR *is exactly the deterministic optimal* CR*: $\alpha_\delta^{B,*} = 2 - \frac{1}{B}$. Thus, the optimal algorithm for this regime purchases deterministically at time B.*

We prove this result in Section 6.C; the proof of part (i) of follows essentially immediately from our analytic upper bound for the continuous setting (Theorem 6.3.2), and the proof of part (ii) is adapted from that for the continuous-time lower bound (Theorem 6.3.4) in order to handle the discrete nature of the problem. Note that this phase transition behavior is in sharp contrast to the behavior of the optimal δ-CR and algorithm in the continuous time setting: whereas in continuous time, $\alpha_\delta^{\mathrm{CSR},*}$ strictly improves on the deterministic optimal for all $\delta < 1$, in discrete time, $\alpha_\delta^{\mathrm{DSR}(B),*}$ is equal to the deterministic optimal for a non-degenerate interval of δ, implying a limit to the benefit of randomization in the risk-sensitive setting. In addition, this phase transition result gives an analytic solution for the algorithm with optimal δ-CR when δ is sufficiently large; a natural, complementary question is whether it is possible to obtain an analytic solution for the optimal algorithm with smaller δ. We prove in the next theorem that such a solution can be obtained when $\delta = O(\frac{1}{B})$.

Theorem 6.4.2. *Suppose* $\delta \leq \left(\frac{B-1}{B}\right)^{B-1} \frac{(1-(1-1/B)^B)^{-1}}{B} = O(\frac{1}{B})$. *Then the optimal* δ-CR $\alpha_\delta^{B,*}$ *and strategy* $\mathbf{p}^{B,\delta,*}$ *for discrete-time ski rental with buying cost* B *are*

$$\alpha_\delta^{B,*} = \frac{C-\delta}{1-\delta} \qquad \text{and} \qquad p_i^{B,\delta,*} = \frac{C}{B}\left(1-\frac{1}{B}\right)^{B-i} \quad \text{for all } i \in [B],$$

where $C = \frac{1}{1-(1-1/B)^B}$ *is the optimal competitive ratio for the* $\delta = 0$ *case. In particular,* $\mathbf{p}^{B,\delta,*}$ *is constant as a function of* δ, *and is identical to the optimal algorithm for the expected cost setting.*

We prove this result in Section 6.C; the proof follows a similar strategy to the proof characterizing the optimal strategy in the continuous-time setting, and in particular involves the proof of several technical lemmas that, similar to Lemmas 6.B.3 and 6.B.6 in the continuous-time setting, characterize the optimal algorithm $\mathbf{p}^{B,\delta,*}$ via the adversary's indifference to its chosen ski season duration. As a consequence of this theorem, we can analytically obtain the optimal algorithm for discrete-time ski rental with the δ-CR objective whenever $\delta = O(\frac{1}{B})$, and the corresponding δ-CR is a rational function of δ. We anticipate that extensions of this result may be possible for larger δ, but in general the optimal δ-CR will be a piecewise function of δ whose pieces, including the number of pieces and the intervals they are defined on, will depend on B, so we leave the problem of characterizing the δ-CR for all δ and general B to future work. However, if computational results suffice, an adapted form of the binary search approach employed in [253, Appendix E] can be used in tandem with a linear programming formulation of the CVaR in order to approximate the optimal solution for any δ with $\alpha_\delta^{B,*} < 2 - \frac{1}{B}$.

6.5 CVaR_δ-Competitive One-Max Search: Asymptotically Optimal Algorithm and Phase Transition

We now turn our focus to the one-max search problem. As noted in Section 6.2, existing results for this problem in the deterministic and randomized settings have established that the optimal deterministic competitive ratio is $\alpha_1^{\theta,*} = \sqrt{\theta}$ and the optimal randomized competitive ratio is $\alpha_0^{\theta,*} = 1 + W_0\left(\frac{\theta-1}{e}\right) = \Theta(\log\theta)$, where $\theta = \frac{U}{L}$ is the fluctuation ratio. We seek to obtain an upper bound on the δ-CR for more general δ; to this end, we prove a lemma that, in an analogous fashion to Lemma 6.3.1 for the continuous-time ski rental problem, leverages the integral form of the conditional value-at risk to let us express the CVaR_δ-reward of a particular randomized threshold algorithm X in terms of the inverse CDF of X.

Lemma 6.5.1. *Let X be a random variable supported in $[L, U]$, and fix an adversary choice of the maximal price $v \in [L, U]$. Then the* CVaR_δ *of the profit earned by the algorithm playing the random threshold X is*

$$\begin{aligned}
&\mathrm{CVaR}_\delta \left[L \cdot \mathbb{1}_{X > v} + X \cdot \mathbb{1}_{X \leq v} \right] \\
&\quad = \begin{cases} L & \text{if } F_X(v) \leq \delta \\ \frac{1}{1-\delta} \left[(1 - F_X(v))L + \int_0^{F_X(v)-\delta} F_X^{-1}(t) \, \mathrm{d}t \right] & \text{otherwise.} \end{cases}
\end{aligned}$$

We prove this lemma in Section 6.D. While the representation of the CVaR_δ of profit in this case differs substantially from the cost representation for ski rental in Lemma 6.3.1, it nonetheless also has a relatively simple parametrization in terms of the inverse CDF of the decision X, which will facilitate algorithm design. This is due, in part, to the piecewise linear structure exhibited by the cost/profit in these problems, and we anticipate that extending our results to online problems with more general classes of piecewise linear costs and rewards may be a fruitful avenue for future work.

While the representation of the CVaR_δ-reward in Lemma 6.5.1 depends on both the CDF and the inverse CDF of X, we can eliminate the CDF so long as the maximal price $v \in F_X^{-1}([0, 1])$. Using this fact, we prove the following theorem, proposing an algorithm and establishing an upper bound on the δ-CR for all $\delta \in [0, 1]$. We prove the result in Section 6.D.

Theorem 6.5.2. *Let $\delta \in [0, 1]$, and let $\phi : [0, 1] \to [L, U]$ be the solution to the following delay differential equation:*

$$\phi'(t) = \frac{\alpha_\delta^\theta}{1 - \delta} \left[\phi(t - \delta) - L \right] \qquad \text{for } t \in [\delta, 1], \tag{6.4}$$

with initial condition $\phi(t) = \alpha_\delta^\theta L$ on $t \in [0, \delta]$, where α_δ^θ is chosen such that $\phi(1) = U$ when $\delta < 1$, and $\alpha_\delta^\theta := \sqrt{\theta}$ when $\delta = 1$. Then ϕ is the inverse CDF of a random threshold algorithm for one-max search with δ-CR α_δ^θ. Moreover, α_δ^θ is bounded above by the unique positive solution $\overline{r}(\delta)$ to the equation

$$(\overline{r}(\delta) - 1) \left(1 + \frac{\overline{r}(\delta)}{\overline{n}(\delta)} \right)^{\overline{n}(\delta)} = \theta - 1, \tag{6.5}$$

where $\overline{n}(\delta) = \max \left\{ 1, \left\lfloor \left(\lfloor \delta^{-1} \rfloor - 1 \right) / 2 \right\rfloor \right\}$, with the $\delta = 0$ case defined by taking $\delta \downarrow 0$. In particular,

$$\alpha_\delta^\theta \leq \begin{cases} 1 + W_0 \left(\frac{\theta - 1}{e} \right) + O(\delta) & \text{as } \delta \downarrow 0 \\ \sqrt{\theta} & \text{when } \delta > \frac{1}{5}, \end{cases} \tag{6.6}$$

with the equality $\alpha_\delta^\theta = \sqrt{\theta}$ *when* $\delta \geq \frac{1}{2}$*, where the asymptotic notation omits dependence on* θ*.*

We make three brief remarks concerning this result. First, note that the proposed algorithm merely gives an upper bound on the δ-CR of one-max search and might not be optimal, although its δ-CR matches the optimal randomized and deterministic algorithms in the $\delta = 0$ and 1 cases. Second, when $\delta \in [0, 1)$, it is possible to analytically solve the delay differential equation (6.4) by integrating step-by-step (see Section 6.D):

$$\phi(t) = L + (\alpha_\delta^\theta - 1)L \sum_{j=0}^{\infty} \frac{(\alpha_\delta^\theta)^j ([t - j\delta]^+)^j}{(1-\delta)^j j!}. \tag{6.7}$$

When $\delta = 0$, (6.7) simplifies to $\phi(t) = L + (\alpha_\delta^\theta - 1)Le^{\alpha_\delta^\theta t}$, the optimal randomized algorithm [176]. On the other hand, when $\delta \in (0, 1)$, all terms with $j \geq \lceil \delta^{-1} \rceil$ disappear for $t \in [0, 1]$, so $\phi(t)$ is a continuous, piecewise polynomial function. In either case, $\phi(1)$ is strictly increasing in $\alpha_\delta^\theta > 0$, so the δ-CR α_δ^θ can be obtained numerically by solving $\phi(1) = U$ via standard root-finding methods.

Finally, when $\delta \geq \frac{1}{2}$, Theorem 6.5.2 asserts that $\alpha_\delta^\theta = \sqrt{\theta}$, which is identical to the optimal deterministic competitive ratio. This raises the question: can *any* algorithm improve upon the deterministic bound when $\delta \geq \frac{1}{2}$, or is this behavior reflective of a phase transition at $\delta = \frac{1}{2}$ such that randomness cannot improve performance when δ is greater than this level? In the following result, which we prove in Section 6.D by leveraging connections with the k-max search problem [79], we provide a lower bound establishing that the latter case is true, and that moreover, the algorithm in Theorem 6.5.2 is asymptotically optimal for small δ.

Theorem 6.5.3. *Fix* $\delta \in [0, 1]$*, let* $\alpha_\delta^{\theta,*}$ *be the optimal* δ-CR *for one-max search, and define* $\underline{r}(\delta)$ *to be the unique positive solution to the equation*

$$(\underline{r}(\delta) - 1)\left(1 + \frac{\underline{r}(\delta)}{\underline{n}(\delta)}\right)^{\underline{n}(\delta)} = \theta - 1, \tag{6.8}$$

where $\underline{n}(\delta) = \max\left\{1, \lceil \delta^{-1} \rceil - 1\right\}$*, with the* $\delta = 0$ *case defined by taking* $\delta \downarrow 0$*. Then* $\alpha_\delta^{\theta,*} \geq \underline{r}(\delta)$*; in particular,*

$$\alpha_\delta^{\theta,*} \geq \begin{cases} 1 + W_0\left(\frac{\theta-1}{e}\right) + \Omega(\delta) & \text{as } \delta \downarrow 0 \\ \sqrt{\theta} & \text{when } \delta \geq \frac{1}{2}, \end{cases} \tag{6.9}$$

where the asymptotic notation omits dependence on θ*.*

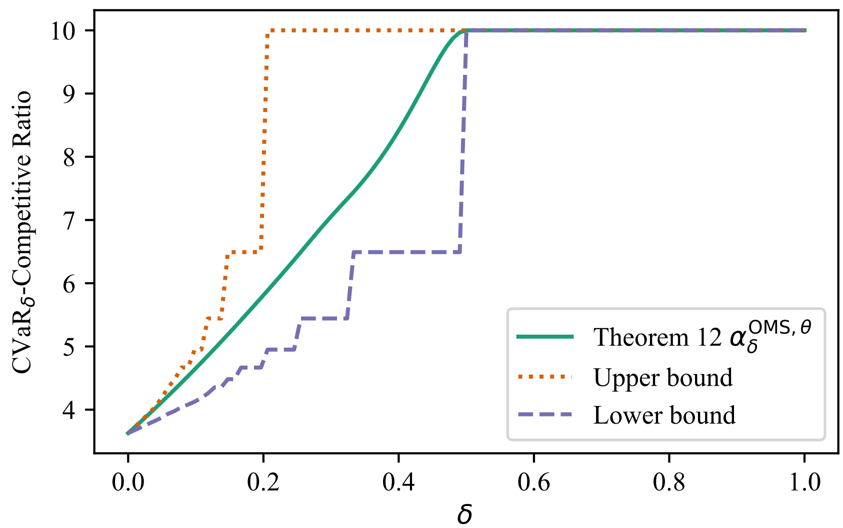


Figure 6.2: CVaR$_\delta$-competitive ratio of the algorithm in Theorem 6.5.2 along with the upper bound (6.5) and lower bound (6.8) for one-max search.

Thus, in contrast to the continuous-time ski rental problem, which exhibited no phase transition in its competitive ratio, and the discrete-time ski rental problem, which had a phase transition that shrank as $B \to \infty$, Theorem 6.5.3 establishes that the one-max search problem has a phase transition at $\delta = \frac{1}{2}$ that remains present even as $\theta \to \infty$. As such, there is a significant limit to the power of randomization in risk-sensitive one-max search. In addition, note that the form of the implicit lower bound (6.8) matches that of the upper bound (6.5), aside from the definitions of the functions $\underline{n}(\delta)$ and $\overline{n}(\delta)$. This suggests that our upper and lower bounds are tight up to the choice of the function $n(\delta) = \Theta(\delta^{-1})$. In particular, this tightness is made clear in the analytic bounds (6.6) and (6.9) in the $\delta \downarrow 0$ limit, which indicate that our algorithm is asymptotically optimal when δ is small. We plot the numerically obtained δ-CR α_δ^θ together with the upper and lower bounds (6.5) and (6.8) in Figure 6.2 when $L = 1$ and $U = 100$, which confirms the near-tightness of the bounds and the phase transition at $\delta = \frac{1}{2}$.

6.6 Conclusion

In this chapter, we consider the problem of designing *risk-sensitive* online algorithms, with performance evaluated via a competitive ratio metric—the δ-CR—defined using the conditional value-at-risk of an algorithm's cost. We consider the continuous- and discrete-time ski rental problems as well as the one-max search problem, obtaining optimal (and suboptimal) algorithms, lower bounds, and analytic characterizations of phase transitions in the optimal δ-CR for discrete-time ski rental and one-max search. Our work motivates many interesting new directions, including (a) obtaining an exact or asymptotic analytic form of the optimal δ-CR for discrete-time ski rental and one-max search across all δ, (b) the design and analysis of risk-sensitive algorithms for online problems with more general classes of cost and reward functions, or more complex problems such as metrical task systems, (c) exploring the use of alternative risk measures in place of the conditional value-at-risk, and (d) exploring potential connections between risk-sensitive online algorithms and robustness to distribution shift in learning-augmented online algorithms, drawing motivation from the framing of CVaR_δ in terms of distribution shift.

Appendix

In these appendix sections, we present proofs of the theoretical results in the main body of the chapter.

6.A Additional Details for Section 6.2

Proof of Lemma 6.2.4

Let $X_1 \sim \mu_1$, and define another random variable $X_2 \sim \mu_2$ with support on $[0, 1]$ as

$$X_2 = \begin{cases} X_1 & \text{if } X_1 \leq 1 \\ 1 & \text{otherwise.} \end{cases}$$

Suppose $s \in [0, 1)$. Clearly $\mathbb{1}_{X_1 > s} = \mathbb{1}_{X_2 > s}$ and $\mathbb{1}_{X_1 \leq s} = \mathbb{1}_{X_2 \leq s}$, and since $X_2 = X_1$ when $X_1 \leq 1$, $X_1 \cdot \mathbb{1}_{X_1 \leq s} = X_2 \cdot \mathbb{1}_{X_2 \leq s}$ for $s < 1$. Thus

$$\begin{aligned} \alpha_\delta^{\mu_1}(s) &= \frac{\mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X_1 > s} + (X_1 + 1) \cdot \mathbb{1}_{X_1 \leq s}]}{\min\{s, 1\}} \\ &= \frac{\mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X_2 > s} + (X_2 + 1) \cdot \mathbb{1}_{X_2 \leq s}]}{\min\{s, 1\}} \\ &= \alpha_\delta^{\mu_2}(s). \end{aligned} \tag{6.10}$$

Now, consider the case $s \in [1, +\infty]$. By construction, $X_2 \leq 1$, so $\mathbb{1}_{X_2 > s} = 0$ and $\mathbb{1}_{X_2 \leq s} = 1$, and thus

$$\alpha_\delta^{\mu_2}(s) = \frac{\mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X_2 > s} + (X_2 + 1) \cdot \mathbb{1}_{X_2 \leq s}]}{\min\{s, 1\}} \tag{6.11}$$

$$= 1 + \mathrm{CVaR}_\delta[X_2] \tag{6.12}$$

where (6.11) uses translation invariance of CVaR. On the other hand, we have

$$\begin{aligned}
\alpha_\delta^{\mu_1}(s) &= \frac{\mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X_1>s} + (X_1+1)\cdot \mathbb{1}_{X_1\le s}]}{\min\{s,1\}} \\
&\geq \mathrm{CVaR}_\delta[1 \cdot \mathbb{1}_{X_1>s} + (X_1+1)\cdot \mathbb{1}_{X_1\le s}] && (6.13)\\
&= 1 + \mathrm{CVaR}_\delta[X_1 \cdot \mathbb{1}_{X_1\le s}] && (6.14)\\
&\geq 1 + \mathrm{CVaR}_\delta[X_2], && (6.15)
\end{aligned}$$

where (6.13) follows by monotonicity of CVaR, (6.14) follows by translation invariance, and (6.15) follows by monotonicity when $s = +\infty$ (in which case $\mathbb{1}_{X_1\le s} = 1$). Combining (6.10), (6.12), and (6.15), we obtain $\alpha_\delta^{\mu_2} \le \alpha_\delta^{\mu_1}$, as claimed. □

On the Restriction to Random Threshold Policies

In this section, we briefly justify the claim that the restriction to random threshold policies for one-max search is made without loss of generality. Let ALG be an arbitrary randomized algorithm for one-max search. Following the argument in the proof of [78, Theorem 1], the lack of memory restrictions in this problem implies, by Kuhn's Theorem, that ALG is, without loss of generality, a mixed strategy, or a probability distribution over deterministic algorithms [264]. For some $k \in \mathbb{N}$, let $\epsilon = \frac{U-L}{k}$, and define a restricted set of adversary price sequences $\mathcal{I}_\epsilon$ as

$$\mathcal{I}_\epsilon = \{\mathbf{v} : \mathbf{v} = (L, L+\epsilon, \ldots, L+n\epsilon), n \in \{0,\ldots,k\}\},$$

i.e., $\mathcal{I}_\epsilon$ is the set of all price sequences that begin at L and increase by ϵ at each time. If the adversary is restricted to choosing price sequences in $\mathcal{I}_\epsilon$, then any deterministic algorithm is equivalent in behavior to some deterministic threshold algorithm. To see why, note that $\mathcal{I}_\epsilon$ comprises $k+1$ price sequences, each of a unique length in $[k+1]$; we will call $\mathbf{v}_n$ the sequence of length $n \in [k+1]$. Moreover, $\mathbf{v}_n$ constitutes the first n entries of $\mathbf{v}_{n+1}$. As such, the behavior of a deterministic algorithm $\overline{\text{ALG}}$ on $\mathbf{v}_n$ will be identical to its behavior on the first n prices revealed in $\mathbf{v}_{n+1}$; in particular, if $\overline{\text{ALG}}$ sells at time $j < n$ in $\mathbf{v}_n$, it will do the same in $\mathbf{v}_{n+1}$ and earn the same profit $L + (j-1)\epsilon$. As a result, $\overline{\text{ALG}}$'s behavior on $\mathcal{I}_\epsilon$ is wholly determined by the price at which it chooses to sell, which will be consistent across price sequences in this set; in other words, $\overline{\text{ALG}}$ is equivalent to a deterministic threshold algorithm, with threshold chosen amongst the $k+1$ choices $\{v : v = L + n\epsilon, n \in \{0,\ldots,k\}\}$.[3] Thus, on $\mathcal{I}_\epsilon$, the mixed strategy ALG is equivalent

[3] If $\overline{\text{ALG}}$ never sells on any of the sequences in $\mathcal{I}_\epsilon$, we can choose a corresponding deterministic threshold of U, which obtains performance at least as good.

to a distribution over such threshold algorithms, i.e., a random threshold algorithm $X \sim \mu$ with support on $\{v : v = L + n\epsilon, n \in \{0, \dots, k\}\}$. Formally, we have

$$\begin{aligned}
\alpha_\delta^{\textsc{Alg}} &= \sup_{\substack{\mathbf{v}\in[L,U]^T,\\ T\in\mathbb{N}}} \frac{v_{\max}}{\mathrm{CVaR}_\delta[\textsc{Alg}(\mathbf{v})]} \\
&\geq \max_{\mathbf{v}\in\mathcal{I}_\epsilon} \frac{v_{\max}}{\mathrm{CVaR}_\delta[\textsc{Alg}(\mathbf{v})]} \\
&\geq \max_{\mathbf{v}\in\mathcal{I}_\epsilon} \frac{v_{\max}}{\mathrm{CVaR}_\delta[L\cdot\mathbb{1}_{X>v_{\max}} + X\cdot\mathbb{1}_{X\leq v_{\max}}]} \\
&= \max_{\substack{v=L+n\epsilon,\\ n\in\{0,\dots,k\}}} \frac{v}{\mathrm{CVaR}_\delta[L\cdot\mathbb{1}_{X>v} + X\cdot\mathbb{1}_{X\leq v}]} \qquad (6.16)
\end{aligned}$$

where $\textsc{Alg}(\mathbf{v})$ denotes the (random) profit of $\textsc{Alg}$ on the price sequence $\mathbf{v}$, $v_{\max} := \max_j v_j$, and (6.16) holds by the construction of $\mathcal{I}_\epsilon$. Then since $X \sim \mu$ is a random threshold algorithm, its δ-CR (with unrestricted adversary) is defined as in (6.3):

$$\begin{aligned}
\alpha_\delta^{\mu} &= \sup_{v\in[L,U]} \frac{v}{\mathrm{CVaR}_\delta[L\cdot\mathbb{1}_{X>v} + X\cdot\mathbb{1}_{X\leq v}]} \\
&\leq \max_{n\in\{0,\dots,k\}} \sup_{v\in[L+n\epsilon,\, L+(n+1)\epsilon)} \frac{v}{\mathrm{CVaR}_\delta[L\cdot\mathbb{1}_{X>v} + X\cdot\mathbb{1}_{X\leq v}]} \\
&\leq \max_{n\in\{0,\dots,k\}} \frac{L+(n+1)\epsilon}{\mathrm{CVaR}_\delta[L\cdot\mathbb{1}_{X>L+n\epsilon} + X\cdot\mathbb{1}_{X\leq L+n\epsilon}]} \qquad (6.17) \\
&\leq \max_{\substack{v=L+n\epsilon,\\ n\in\{0,\dots,k\}}} \frac{v}{\mathrm{CVaR}_\delta[L\cdot\mathbb{1}_{X>v} + X\cdot\mathbb{1}_{X\leq v}]} + \frac{\epsilon}{L} \qquad (6.18) \\
&\leq \alpha_\delta^{\textsc{Alg}} + \frac{\epsilon}{L} \qquad (6.19)
\end{aligned}$$

where the inequality (6.17) holds due to X having support restricted to $\{v : v = L + n\epsilon, n \in \{0, \dots, k\}\}$, which implies that $\mathrm{CVaR}_\delta[L\cdot\mathbb{1}_{X>v} + X\cdot\mathbb{1}_{X\leq v}]$ is equal to the $v = L+n\epsilon$ case for all $v \in [L + n\epsilon,\ L + (n+1)\epsilon)$, (6.18) follows by the fact that the algorithm's profit is lower bounded by L, and (6.19) follows by the inequality in (6.16). Thus, by selecting k arbitrarily large (i.e., ϵ arbitrarily small), the random threshold algorithm X can be made to have δ-CR arbitrarily close to the original randomized algorithm $\textsc{Alg}$.

6.B Proofs and Additional Results for Section 6.3

Proof of Lemma 6.3.1

Before proving the result, we first prove a general lemma that allows for writing an algorithm's CVaR_δ-cost given a particular adversary's decision $s \in [0, 1]$ in terms of the inverse CDF of the algorithm's decision.

Lemma 6.B.1. *Let X be a random variable supported in $[0,1]$, and fix an adversary's decision $s \in [0,1]$. Then the inverse CDF of the ski rental cost given by the random variable $C(X,s) = s \cdot \mathbb{1}_{X>s} + (X+1) \cdot \mathbb{1}_{X\leq s}$ is*

$$F^{-1}_{C(X,s)}(p) = \begin{cases} s & \text{if } p \leq 1 - F_X(s) \\ 1 + F_X^{-1}(p + F_X(s) - 1) & \text{otherwise,} \end{cases}$$

for $p \in [0,1]$.

Proof. Observe that the cost $C(X,s)$ takes value s when $X > s$, and is equal to $X+1$ otherwise, which is always at least s; as such, we can easily compute its CDF:

$$F_{C(X,s)}(x) = \begin{cases} 0 & \text{if } x < s \\ 1 - F_X(s) & \text{if } x \in [s,1) \\ 1 - F_X(s) + F_X(x-1) & \text{if } x \in [1, 1+s]. \end{cases}$$

Note that $C(X,s)$ is supported in $[s, 1+s]$; thus, we define its inverse CDF as

$$\begin{aligned} F^{-1}_{C(X,s)}(p) &= \inf\{x \in [s,1+s] : F_{C(X,s)}(x) \geq p\} \\ &= \begin{cases} s & \text{if } p = 0 \\ s & \text{if } p \in (0, 1 - F_X(s)] \\ \inf\{x \in [1,1+s] : 1 - F_X(s) + F_X(x-1) \geq p\} & \text{otherwise} \end{cases} \\ &= \begin{cases} s & \text{if } p \leq 1 - F_X(s) \\ 1 + \inf\{x \in [0,s] : F_X(x) \geq p + F_X(s) - 1\} & \text{otherwise} \end{cases} \\ &= \begin{cases} s & \text{if } p \leq 1 - F_X(s) \\ 1 + F_X^{-1}(p + F_X(s) - 1) & \text{otherwise,} \end{cases} \end{aligned}$$

just as claimed. □

Lemma 6.3.1 now follows as a near-immediate consequence of the preceding lemma.

Proof of Lemma 6.3.1. Define $C(X,s) = s \cdot \mathbb{1}_{X>s} + (X+1) \cdot \mathbb{1}_{X\leq s}$ as the algorithm's cost given a strategy X and adversary's decision s, just as in Lemma 6.B.1. By the second definition of CVaR_δ in (6.1) expressing it as an integral of the inverse CDF, we may write $\mathrm{CVaR}_\delta[C(X,s)]$ as:

$$\mathrm{CVaR}_\delta[C(X,s)] = \frac{1}{1-\delta} \int_\delta^1 F^{-1}_{C(X,s)}(t)\, \mathrm{d}t.$$

We break into two cases. If $\delta > 1 - F_X(s)$, then by Lemma 6.B.1, $F^{-1}_{C(X,s)}(t) = 1 + F^{-1}_X(t + F_X(s) - 1)$ on the entire domain of integration, so we have

$$\begin{aligned}\mathrm{CVaR}_\delta[C(X,s)] &= \frac{1}{1-\delta}\int_\delta^1 1 + F^{-1}_X(t + F_X(s) - 1)\,\mathrm{d}t \\ &= \frac{1}{1-\delta}\int_{F_X(s)-(1-\delta)}^{F_X(s)} 1 + F^{-1}_X(t)\,\mathrm{d}t.\end{aligned}$$

On the other hand, if $\delta \leq 1 - F_X(s)$, then by Lemma 6.B.1, $F^{-1}_{C(X,s)}(t) = s$ on $[\delta, 1 - F_X(s)]$ and $F^{-1}_{C(X,s)}(t) = 1 + F^{-1}_X(t + F_X(s) - 1)$ on $[1 - F_X(s), 1]$. Thus,

$$\begin{aligned}\mathrm{CVaR}_\delta[C(X,s)] &= \frac{1}{1-\delta}\left(\int_\delta^{1-F_X(s)} s\,\mathrm{d}t + \int_{1-F_X(s)}^1 1 + F^{-1}_X(t + F_X(s) - 1)\,\mathrm{d}t\right) \\ &= \frac{1}{1-\delta}\left[(1-\delta-F_X(s))s + \int_0^{F_X(s)} 1 + F^{-1}_X(t)\,\mathrm{d}t\right].\end{aligned}$$

□

Proof of Theorem 6.3.2

First, note that the CDF of the strategy X on $[0, 1]$ is

$$F_X(x) = \int_0^x p_\delta(y)\,\mathrm{d}y = -\frac{1-\delta}{c}\log\left[1 + \left(e^{-\frac{c}{1-\delta}} - 1\right)x\right],$$

which is strictly increasing (and hence one-to-one) on $[0, 1]$, with $F_X(0) = 0$ and $F_X(1) = 1$. The corresponding inverse CDF is

$$F^{-1}_X(y) = \frac{1 - e^{-\frac{cy}{1-\delta}}}{1 - e^{-\frac{c}{1-\delta}}},$$

for $y \in [0, 1]$. This is strictly increasing in y, so F^{-1}_X is one-to-one, and any adversary decision $s \in [0, 1]$ corresponds to some $y \in [0, 1]$ such that $s = F^{-1}_X(y)$.

Now suppose the adversary's decision is $s = F_X^{-1}(y)$ for $y \leq 1-\delta$. Then by Lemma 6.3.1, the the δ-CR of the algorithm's cost in this case is

$$
\begin{aligned}
&\frac{\mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X>s} + (X+1)\cdot \mathbb{1}_{X\leq s}]}{s} \\
&\quad = \frac{1}{F_X^{-1}(y)}\frac{1}{1-\delta}\left[(1-\delta-y)F_X^{-1}(y) + \int_0^y 1 + F_X^{-1}(t)\,\mathrm{d}t\right] \\
&\quad = \frac{1}{1-\delta}\left[(1-\delta-y) + \frac{1-e^{-\frac{c}{1-\delta}}}{1-e^{-\frac{cy}{1-\delta}}}\left(y + \int_0^y \frac{1-e^{-\frac{ct}{1-\delta}}}{1-e^{-\frac{c}{1-\delta}}}\,\mathrm{d}t\right)\right] \\
&\quad = \frac{1}{1-\delta}\left[(1-\delta-y) + \frac{1-e^{-\frac{c}{1-\delta}}}{1-e^{-\frac{cy}{1-\delta}}}\left(y + \frac{1-cy-e^{-\frac{cy}{1-\delta}}(1-\delta)-\delta}{c(e^{-\frac{c}{1-\delta}}-1)}\right)\right] \\
&\quad = \frac{1}{1-\delta}\left[(1-\delta-y) + \frac{2-e^{-\frac{c}{1-\delta}}}{1-e^{-\frac{cy}{1-\delta}}}y - \frac{1-\delta}{c}\right] \\
&\quad = 1 + \frac{y}{1-\delta}\left(\frac{2-e^{-\frac{c}{1-\delta}}}{1-e^{-\frac{cy}{1-\delta}}} - 1\right) - \frac{1}{c} \qquad (6.20) \\
&\quad \leq \frac{2-e^{-\frac{c}{1-\delta}}}{1-e^{-c}} - \frac{1}{c} \qquad (6.21)
\end{aligned}
$$

where the final inequality (6.21) follows from the straightforward observation that (6.20) is increasing in y, so is maximized in this case at $y = 1-\delta$ (recall we have assumed $y \leq 1-\delta$).

Now, consider the alternative case that $y > 1-\delta$. By Lemma 6.3.1, the the δ-CR of the algorithm's cost in this case is

$$
\begin{aligned}
\frac{\mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X>s} + (X+1)\cdot \mathbb{1}_{X\leq s}]}{s} &= \frac{1}{F_X^{-1}(y)}\frac{1}{1-\delta}\int_{y-(1-\delta)}^y 1 + F_X^{-1}(t)\,\mathrm{d}t \\
&= \frac{1-e^{-\frac{c}{1-\delta}}}{1-e^{-\frac{cy}{1-\delta}}}\left(1 + \frac{e^{-\frac{cy}{1-\delta}}(1-e^c)+c}{c(1-e^{-\frac{c}{1-\delta}})}\right) \\
&= \frac{2-e^{-\frac{c}{1-\delta}}}{1-e^{-\frac{cy}{1-\delta}}} + \frac{e^{-\frac{cy}{1-\delta}}(1-e^c)}{c(1-e^{-\frac{cy}{1-\delta}})} \qquad (6.22) \\
&\leq \frac{2-e^{-\frac{c}{1-\delta}}}{1-e^{-\frac{c}{1-\delta}}} + \frac{e^{-\frac{c}{1-\delta}}(1-e^c)}{c(1-e^{-\frac{c}{1-\delta}})} \qquad (6.23)
\end{aligned}
$$

where the final inequality (6.23) follows from the fact that (6.22) is increasing in y, and thus is maximized for $y = 1$ (recall that $y \in (1-\delta, 1]$ in this case). To see that this is the case, observe that

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}y}\left(\frac{2-e^{-\frac{c}{1-\delta}}}{1-e^{-\frac{cy}{1-\delta}}}+\frac{e^{-\frac{cy}{1-\delta}}(1-e^{c})}{c(1-e^{-\frac{cy}{1-\delta}})}\right) &= -\frac{e^{\frac{c(y-1)}{1-\delta}}(-c+e^{\frac{c}{1-\delta}}(1+2c-e^{c}))}{(1-e^{\frac{cy}{1-\delta}})^{2}(1-\delta)} \\
&= \frac{e^{\frac{c(y-1)}{1-\delta}}c}{(1-e^{\frac{cy}{1-\delta}})^{2}(1-\delta)} \qquad (6.24) \\
&> 0 \quad \text{for all } y \in (1-\delta, 1],
\end{aligned}$$

where (6.24) follows by the assumption in the theorem statement that $c = -\frac{1+2W_{-1}(-1/2\sqrt{e})}{2}$, since if we substitute this definition of c into $1+2c-e^{c}$, we obtain

$$\begin{aligned}
1+2c-e^{c} &= -2W_{-1}\left(-1/2\sqrt{e}\right) - e^{-\frac{1+2W_{-1}(-1/2\sqrt{e})}{2}} \\
&= -e^{-\frac{1}{2}}\left(2\sqrt{e}\cdot W_{-1}\left(-1/2\sqrt{e}\right) + e^{-W_{-1}(-1/2\sqrt{e})}\right) \\
&= 0, \qquad (6.25)
\end{aligned}$$

since the Lambert W function is defined to satisfy $W_k(z)\cdot e^{W_k(z)} = z$.

Combining the two cases (6.21) and (6.23), we have that the δ-CR of the algorithm that buys on a random day with density p_δ is

$$\alpha_{\delta}^{p_\delta} = \max\left\{\frac{2-e^{-\frac{c}{1-\delta}}}{1-e^{-c}} - \frac{1}{c}, \frac{2-e^{-\frac{c}{1-\delta}}}{1-e^{-\frac{c}{1-\delta}}} + \frac{e^{-\frac{c}{1-\delta}}(1-e^{c})}{c(1-e^{-\frac{c}{1-\delta}})}\right\}. \qquad (6.26)$$

We will now show that for our chosen constant c, the latter entry in the maximum is larger for all $\delta \in [0,1)$. Define a function f as the difference of (6.23) and (6.21):

$$f(\delta;c) = \frac{2-e^{-\frac{c}{1-\delta}}}{1-e^{-\frac{c}{1-\delta}}} + \frac{e^{-\frac{c}{1-\delta}}(1-e^{c})}{c(1-e^{-\frac{c}{1-\delta}})} - \left(\frac{2-e^{-\frac{c}{1-\delta}}}{1-e^{-c}} - \frac{1}{c}\right).$$

Our goal is to show that $f(\delta;c) \geq 0$ for all $\delta \in [0,1)$. First, observe that $f(0) = 0$. Moreover, since $\lim_{\delta\uparrow 1} e^{-\frac{c}{1-\delta}} = 0$, we have

$$\begin{aligned}
\lim_{\delta\uparrow 1} f(\delta;c) &= 2 - \frac{2}{1-e^{-c}} + \frac{1}{c} \\
&= -\frac{2e^{-c}}{1-e^{-c}} + \frac{1}{c} \\
&= -\frac{2}{e^{c}-1} + \frac{1}{c} \\
&= 2\left(\frac{1}{1-e^{c}} + \frac{1}{2c}\right) \\
&= 0,
\end{aligned}$$

where the final equality follows from rearranging the equality $1+2c-e^c=0$ shown in (6.25), which follows from our choice of c. Thus the function $f(\delta;c)$ is zero at the endpoints of the interval $[0,1)$. Since f' is continuously differentiable, if we can show that $f'(0;c)>0$ and that $f'(\delta;c)=0$ exactly once on the interval $[0,1)$, these together will imply the desired property that $f(\delta;c)\geq 0$ for all $\delta\in[0,1)$.[4]

Computing the derivative of f at $\delta=0$, we find

$$
\begin{aligned}
f'(\delta;c)\Big|_{\delta=0} &= \frac{e^{-\frac{c}{1-\delta}}\left((1-e^c)^2 - c\left(-1+2e^c-2e^{-\frac{c\delta}{1-\delta}}+e^{-\frac{c(1+\delta)}{1-\delta}}\right)\right)}{(e^c-1)(1-e^{-\frac{c}{1-\delta}})^2(1-\delta)^2}\Bigg|_{\delta=0} \qquad (6.27)\\
&= \frac{c+e^c(e^c-1-2c)}{(e^c-1)^2}\\
&= \frac{c}{(e^c-1)^2} \qquad (6.28)\\
&> 0,
\end{aligned}
$$

where (6.28) follows from (6.25). Moreover, inspecting the form of (6.27), it is clear that $e^{-\frac{c}{1-\delta}}$ and the denominator $(e^c-1)(1-e^{-\frac{c}{1-\delta}})^2(1-\delta)^2$ are both strictly positive (recall, in particular, that $c>1$). As such, the sign of $f'(\delta;c)$ is exactly the sign of $(1-e^c)^2-c\left(-1+2e^c-2e^{-\frac{c\delta}{1-\delta}}+e^{-\frac{c(1+\delta)}{1-\delta}}\right)$, so to determine the zeros of $f'(\delta;c)$, we may instead determine the zeros of the expression

$$
(1-e^c)^2-c\left(-1+2e^c-2e^{-\frac{c\delta}{1-\delta}}+e^{-\frac{c(1+\delta)}{1-\delta}}\right).
$$

To this end, we compute another derivative:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\delta}\left[(1-e^c)^2-c\left(-1+2e^c-2e^{-\frac{c\delta}{1-\delta}}+e^{-\frac{c(1+\delta)}{1-\delta}}\right)\right] &= \frac{2c^2e^{-\frac{c\delta}{1-\delta}}(e^{-\frac{c}{1-\delta}}-1)}{(1-\delta)^2}\\
&< 0
\end{aligned}
$$

for all $\delta\in[0,1)$, since $e^{-\frac{c}{1-\delta}}<1$. As

$$
\begin{aligned}
(1-e^c)^2-c\left(-1+2e^c-2e^{-\frac{c\delta}{1-\delta}}+e^{-\frac{c(1+\delta)}{1-\delta}}\right)\Bigg|_{\delta=0} &= 4c^2-c(-3+2e^c+e^{-c})\\
&= 4c^2-c(-1+4c+e^{-c})\\
&= -\frac{(1+2\cdot W_{-1}\left({}^{-1}/_{2\sqrt{e}}\right))^2}{4\cdot W_{-1}\left({}^{-1}/_{2\sqrt{e}}\right)}\\
&\approx 0.899 > 0
\end{aligned}
$$

[4] To see that this is the case, suppose instead that $f(\delta';c)<0$ for some $\delta'\in(0,1)$, and note that strict positivity of the initial derivative $f'(0;c)>0$, continuity of f', and the limit $\lim_{\delta\uparrow 1} f(\delta;c)=0$ imply that $f'(\delta;c)$ must be zero at least twice on the interval, contradicting the supposition that $f'(\delta;c)=0$ exactly once.

and

$$\lim_{\delta \uparrow 1} (1-e^c)^2 - c\left(-1 + 2e^c - 2e^{-\frac{c\delta}{1-\delta}} + e^{-\frac{c(1+\delta)}{1-\delta}}\right) = c - 2ce^c + (e^c-1)^2$$
$$= \frac{1}{2} + W_{-1}\left(-1/2\sqrt{e}\right)$$
$$\approx -1.256 < 0,$$

it follows that $(1-e^c)^2 - c\left(-1 + 2e^c - 2e^{-\frac{c\delta}{1-\delta}} + e^{-\frac{c(1+\delta)}{1-\delta}}\right)$, and thus $f'(\delta; c)$, has exactly one zero on $[0, 1)$. As argued previously, this implies that $f(\delta; c) \geq 0$ for all $\delta \in [0, 1)$, and hence the second entry on the right-hand side of (6.26) is always larger:

$$\alpha_\delta^{p_\delta} = \frac{2 - e^{-\frac{c}{1-\delta}}}{1 - e^{-\frac{c}{1-\delta}}} + \frac{e^{-\frac{c}{1-\delta}}(1-e^c)}{c(1-e^{-\frac{c}{1-\delta}})}.$$

Simplifying this formula via (6.25), we have

$$\alpha_\delta^{p_\delta} = 2 - \frac{1}{e^{\frac{c}{1-\delta}} - 1},$$

from which it is readily observed that $\alpha_\delta^{p_\delta} < 2$ for all $\delta \in [0, 1)$; moreover, $\lim_{\delta \uparrow 1} 2 - \frac{1}{e^{\frac{c}{1-\delta}}-1} = 2$, so the above expression for $\alpha_\delta^{p_\delta}$ is valid, and indeed optimal, in the case of $\delta = 1$ (in this case, we interpret the algorithm as placing full probability mass on purchasing at time 1). On the other hand, $\alpha_0^{p_0} = 2 - \frac{1}{e^c - 1} \approx 1.60 > \alpha_0^* \approx 1.58$, so this algorithm is not optimal for all δ, though it provides a very close approximation of the optimal competitive ratio in the case of $\delta = 0$. □

Proof of Theorem 6.3.3

It is known that the optimal ski-rental algorithm μ^* is *indifferent* to the adversary's decision $s \in (0, 1]$ when $\delta = 0$ (in the expected cost case), i.e., $\alpha_0^{\mu^*}(s) = \frac{e}{e-1}$ for all $s \in (0, 1]$ [231]. A similar tightness property was proved in [253] in the setting of discrete-time ski rental with VaR constraints. In the following, we show that this tightness property also holds for any $\delta \in (0, 1)$ for continuous-time ski rental: if μ_δ^* is optimal for the δ-CR, then $\alpha_\delta^{\mu^*}(s) = \alpha_\delta^{\mu^*}$ for all $s \in (0, 1]$. Following the high-level strategy of [253], we prove this result in two steps: first, we prove that $\alpha_\delta^{\mu^*}(1) = \alpha_\delta^{\mu^*}$. Then, we prove that for any algorithm μ, if $\alpha_\delta^{\mu}(s) < \alpha_\delta^{\mu}$ for some $s \in (0, 1]$, we can construct an algorithm $\hat{\mu}$ with a competitive ratio that is no worse than μ, yet which has $\alpha_\delta^{\hat{\mu}}(1) < \alpha_\delta^{\hat{\mu}}$, thus implying μ is not optimal. We begin with a lemma establishing that any optimal algorithm cannot have a probability mass more than $(1-\delta)(\alpha_\delta^* - 1)$ on any single point.

Lemma 6.B.2. *Let $\delta \in [0,1)$, and let μ^* be an algorithm with optimal δ-*CR *for continuous-time ski rental. Then μ^* cannot assign any point a probability mass greater than $(1-\delta)(\alpha_\delta^* - 1)$.*

Proof. By Lemma 6.2.4 we can assume that μ^* has support in $[0,1]$; now suppose for the sake of contradiction that $\mu^*(x) > (1-\delta)(\alpha_\delta^* - 1)$ for some $x \in [0,1]$. We can easily construct a lower bound on the δ-CR as follows:

$$\begin{aligned}
\alpha_\delta^{\mu^*}(x) &> \frac{(1-\delta)(\alpha_\delta^* - 1)(1+x) + (1-\delta-(1-\delta)(\alpha_\delta^*-1))x}{(1-\delta)x} \\
&= \frac{(1-\delta)(\alpha_\delta^*-1) + (1-\delta)x}{(1-\delta)x} \\
&= \frac{\alpha_\delta^* - 1}{x} + 1 \\
&\geq \alpha_\delta^*
\end{aligned}$$

where the final inequality follows since $\frac{\alpha_\delta^*-1}{x}$ is decreasing in x and $\left.\frac{\alpha_\delta^*-1}{x}+1\right|_{x=1} = \alpha_\delta^*$. Since μ^* was assumed optimal, this strict inequality clearly yields a contradiction.

We conclude by briefly noting that $(1-\delta)(\alpha_\delta^*-1) < 1-\delta$, since by Theorem 6.3.2, $\alpha_\delta^* < 2$ for any $\delta \in [0,1)$. □

Now, we prove that the competitive ratio must be tight when $s = 1$.

Lemma 6.B.3. *Let $\delta \in [0,1)$, and let μ be an algorithm with optimal δ-*CR *for continuous-time ski rental, so $\alpha_\delta^\mu = \alpha_\delta^*$. Then $\alpha_\delta^\mu(1) = \alpha_\delta^\mu$.*

Proof. Suppose otherwise, and let $\alpha_\delta^\mu(1) = \alpha_\delta^\mu - \epsilon$ for some $\epsilon > 0$. Note that by Lemma 6.B.2, μ cannot place a probability mass greater than $(1-\delta)(\alpha_\delta^*-1)$ on any single point, which is strictly less than $(1-\delta)$ by the fact that $\alpha_\delta^* < 2$ for $\delta \in [0,1)$ (Theorem 6.3.2). In particular, this implies that μ does not assign all its probability to the decision $x = 1$, and decreasing the probability mass assigned to a particular decision x and moving it to an earlier decision will strictly decrease the δ-CR at that decision.

Now, consider another algorithm $\hat{\mu}$ with measure defined as:

$$\hat{\mu} = (1-\gamma)\mu + \gamma\delta_1,$$

with $\gamma > 0$ a small constant and where δ_1 is a unit point mass at $x = 1$. For any $s \in (0,1)$, one of the following two cases must hold:

(a) Suppose $\alpha_\delta^\mu(s) = s^{-1} \cdot \mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X>s} + (X+1) \cdot \mathbb{1}_{X\leq s}] = 1$. This means that the worst $(1-\delta)$-sized subpopulation of the loss distribution (i.e., the distribution of the random variable $\mathbb{1}_{X>s} + s^{-1}(X+1) \cdot \mathbb{1}_{X\leq s}$) is contained in the event $X > s$, so it must be that X takes values at most s with probability zero, i.e., $\mu[0,s] = 0$. Likewise, we must have $\hat{\mu}[0,s] = 0$, so $\alpha_\delta^{\hat{\mu}}(s) = 1$ as well.

(b) Alternatively, let $\alpha_\delta^\mu(s) > 1$. This means that $\mu[0,s] = c > 0$. We break into two subcases:

(i) If $c < 1-\delta$, then the worst $(1-\delta)$-sized subpopulation of the loss distribution must yield a loss of 1 with probability $1-\delta-c > 0$; hence we may write

$$\begin{aligned}
\alpha_\delta^\mu(s) &= \mathbb{E}[s^{-1}(X+1) \cdot \mathbb{1}_{X\leq s}] + 1 - \delta - c \\
&= c \cdot \mathbb{E}[s^{-1}(X+1) \cdot \mathbb{1}_{X\leq s} | X \leq S] + 1 - \delta - c.
\end{aligned}$$

Let $B \sim \mathrm{Bernoulli}(\gamma)$ be independent of X, and define $\hat{X}$ as a random variable that is equal to X when $B = 0$ and is 1 otherwise; clearly $\hat{X}$ has distribution $\hat{\mu}$. Then since $\hat{\mu}[0,s] = (1-\gamma)\mu[0,s] = (1-\gamma)c$, we similarly obtain

$$\begin{aligned}
\alpha_\delta^{\hat{\mu}}(s) &= \mathbb{E}[s^{-1}(\hat{X}+1) \cdot \mathbb{1}_{\hat{X}\leq s}] + 1 - \delta - (1-\gamma)c \\
&= \mathbb{E}[s^{-1}(X+1) | \hat{X} \leq s]\mathbb{P}(\hat{X} \leq s) + 1 - \delta - (1-\gamma)c \qquad (6.29) \\
&= \mathbb{E}[s^{-1}(X+1) | X \leq s]\mathbb{P}(\hat{X} \leq s) + 1 - \delta - (1-\gamma)c \qquad (6.30) \\
&= (1-\gamma)c \cdot \mathbb{E}[s^{-1}(X+1) | X \leq s] + 1 - \delta - (1-\gamma)c \\
&< \alpha_\delta^\mu(s), \qquad (6.31)
\end{aligned}$$

where (6.29) holds since, by construction, $\hat{X} \leq s < 1$ implies $\hat{X} = X$; (6.30) follows from the fact that the event $\hat{X} \leq s$ is exactly the joint event $\{X \leq S, B = 0\}$ and X is independent of B; and (6.31) is a consequence of $\gamma > 0$ and $\mathbb{E}[s^{-1}(X+1) | X \leq s] > 1$ for $s < 1$.

(ii) If $c \geq 1-\delta$, then the worst $(1-\delta)$-sized subpopulation of the loss distribution is wholly induced by outcomes of X lying in the interval $[0,s]$; calling ν this subpopulation distribution of X, we have $\alpha_\delta^\mu(s) = \mathbb{E}_{X\sim\nu}[s^{-1}(X+1)]$. This subpopulation shrinks to size $(1-\delta)(1-\gamma)$ in the construction of $\hat{\mu}$, so in order to construct the worst-case $(1-\delta)$-sized loss subpopulation of $\hat{\mu}$, we must augment ν with an additional loss subpopulation (call it $\hat{\nu}$) with size $(1-\delta)\gamma$. It must hold that some nontrivial portion of the losses included in $\hat{\nu}$ are strictly less than $s \cdot \alpha_\delta^\mu(s)$, for if this were not the case, it would imply that μ contains a probability atom of size $\frac{1-\delta}{1-\gamma}$, violating optimality. Thus, we may calculate:

$$\begin{aligned}
\alpha_\delta^{\hat{\mu}}(s) &= (1-\gamma) \cdot \mathbb{E}_{X\sim\nu}[s^{-1}(X+1)] + \gamma \cdot \mathbb{E}_{X\sim\hat{\nu}}[\mathbb{1}_{X>s} + s^{-1}(X+1) \cdot \mathbb{1}_{X\le s}] \\
&< (1-\gamma) \cdot \mathbb{E}_{X\sim\nu}[s^{-1}(X+1)] + \gamma \cdot \alpha_\delta^\mu(s) \qquad (6.32) \\
&= \alpha_\delta^\mu(s),
\end{aligned}$$

where (6.32) follows from a nontrivial portion of the losses in $\hat{\nu}$ being strictly less than $\alpha_\delta^\mu(s)$.

Finally, consider the case of $s = 1$. In this case, the loss is always $X + 1$, since $X \le 1$ without loss of generality (Lemma 6.2.4); since this is strictly increasing in the outcome of X, the worst $(1-\delta)$-sized subpopulation of the loss distribution is exactly the $(1-\delta)$ tail of X, which we call ν. This tail shrinks to size $(1-\delta)(1-\gamma)$ in the construction of $\hat{\mu}$, but an additional probability mass of weight γ is added to the outcome $X = 1$, and as this outcome maximizes the loss, the worst $(1-\delta)$-sized subpopulation of the loss under $\hat{\mu}$ is easily seen to be $(1-\gamma)\nu + \gamma\delta_1$. Thus, if we choose $\gamma = \frac{\epsilon}{2}$,

$$\begin{aligned}
\alpha_\delta^{\hat{\mu}}(1) &= (1-\gamma) \cdot \mathbb{E}_{X\sim\nu}[X+1] + \gamma \cdot \mathbb{E}_{X\sim\delta_1}[X+1] \\
&= (1-\gamma)\alpha_\delta^\mu(1) + 2\gamma \\
&= (1-\gamma)(\alpha_\delta^\mu - \epsilon) + 2\gamma \\
&< (\alpha_\delta^\mu - \epsilon) + 2\gamma \\
&= \alpha_\delta^\mu.
\end{aligned}$$

It follows from the above cases that $\alpha_\delta^{\hat{\mu}} = \sup_{s\in(0,1]} \alpha_\delta^{\hat{\mu}} < \alpha_\delta^\mu$: for $s \in (0,1)$ satisfying case (a), $\alpha_\delta^{\hat{\mu}}(s) = 1 < \alpha_\delta^\mu$, and in case (b) and the case of $s = 1$, we have shown $\alpha_\delta^{\hat{\mu}}(s) < \alpha_\delta^\mu$. However, this implies that $\hat{\mu}$ has strictly better δ-CR than μ, contradicting the optimality of μ. As a result, we must have $\alpha_\delta^\mu(1) = \alpha_\delta^\mu$. □

We will now begin to prove the second structural result: that if $\alpha_\delta^\mu(s) < \alpha_\delta^\mu$ for some $s \in (0,1)$, we can construct an algorithm $\hat{\mu}$ with δ-CR that is no worse than μ and for which $\alpha_\delta^{\hat{\mu}}(1) < \alpha_\delta^{\hat{\mu}}$, which by the previous lemma implies that μ is not optimal. We will first prove a series of technical lemmas that support this proof: the first tells us that, if there is "slack" in the δ-CR for a given adversary decision $x \in (0,1]$, then this implies that there is slack of a comparable magnitude in a small interval $[x, x+\epsilon]$.

Lemma 6.B.4. *Let $\delta \in [0, 1)$, and let μ be an algorithm for continuous-time ski rental with support in $[0, 1]$. Suppose there exists an $x \in (0, 1)$ for which $\alpha_\delta^\mu(x) < \alpha_\delta^\mu$, and define $2\gamma := \alpha_\delta^\mu - \alpha_\delta^\mu(x) > 0$. Then there exists some $\epsilon > 0$ such that for any $y \in [x, x + \epsilon]$, $\alpha_\delta^\mu(y) \leq \alpha_\delta^\mu - \gamma$.*

Proof. When $\delta = 0$, CVaR_δ is exactly the expectation, so we have:

$$\begin{aligned}
\alpha_0^\mu(x) &= \mathbb{E}[\mathbb{1}_{X>x} + x^{-1}(X+1) \cdot \mathbb{1}_{X \leq x}] \\
&= 1 - F_X(x) + x^{-1}\left(F_X(x) + \mathbb{E}[X \cdot \mathbb{1}_{X \leq x}]\right) \\
&= 1 - F_X(x) + x^{-1}\left(F_X(x) + \int_0^x 1 - F_{X \cdot \mathbb{1}_{X \leq x}}(t)\, \mathrm{d}t\right) \\
&= 1 - F_X(x) + x^{-1}\left(F_X(x) + \int_0^x F_X(x) - F_X(t)\, \mathrm{d}t\right),
\end{aligned} \tag{6.33}$$

which is easily seen to be right-continuous at x, by right-continuity of the CDF. Thus there must exist some ϵ ensuring $|\alpha_0^\mu(x) - \alpha_0^\mu(y)| \leq \gamma$ for all $y \in [x, x + \epsilon]$, which implies the desired property.

On the other hand, suppose $\delta \in (0, 1)$. We can choose $\epsilon > 0$ sufficiently small such that $\rho := F_X(x + \epsilon) - F_X(x) = \mathbb{P}(X \in (x, x + \epsilon]) \leq \delta$ (this is always possible due to right-continuity of F_X). Defining the algorithm's cost given an adversary decision s as $C(X, s) = s \cdot \mathbb{1}_{X>s} + (X + 1) \cdot \mathbb{1}_{X \leq s}$, we have

$$\begin{aligned}
\alpha_\delta^\mu(x+\epsilon) &= \frac{1}{(1-\delta)(x+\epsilon)} \int_\delta^1 F_{C(X,x+\epsilon)}^{-1}(t)\, \mathrm{d}t \\
&\leq \frac{1}{(1-\delta)(x+\epsilon)} \left[\int_{\delta-\rho}^{1-\rho} F_{C(X,x+\epsilon)}^{-1}(t)\, \mathrm{d}t + \int_{1-\rho}^1 F_{C(X,x+\epsilon)}^{-1}(t)\, \mathrm{d}t\right].
\end{aligned} \tag{6.34}$$

Now, applying Lemma 6.B.1, we can bound each of the two integrals in (6.34). For the first integral, there are two cases according to the two cases in Lemma 6.B.1: if $\delta - \rho > 1 - F_X(x + \epsilon)$, then $F_{C(X,x+\epsilon)}^{-1}(t) = 1 + F_X^{-1}(t + F_X(x + \epsilon) - 1)$ on the domain of integration, so we have

$$\begin{aligned}
\int_{\delta-\rho}^{1-\rho} F_{C(X,x+\epsilon)}^{-1}(t)\, \mathrm{d}t &= \int_{\delta-\rho}^{1-\rho} 1 + F_X^{-1}(t + F_X(x+\epsilon) - 1)\, \mathrm{d}t \\
&= \int_{F_X(x+\epsilon)-(1-\delta)-\rho}^{F_X(x+\epsilon)-\rho} 1 + F_X^{-1}(t)\, \mathrm{d}t \\
&= \int_{F_X(x)-(1-\delta)}^{F_X(x)} 1 + F_X^{-1}(t)\, \mathrm{d}t \\
&= (1-\delta)\, \mathrm{CVaR}_\delta[C(X, x)]
\end{aligned}$$

where the final equality follows by Lemma 6.3.1. Dividing both sides by $(1-\delta)(x+\epsilon)$, we obtain

$$\frac{1}{(1-\delta)(x+\epsilon)}\int_{\delta-\rho}^{1-\rho} F^{-1}_{C(X,x+\epsilon)}(t)\,\mathrm{d}t = \frac{\mathrm{CVaR}_{\delta}[C(X,x)]}{x+\epsilon} < \alpha^{\mu}_{\delta}(x) \tag{6.35}$$

since $\epsilon > 0$ implies $\frac{x}{x+\epsilon} < 1$. Similarly, if $\delta - \rho \leq 1 - F_X(x+\epsilon)$, the value of $F^{-1}_{C(X,x+\epsilon)}(t)$ depends on which part of the domain of integration contains t:

$$\begin{aligned}
&\int_{\delta-\rho}^{1-\rho} F^{-1}_{C(X,x+\epsilon)}(t)\,\mathrm{d}t \\
&\quad = \int_{\delta-\rho}^{1-F_X(x+\epsilon)} x+\epsilon\,\mathrm{d}t + \int_{1-F_X(x+\epsilon)}^{1-\rho} 1 + F_X^{-1}(t+F_X(x+\epsilon)-1)\,\mathrm{d}t \\
&\quad = (1-\delta-F_X(x))(x+\epsilon) + \int_0^{F_X(x)} 1 + F_X^{-1}(t)\,\mathrm{d}t.
\end{aligned}$$

Dividing both sides by $(1-\delta)(x+\epsilon)$, we obtain

$$\begin{aligned}
&\frac{1}{(1-\delta)(x+\epsilon)}\int_{\delta-\rho}^{1-\rho} F^{-1}_{C(X,x+\epsilon)}(t)\,\mathrm{d}t \\
&\quad = \frac{1}{1-\delta}\left[(1-\delta-F_X(x)) + \frac{1}{x+\epsilon}\int_0^{F_X(x)} 1 + F_X^{-1}(t)\,\mathrm{d}t\right] \\
&\quad < \frac{1}{(1-\delta)x}\left[(1-\delta-F_X(x))x + \int_0^{F_X(x)} 1 + F_X^{-1}(t)\,\mathrm{d}t\right] \\
&\quad = \alpha^{\mu}_{\delta}(x),
\end{aligned} \tag{6.36}$$

where the final step is a consequence of Lemma 6.3.1; note this exactly matches the bound (6.35) in the first case. For the second integral in (6.34), since $\rho = F_X(x+\epsilon) - F_X(x)$, we have $1-\rho = 1 - F_X(x+\epsilon) + F_X(x) \geq 1 - F_X(x+\epsilon)$, so by Lemma 6.B.1 we may calculate

$$\begin{aligned}
\int_{1-\rho}^{1} F^{-1}_{C(X,x+\epsilon)}(t)\,\mathrm{d}t &= \int_{1-\rho}^{1} 1 + F_X^{-1}(t+F_X(x+\epsilon)-1)\,\mathrm{d}t \\
&= \int_{F_X(x)}^{F_X(x+\epsilon)} 1 + F_X^{-1}(t)\,\mathrm{d}t \\
&\leq (F_X(x+\epsilon) - F_X(x))(1 + F_X^{-1}(F_X(x+\epsilon))) \qquad (6.37) \\
&\leq (F_X(x+\epsilon) - F_X(x))(1 + x + \epsilon) \qquad (6.38)
\end{aligned}$$

where the bound (6.37) follows by monotonicity of the inverse CDF, and (6.38) is from the well-known bound $F_X^{-1}(F_X(y)) \leq y$ (e.g., [259, Lemma 1.17f]).

Inserting (6.35), (6.36), and (6.38) into (6.34), we obtain the bound

$$\alpha_\delta^\mu(x+\epsilon) < \alpha_\delta^\mu(x) + \frac{1}{1-\delta}\left(1+\frac{1}{x+\epsilon}\right)(F_X(x+\epsilon)-F_X(x)). \tag{6.39}$$

Right-continuity of the CDF ensures that the right-hand side of (6.39) can be made at most $\alpha_\delta^\mu(x)+\gamma$ by choosing $\epsilon > 0$ sufficiently small, thus establishing the result. □

The second technical lemma we will need to prove the structural result asserts that if X comes from a distribution with a probability atom at $x \in (0,1]$, then there is a non-degenerate interval of slack in the δ-CR. The assumed bound on the size of the probability atom is made without loss of generality due to Lemma 6.B.2, since we solely focus on the optimal algorithm.

Lemma 6.B.5. *Fix $\delta \in [0,1)$, and let $X \sim \mu$ be a random variable supported in $[0,1]$ with a probability atom of mass $\eta \le (\alpha_\delta^*-1)(1-\delta)$ at some $x \in (0,1]$. Then there is some $\epsilon > 0$ and $\gamma > 0$ such that $\alpha_\delta^\mu(y) \le \alpha_\delta^\mu(x) - \gamma$ for all $y \in [x-\epsilon, x)$.*

Proof. Let us assume that there exists an $\epsilon > 0$ such that $F_X(x-\epsilon) > 1-\delta$; the alternative case follows from an essentially identical argument. Then by Lemma 6.3.1,

$$\begin{aligned}
&\alpha_\delta^\mu(x) - \alpha_\delta^\mu(x-\epsilon)\\
&\quad= \frac{1}{1-\delta}\left[\frac{1}{x}\int_{F_X(x)-(1-\delta)}^{F_X(x)} 1+F_X^{-1}(t)\,\mathrm{d}t - \frac{1}{x-\epsilon}\int_{F_X(x-\epsilon)-(1-\delta)}^{F_X(x-\epsilon)} 1+F_X^{-1}(t)\,\mathrm{d}t\right]\\
&\quad\overset{\text{as }\epsilon\downarrow 0}{=} \frac{1}{(1-\delta)x}\left[\int_{F_X(x)-\eta}^{F_X(x)} 1+F_X^{-1}(t)\,\mathrm{d}t - \int_{F_X(x)-\eta-(1-\delta)}^{F_X(x)-(1-\delta)} 1+F_X^{-1}(t)\,\mathrm{d}t\right] \qquad (6.40)\\
&\quad> 0,
\end{aligned}$$

where (6.40) follows by taking the limit $\epsilon \downarrow 0$ and by the assumption $\eta \le (\alpha_\delta^*-1)(1-\delta) < 1-\delta$. The final inequality follows from $F_X^{-1}(t) = x$ on $t \in (F_X(x)-\eta, F_X(x)]$, along with the fact that $F_X^{-1}(t) < x$ on $t \le F_X(x)-\eta$, which implies that the integrand of the second integral in (6.40) is strictly less than the first integrand (which is $1+x$), as $\eta < 1-\delta$. This strict inequality holds in the limit, so there exists some $\gamma > 0$ for which $\lim_{\epsilon\downarrow 0} \alpha_\delta^\mu(x) - \alpha_\delta^\mu(x-\epsilon) = 2\gamma$; that this holds in the limit implies that there is some $\epsilon > 0$ such that $\alpha_\delta^\mu(y) \le \alpha_\delta^\mu(x) - \gamma$ for all $y \in [x-\epsilon, x)$, as desired. □

We are now prepared to prove the structural result establishing that the optimal algorithm has δ-CR that is independent of the adversary's choice of season duration $x \in (0,1)$.

Lemma 6.B.6. *Let $\delta \in [0,1)$, and let μ be an algorithm with optimal δ-*CR *for continuous-time ski rental, so $\alpha_\delta^\mu = \alpha_\delta^*$. Then $\alpha_\delta^\mu(x) = \alpha_\delta^\mu$ for all $x \in (0,1)$.*

Proof. Suppose for the sake of contradiction that $\alpha_\delta^\mu(x) < \alpha_\delta^\mu$ for some $x \in (0,1)$. We will construct another algorithm $\hat{\mu}$ with $\alpha_\delta^{\hat{\mu}} \le \alpha_\delta^\mu$ and $\alpha_\delta^{\hat{\mu}}(1) < \alpha_\delta^{\hat{\mu}}$, which by Lemma 6.B.3 immediately implies that $\hat{\mu}$, and therefore μ, is not optimal. In the following proof, we will say that μ has "slack" at x when $\alpha_\delta^\mu(x) < \alpha_\delta^\mu$.

Define $x = \sup\{y \in [0,1] : \alpha_\delta^\mu(y) < \alpha_\delta^\mu\}$. If $x = 1$ and $\alpha_\delta^\mu(x) < \alpha_\delta^\mu$, clearly by Lemma 6.B.3 we're done. On the other hand, if $x < 1$, we must have $\alpha_\delta^\mu(x) = \alpha_\delta^\mu$ by the supremum definition of x and Lemma 6.B.4. Thus, we may proceed assuming that $x \in (0,1]$ and $\alpha_\delta^\mu(x) = \alpha_\delta^\mu$. Note that the interval $(x,1]$, if it is nonempty, cannot contain any probability atoms, by Lemma 6.B.5 and the definition of x.

We will now proceed to prove the result in two parts. First, we will show that we can construct a distribution $\tilde{\mu}$ with no worse δ-CR than μ that has slack at x, i.e., $\alpha_\delta^{\tilde{\mu}}(x) < \alpha_\delta^{\tilde{\mu}}$. Then, we will show that we can construct new distributions iteratively propagating this slack toward 1, eventually culminating with the desired $\hat{\mu}$ with $\alpha_\delta^{\hat{\mu}} \le \alpha_\delta^\mu$ and $\alpha_\delta^{\hat{\mu}}(1) < \alpha_\delta^{\hat{\mu}}$.

Part (1): Obtaining slack at x We break into two cases depending on whether μ has a probability atom at x.

(a) Suppose μ has a probability atom at x of size ζ; by Lemma 6.B.2, ζ must be bounded as

$$\zeta \le (\alpha_\delta^* - 1)(1-\delta) < 1 - \delta.$$

Then by Lemma 6.B.5, there is an $\epsilon > 0$ and $\gamma > 0$ such that $\alpha_\delta^\mu(y) \le \alpha_\delta^\mu - \gamma$ for all $y \in [x-\epsilon, x)$. Define a measure $\tilde{\mu}$ from μ by moving a small amount of mass $\eta > 0$ from x to $x - \epsilon$; it can easily be seen that this will not change $\alpha_\delta^\mu(y)$ for $y < x - \epsilon$, it will strictly decrease $\alpha_\delta^\mu(x)$ due to shifting of its mass to an action with a smaller cost, and similarly it will either decrease or not affect $\alpha_\delta^\mu(y)$ for $y > x$. Note that Lemma 6.B.2's bound on ζ's size $\zeta \le (\alpha_\delta^* - 1)(1-\delta) < 1-\delta$ is crucial to obtain that $\alpha_\delta^\mu(x)$ strictly decreases, since if ζ were larger than $1-\delta$, decreasing its mass by a small amount might not change the δ-CR at x.

On the interval $[x-\epsilon, x)$, the δ-CR will increase, but at most by $\frac{\eta}{1-\delta}\left(1 + \frac{1}{x-\epsilon}\right)$, which by choosing η small can be kept sufficiently small that $\alpha_\delta^{\tilde{\mu}}(y) < \alpha_\delta^\mu$

remains true for all $y \in [x-\epsilon, x)$. To see this, note that this movement of mass increases F_X by η on the interval $[x-\epsilon, x)$; as a result, on the interval $(F_X(x-\epsilon), F_X(x-\epsilon)+\eta]$, F_X^{-1} will decrease to $x-\epsilon$. That is, $\tilde{X} \sim \tilde{\mu}$ will have inverse CDF of the form

$$F_{\tilde{X}}^{-1}(p) = \begin{cases} F_X^{-1}(p) & \text{if } p \le F_X(x-\epsilon) \\ x-\epsilon & \text{if } p \in (F_X(x-\epsilon), F_X(x-\epsilon)+\eta] \\ F_X^{-1}(p-\eta) & \text{if } p \in (F_X(x-\epsilon)+\eta, F_X(x)] \\ F_X^{-1}(p) & \text{otherwise.} \end{cases} \tag{6.41}$$

Assuming that $F_X(x) > 1-\delta$ (the alternative case proceeds similarly), we may compute, using Lemma 6.3.1:

$$\begin{aligned}(1-\delta)(x-\epsilon)\cdot\alpha_\delta^{\tilde{\mu}}(x-\epsilon) &= \int_{F_{\tilde{X}}(x-\epsilon)-(1-\delta)}^{F_{\tilde{X}}(x-\epsilon)} 1 + F_{\tilde{X}}^{-1}(t)\,\mathrm{d}t \\ &= \int_{F_X(x-\epsilon)+\eta-(1-\delta)}^{F_X(x-\epsilon)+\eta} 1 + F_{\tilde{X}}^{-1}(t)\,\mathrm{d}t \\ &= \eta(1+x-\epsilon) + \int_{F_X(x-\epsilon)+\eta-(1-\delta)}^{F_X(x-\epsilon)} 1 + F_X^{-1}(t)\,\mathrm{d}t \\ &\le \eta(1+x-\epsilon) + (1-\delta)(x-\epsilon)\cdot\alpha_\delta^{\mu}(x-\epsilon),\end{aligned}$$

implying that $\alpha_\delta^{\tilde{\mu}}(x-\epsilon) \le \alpha_\delta^{\mu}(x-\epsilon) + \frac{\eta}{1-\delta}\left(1+\frac{1}{x-\epsilon}\right)$. Similarly, for any $y \in (x-\epsilon, x)$ with $F_X(y) - F_X(x) < 1-\delta$ (any others are not impacted by this change), we can obtain an analogous bound:

$$(1-\delta)y\cdot\alpha_\delta^{\tilde{\mu}}(y) \le \eta(1+x-\epsilon) + (1-\delta)y\cdot\alpha_\delta^{\mu}(y)$$

implying that $\alpha_\delta^{\tilde{\mu}}(y) \le \alpha_\delta^{\mu}(y) + \frac{\eta}{1-\delta}\frac{1+x-\epsilon}{y} \le \alpha_\delta^{\mu}(y) + \frac{\eta}{1-\delta}\left(1+\frac{1}{x-\epsilon}\right)$. Note that these bounds are coarse, and intuitively capture the idea that if we introduce a new point mass of size η within the support of the worst-case loss subpopulation distribution realizing the CVaR_δ, the worst that this added loss can do is increase the CVaR_δ in proportion to the loss value that it adds, weighted by its probability normalized by $1-\delta$. Thus, it is clear that by selecting η sufficiently small, we can guarantee that $\alpha_\delta^{\tilde{\mu}}(y) < \alpha_\delta^{\mu}(y)$ for all $y \in [x-\epsilon, x)$, while we have still strictly decreased $\alpha_\delta^{\mu}(x)$ by moving some of its mass to an earlier decision, thus introducing slack at x.

(b) Suppose that μ does not have a point mass at x; thus, F_X is continuous at x. We break into two further cases:

(i) Suppose that there is some $y < x$ such that μ has a point mass at y and

$$\lim_{h \downarrow 0} F_X(x) - F_X(y-h) < 1-\delta.$$

By the argument in part (a), we can construct a measure $\tilde{\mu}$ by moving some small amount $\eta > 0$ of mass from y to $y-\epsilon$ for some $\epsilon > 0$, which will not impact $\alpha_\delta^\mu(z)$ for $z < y-\epsilon$, will increase it in a controlled manner (such that we can maintain slack by choosing η sufficiently small) for $z \in [y-\epsilon, y)$, and will strictly decrease it for $z = y$. We can choose ϵ sufficiently small that $F_X(x) - F_X(y-\epsilon) \leq 1-\delta$, by the strict inequality in the limit assumed above. Moreover, this modification will also strictly decrease $\alpha_\delta^\mu(z)$ for $z \in [y, x]$. In particular, for x we have, using the inverse CDF expression (6.41) and Lemma 6.3.1,

$$\begin{aligned}
(1-\delta)x \cdot \alpha_\delta^{\tilde{\mu}}(x) &= \int_{F_{\tilde{X}}(x)-(1-\delta)}^{F_{\tilde{X}}(x)} 1 + F_{\tilde{X}}^{-1}(t)\,\mathrm{d}t \\
&= \int_{F_X(x)-(1-\delta)}^{F_X(x)} 1 + F_{\tilde{X}}^{-1}(t)\,\mathrm{d}t \\
&= \int_{F_X(x)-(1-\delta)}^{F_X(y)} 1 + F_{\tilde{X}}^{-1}(t)\,\mathrm{d}t + \int_{F_X(y)}^{F_X(x)} 1 + F_{\tilde{X}}^{-1}(t)\,\mathrm{d}t \\
&< \int_{F_X(x)-(1-\delta)}^{F_X(y)} 1 + F_X^{-1}(t)\,\mathrm{d}t + \int_{F_X(y)}^{F_X(x)} 1 + F_X^{-1}(t)\,\mathrm{d}t \\
&= (1-\delta)x \cdot \alpha_\delta^\mu(x),
\end{aligned}$$

where the strict inequality results from $F_X^{-1}(t)$ being strictly decreased on the domain $(F_X(y-\epsilon), F_X(y-\epsilon)+\eta]$ and the choice of ϵ satisfying $F_X(x) - F_X(y-\epsilon) \leq 1-\delta$. In the above derivation we have assumed that $F_X(x) > 1-\delta$; the alternative case proceeds similarly.

Thus, if the worst $(1-\delta)$ fraction of loss outcomes when the adversary chooses x contains decisions (of positive probability) from a non-degenerate set $[y-\epsilon, y]$, and y has a probability atom, we have that introducing slack at y in turn introduces slack at x.

(ii) Suppose that μ has no point mass at any y satisfying the property that $y < x$ and $\lim_{h \downarrow 0} F_X(x) - F_X(y-h) < 1-\delta$. This implies that there must be a small interval to the left of x on which F_X is continuous, since F_X must increase from $F_X(x) - (1-\delta)$ to $F_X(x)$ without any discontinuities. There thus must exist, by the supremum definition of x, some $y < x$ at which F_X is continuous, $\alpha_\delta^\mu(y) < \alpha_\delta^\mu$, and $F_X(x) - F_X(y) < 1 - \delta$. Continuity of the δ-CR on this subinterval, and the bound (6.39) in particular, imply that there is some non-degenerate interval $[y, y+\epsilon]$ and $\gamma > 0$ such that $\alpha_\delta^\mu(z) \le \alpha_\delta^\mu - \gamma$ for all $z \in [y, y+\epsilon]$. By the assumption that $\alpha_\delta^\mu(x) = \alpha_\delta^\mu$ while there are no point masses on the interval $[y, x]$, the bound (6.39) in particular tells us that we can choose ϵ such that the half-open interval $(y, y+\epsilon]$ has strictly positive measure. Then suppose we move a small fraction η of the probability mass on $(y, y+\epsilon]$ to y. By the same basic argument as employed previously, this will not affect the δ-CR $\alpha_\delta^\mu(z)$ for $z < y$, and if we choose η small enough, it will increase $\alpha_\delta^\mu(z)$ for $z \in [y, y+\epsilon)$ in a controlled fashion so that we can keep $\alpha_\delta^{\tilde{\mu}}(z) < \alpha_\delta^\mu$. Moreover, by the assumption $F_X(x) - F_X(y) < 1 - \delta$, $\alpha_\delta^\mu(x)$ will strictly decrease, just as it did in the previous subcase. Thus, we can introduce slack at x while not increasing the δ-CR.

Having obtained a measure $\tilde{\mu}$ with a δ-CR not worse than μ and with slack at x, we now proceed to the second part.

Part (2): Obtaining slack at 1 If $x = 1$, we are done; otherwise, recall that by definition $(x, 1]$ cannot have any probability atoms, since this would introduce slack in the interval by Lemma 6.B.5, so $F_{\tilde{X}}$ is continuous on $[x, 1]$ (note that at x itself, it may only be right-continuous). The argument employed to obtain slack at 1 follows an iterated form of the approach in case (b.ii) from part 1: Because $\tilde{\mu}$ has slack at x, the bound (6.39) implies that it has slack in an interval $[x, x+\epsilon]$ with the property that $\tilde{\mu}(x, x+\epsilon] > 0$. Then we may transfer a fraction η of the probability mass on $(x, x+\epsilon]$ to x, and as long as η is chosen sufficiently small, this will increase $\alpha_\delta^{\tilde{\mu}}(y)$ for $y \in [x, x+\epsilon)$ while maintaining slack, and it will strictly decrease $\alpha_\delta^{\tilde{\mu}}(z)$ for all $z \ge x+\epsilon$ such that $F_{\tilde{X}}(z) - F_{\tilde{X}}(x) \le 1 - \delta$ (note we can choose ϵ so that this set of z is nonempty). Thus we can "propagate" the slack in the δ-CR to decisions whose CDF value is up to $1 - \delta$ greater than that of the original slack point, x, without increasing the δ-CR of the algorithm. Iteratively applying this construction at most $O(\frac{1}{1-\delta})$ times, we eventually obtain an algorithm $\hat{\mu}$ with δ-CR no worse than μ, and with slack at 1: $\alpha_\delta^{\hat{\mu}}(1) < \alpha_\delta^{\hat{\mu}} \le \alpha_\delta^\mu$. □

Using the structural characterization of the optimal algorithm in terms of its δ-CR's indifference to the adversary's choice of ski season duration, we may now prove that the optimal algorithm has a CDF that is both strictly increasing and continuous (i.e., one-to-one) on the interval $[0, 1]$.

Lemma 6.B.7. *Fix $\delta \in [0, 1)$, and let $X \sim \mu$ be a random variable yielding the optimal δ-CR for continuous-time ski rental, i.e., $\alpha_\delta^\mu = \alpha_\delta^*$. Then F_X is strictly increasing on $[0, 1]$.*

Proof. When $\delta \in (0, 1)$, this is an immediate consequence of the strict inequality (6.39) in the proof of Lemma 6.B.4: if F_X were not strictly increasing, there would exist a non-degenerate interval $[a, b] \subseteq [0, 1]$ with $F_X(x) = c$ for all $x \in [a, b]$, which by (6.39) would imply that $\alpha_\delta^\mu(a + \epsilon) < \alpha_\delta^\mu(a)$ for some small $\epsilon > 0$, contradicting (via Lemma 6.B.6) the optimality of μ. Likewise, in the $\delta = 0$ case, F_X not strictly increasing means $F_X(a + \epsilon) = F_X(a)$ for all sufficiently small ϵ, so the expression (6.33) in the proof of Lemma 6.B.4 implies that α_0^μ will strictly decrease for some small interval starting at a, again contradicting the optimality of μ by Lemma 6.B.6. □

Lemma 6.B.8. *Fix $\delta \in [0, 1)$, and let $X \sim \mu$ be a random variable yielding the optimal δ-CR for continuous-time ski rental, i.e., $\alpha_\delta^\mu = \alpha_\delta^*$. Then F_X is continuous on $[0, 1]$, $F_X(0) = 0$, and $F_X(1) = 1$.*

Proof. The first two properties amount to proving that the optimal algorithm contains no probability atoms; the third is always satisfied, since we can assume without loss of generality that X is supported on $[0, 1]$ (Lemma 6.2.4). Suppose for the sake of contradiction that μ has an atom at $x \in [0, 1]$. If $x = 0$, μ cannot be competitive, let alone optimal, yielding a contradiction and establishing $F_X(0) = 0$. Otherwise, if $x \in (0, 1]$, Lemma 6.B.5 implies that there is slack in the δ-CR—i.e., there is some $\epsilon > 0$ such that $\alpha_\delta^\mu(x - \epsilon) < \alpha_\delta^\mu(x)$, which by Lemmas 6.B.3 and 6.B.6 contradicts the optimality of μ. Thus μ has no atoms and F_X is continuous on $[0, 1]$. □

We are now adequately equipped to prove the main result of this section, Theorem 6.3.3.

Proof of Theorem 6.3.3. Let $\phi : [0, 1] \to [0, 1]$ be the inverse CDF of the optimal strategy for continuous-time ski rental with the δ-CR metric. By Lemmas 6.B.7 and 6.B.8, ϕ is strictly increasing and continuous on $[0, 1]$, and hence one-to-one, with $\phi(0) = 0$ and $\phi(1) = 1$. By Lemma 6.3.1, for any fixed adversary decision $s \in (0,1]$, we may express the CVaR_δ of the cost incurred by playing $X \sim \phi$ as

$$\begin{aligned}
&\mathrm{CVaR}_\delta[s \cdot \mathbb{1}_{X>s} + (X+1) \cdot \mathbb{1}_{X \le s}] \\
&\quad = \begin{cases} \frac{1}{1-\delta}\left[(1-\delta-\phi^{-1}(s))s + \int_0^{\phi^{-1}(s)} (1+\phi(p))\,\mathrm{d}p\right] & \text{if } \phi^{-1}(s) \le 1-\delta \\ \frac{1}{1-\delta}\int_{\phi^{-1}(s)-(1-\delta)}^{\phi^{-1}(s)} (1+\phi(p))\,\mathrm{d}p & \text{otherwise.} \end{cases}
\end{aligned}$$

By Lemmas 6.B.3 and 6.B.6, the optimal algorithm has competitive ratio independent of the adversary's choice $s \in (0,1]$ of the ski season duration—that is, $\alpha^*_\delta(s) = \alpha^*_\delta$ for all $s \in (0,1]$. As such, ϕ must satisfy the following equations:

$$\begin{aligned}
\frac{1}{1-\delta}\left[(1-\delta-\phi^{-1}(s))s + \int_0^{\phi^{-1}(s)} (1+\phi(p))\,\mathrm{d}p\right] &= \alpha^*_\delta \cdot s \quad \text{if } \phi^{-1}(s) \le 1-\delta \\
\frac{1}{1-\delta}\int_{\phi^{-1}(s)-(1-\delta)}^{\phi^{-1}(s)} (1+\phi(p))\,\mathrm{d}p &= \alpha^*_\delta \cdot s \qquad \text{otherwise}
\end{aligned}$$

for all $s \in (0,1]$. Because ϕ is one-to-one, the above equations holding for all $s \in (0,1]$ is equivalent to their holding for all $t \in (0,1]$ when $s := \phi(t)$ (and hence $t = \phi^{-1}(s)$):

$$\frac{1}{1-\delta}\left[(1-\delta-t)\phi(t) + \int_0^{t} (1+\phi(p))\,\mathrm{d}p\right] = \alpha^*_\delta \cdot \phi(t) \quad \text{if } t \le 1-\delta \tag{6.42}$$

$$\frac{1}{1-\delta}\int_{t-(1-\delta)}^{t} (1+\phi(p))\,\mathrm{d}p = \alpha^*_\delta \cdot \phi(t) \quad \text{otherwise.} \tag{6.43}$$

Differentiating (6.42) with respect to t, we find

$$\begin{aligned}
\phi'(t) + \frac{1}{1-\delta}\left[-\phi(t) - t\phi'(t) + 1 + \phi(t)\right] &= \alpha^*_\delta \cdot \phi'(t) \\
\implies \phi'(t) &= \frac{1}{(\alpha^*_\delta - 1)(1-\delta) + t},
\end{aligned}$$

and integrating this and applying the initial condition $\phi(0) = 0$, we obtain $\phi(t) = \log\left(1 + \frac{t}{(\alpha^*_\delta - 1)(1-\delta)}\right)$ on $t \in [0, 1-\delta]$. Differentiating the second equation (6.43) and rearranging, we obtain

$$\phi'(t) = \frac{1}{\alpha^*_\delta(1-\delta)}\left[\phi(t) - \phi(t-(1-\delta))\right]. \tag{6.44}$$

Thus, as claimed, ϕ is the solution to the delay differential equation (6.44) subject to the initial condition $\phi(t) = \log\left(1 + \frac{t}{(\alpha^*_\delta - 1)(1-\delta)}\right)$ on $t \in [0, 1-\delta]$. Uniqueness of ϕ as the optimal strategy follows from uniqueness results in the theory of delay differential equations, e.g., [265, Theorem 3.1].

□

Optimal solution is strictly decreasing in α

Here, we will prove that the solution $\phi(t)$ to the delay differential equation posed in Theorem 6.3.3 is strictly decreasing in α for $t \in (0, 1]$. Let $\phi_\alpha(t)$ denote the solution at time t for a given choice of α. We will employ a form of induction on the continuum in our argument:

Definition 6.B.9 (Induction on the continuum, [266])**.** *Let $P(t) : \mathbb{R} \to \{0, 1\}$ be a truth-valued function, and let $[a, b]$ be a closed interval. Suppose the following three properties hold:*

(1) $P(a) = 1$;

(2) For any $x \in [a, b)$, $P(t) = 1$ for all $t \in [a, x]$ implies $P(t) = 1$ in some non-degenerate interval $[x, x + \epsilon)$;

(3) For any $x \in (a, b]$, $P(t) = 1$ for all $t \in [a, x)$ implies $P(x) = 1$.

Then $P(t) = 1$ for all $t \in [a, b]$.

Fix $\alpha < \alpha'$; in our setting, $P(t)$ will be the truth function of the strict inequality $\phi_\alpha(t) > \phi_{\alpha'}(t)$, and the interval of interest will be $t \in [1 - \delta, 1]$.

First, note that we clearly have $P(1-\delta) = 1$, since $\phi_\alpha(1-\delta) = \log\left(1 + \frac{1}{\alpha-1}\right)$, which is strictly decreasing in α; in fact, we have that the initial condition $\phi(t) = \log\left(1 + \frac{t}{\alpha-1}\right)$ is strictly decreasing for all $t \in (0, 1 - \delta]$. Thus property (1) is satisfied.

Second, note that the solutions $\phi_\alpha, \phi_{\alpha'}$ are both continuous, as they are the solutions to a delay differential equation. Continuity guarantees that if $\phi_\alpha(t) > \phi_{\alpha'}(t)$, then $\phi_\alpha(x) > \phi_{\alpha'}(x)$ for x in some interval $[t, t + \epsilon]$ with $\epsilon > 0$. Thus property (2) holds.

Finally, suppose $\phi_\alpha(t) > \phi_{\alpha'}(t)$ for t in some interval $[1 - \delta, x)$. By the integral form (6.43) of the delay differential equation, we have

$$\phi_\alpha(x) = \frac{1}{\alpha(1-\delta)} \int_{x-(1-\delta)}^{x} 1 + \phi_\alpha(t)\,\mathrm{d}t > \frac{1}{\alpha'(1-\delta)} \int_{x-(1-\delta)}^{x} 1 + \phi_{\alpha'}(t)\,\mathrm{d}t = \phi_{\alpha'}(x),$$

where the strict inequality follows from $\alpha < \alpha'$ and the assumption that $\phi_\alpha(t) > \phi_{\alpha'}(t)$ for t in $[1 - \delta, x)$ (this inequality also holds for the initial condition on the region $(0, 1 - \delta]$). Thus property (3) holds, and we conclude that $\phi_\alpha(t) > \phi_{\alpha'}(t)$ for all $t \in (0, 1]$.

Analytic solution when $\delta \leq \frac{1}{2}$

When $\delta \leq \frac{1}{2}$, the delay differential equation on the domain $[1-\delta, 1]$ can be written as an ordinary differential equation by substituting the initial condition in for $\phi(t-(1-\delta))$:

$$\phi'(t) = \frac{1}{\alpha(1-\delta)}\left[\phi(t) - \log\left(1 + \frac{t-(1-\delta)}{(\alpha-1)(1-\delta)}\right)\right]$$

with initial value $\phi(1-\delta) = \log\left(1 + \frac{1}{\alpha-1}\right)$. Solving this initial value problem with Mathematica, we obtain that ϕ takes the value

$$\begin{aligned}\phi(t) = e^{-\frac{2(1-\delta)-t}{\alpha(1-\delta)}}&\left[e\cdot\mathrm{Ei}\left(\frac{1}{\alpha}-1\right) - e\cdot\mathrm{Ei}\left(\frac{(2-\alpha)(1-\delta)-t}{\alpha(1-\delta)}\right) + e^{\frac{1}{\alpha}}\log\left(\frac{\alpha}{\alpha-1}\right)\right]\\ &+\log\left(1 + \frac{t-(1-\delta)}{(\alpha-1)(1-\delta)}\right),\end{aligned}$$

on the interval $[1-\delta, 1]$, where $\mathrm{Ei}(x) = -\int_{-x}^{\infty}\frac{e^{-t}}{t}\,\mathrm{d}t$ is the exponential integral.

Proof of Theorem 6.3.4

This proof is essentially a tightening of the phase transition bound in the discrete setting (Theorem 6.4.1(ii); see Section 6.C), enabled by the continuity of the decision space $[0,1]$. The lower bound of $\frac{e}{e-1}$ holds trivially, since increasing δ increases the δ-CR; thus we focus on the second element in the max. Let μ be an arbitrary distribution supported in $[0,1]$.

Let $n = \left\lfloor\frac{1}{1-\delta}\right\rfloor - 1$, and define the indices $i_0 = 0$, $i_k = 1 - \frac{1}{2^k}$ for each $k \in [n]$, and $i_{n+1} = 1$. We partition the unit interval into $n+1$ different sets I_k, with

$$I_k = [i_{k-1}, i_k]$$

for each $k \in [n+1]$. Because $n + 1 = \left\lfloor\frac{1}{1-\delta}\right\rfloor$, by the pigeonhole principle there must be at least one interval I_k with $\mu(I_k) \geq 1-\delta$. Let the adversary's choice of ski season duration be i_k, and suppose $k \leq n$. Because $\mu(I_k) \geq 1-\delta$, there is at least a $(1-\delta)$ fraction of outcomes in which the algorithm's decision x lies in I_k, and thus we can lower bound the δ-CR as

$$\begin{aligned}\alpha_\delta^\mu(i_k) &= \frac{\mathrm{CVaR}_\delta[i_k\cdot\mathbb{1}_{X>i_k} + (X+1)\cdot\mathbb{1}_{X\leq i_k}]}{i_k}\\ &\geq \frac{i_{k-1}+1}{i_k}\\ &= \frac{2-\frac{1}{2^{k-1}}}{1-\frac{1}{2^k}}\\ &= 2.\end{aligned}$$

Alternatively, suppose $k = n + 1$. Again, there is at least a $(1 - \delta)$ fraction of outcomes in which the algorithm's decision x lies in I_{n+1}, so we can lower bound the δ-CR as

$$\begin{aligned}\alpha_\delta^\mu(i_{n+1}) &= \frac{\mathrm{CVaR}_\delta[i_{n+1} \cdot \mathbb{1}_{X > i_{n+1}} + (X+1) \cdot \mathbb{1}_{X \le i_{n+1}}]}{i_{n+1}} \\ &\ge \frac{i_n + 1}{i_{n+1}} \\ &= 2 - \frac{1}{2^n}.\end{aligned}$$

Thus, regardless of which set I_k contains the at least $(1 - \delta)$ fraction of mass, the δ-CR will be at least $2 - \frac{1}{2^n}$. Substituting in the definition of n, we obtain

$$\alpha_\delta^\mu \ge 2 - \frac{1}{2^{\lfloor \frac{1}{1-\delta} \rfloor - 1}}$$

for any algorithm μ. □

6.C Proofs for Section 6.4

Proof of Theorem 6.4.1

We will prove parts (i) and (ii) of the theorem separately.

Proof of Theorem 6.4.1, part (i). Recall that $\alpha_\delta^{\mathrm{DSR}(B),*} \le \alpha_\delta^{\mathrm{CSR},*}$ for all $B \in \mathbb{N}$, since the discrete-time version of the ski rental amounts to restricting the continuous adversary's power. The bound in Theorem 6.3.2 thus implies that $\alpha_\delta^{B,*} \le 2 - \frac{1}{e^{\frac{c}{1-\delta}} - 1}$, so a sufficient condition for $\alpha_\delta^{B,*}$ to strictly improve on the deterministic worst-case competitive ratio of $2 - \frac{1}{B}$ is to have $2 - \frac{1}{e^{\frac{c}{1-\delta}} - 1} < 2 - \frac{1}{B}$. Rearranging this equation to isolate δ yields $\delta < 1 - \frac{c}{\log(B+1)}$, as claimed. □

Proof of Theorem 6.4.1, part (ii). Let $n = \log_2 B$; note that $2^{\lfloor n \rfloor} \le B \le 2^{\lceil n \rceil}$, and hence $1 \le \frac{B}{2^{\lfloor n \rfloor}} < 2$—the second inequality is strict because if it held with equality, this would imply $B = 2^{\lfloor n \rfloor + 1}$, or $\log_2 B = n = \lfloor n \rfloor + 1$. Define the (not necessarily integer) indices $i_k = \frac{(2^k - 1)B}{2^k}$ for every $k \in \{0, \ldots, \lfloor n \rfloor\}$. We construct the following sets: for each $k \in [\lfloor n \rfloor]$, define:

$$\begin{aligned} I_k &= \{\lceil i_{k-1} \rceil + 1, \ldots, \lfloor i_k \rfloor\}, \\ J_k &= \{\lceil i_k \rceil\}, \end{aligned}$$

and define $J_{\lfloor n \rfloor+1} = \{B\}$. It is straightforward to observe that these sets form a cover of the action set $[B]$:

$$\left(\bigcup_{k=1}^{\lfloor n \rfloor} I_k \right) \cup \left(\bigcup_{k=1}^{\lfloor n \rfloor+1} J_k \right) = [B].$$

To see this, simply note that $\lceil i_0 \rceil + 1 = 0 + 1 = 1$ and

$$\lceil i_{\lfloor n \rfloor} \rceil = \left\lceil B - \frac{B}{2^{\lfloor n \rfloor}} \right\rceil = B - 1,$$

since $\frac{B}{2^{\lfloor n \rfloor}} < 2$.

Now suppose that $\frac{1}{1-\delta} \geq 2\lfloor n \rfloor + 1$; then for any strategy $\mathbf{p} \in \Delta_B$, the pigeonhole principle assures us that $\mathbf{p}$ must assign probability at least $(1 - \delta)$ to one of the action sets I_k or J_k in the cover. However, for each of these action sets, there is an adversary decision that forces each action in the set to have competitive ratio at least $2 - \frac{1}{B}$. If the set in question is I_k for some $k \in [\lfloor n \rfloor]$, then if the adversary chooses the true number of skiing days to be $\lfloor i_k \rfloor$, the action in I_k with the smallest competitive ratio is $\lceil i_{k-1} \rceil + 1$, which has competitive ratio lower bounded as

$$\begin{aligned}
\frac{B + (\lceil i_{k-1} \rceil + 1) - 1}{\lfloor i_k \rfloor} &\geq \frac{B + i_{k-1}}{i_k} \\
&= \frac{1 + \frac{2^{k-1}-1}{2^{k-1}}}{\frac{2^k - 1}{2^k}} \\
&= 2.
\end{aligned}$$

On the other hand, if the set in question is one of the singleton sets $J_k = \{x\}$, then if the adversary chooses the true number of skiing days as x, the competitive ratio of this action is lower bounded as

$$\begin{aligned}
\frac{B + x - 1}{x} &= 1 + \frac{B-1}{x} \\
&= 2 - \frac{1}{B} + \left(\frac{B-1}{x} - \frac{B-1}{B} \right) \\
&\geq 2 - \frac{1}{B}
\end{aligned}$$

since $x \in [B]$, and in particular $x \leq B$. Thus, for each set in this cover, there is an adversary decision forcing every action in the set to have competitive ratio at least $2 - \frac{1}{B}$. However, since one of these sets S must have probability at least $1 - \delta$ assigned to it by $\mathbf{p}$, the "bad" adversary decision corresponding to S will yield a competitive ratio of at least $2 - \frac{1}{B}$ with probability at least $1 - \delta$. It immediately follows that the adversary can force a δ-CR of at least $2 - \frac{1}{B}$ in this setting, which implies that the optimal strategy is to buy deterministically at time B, which has a δ-CR of exactly $2 - \frac{1}{B}$.

Rearranging the condition $\frac{1}{1-\delta} \geq 2\lfloor n \rfloor + 1$ to isolate δ, we obtain $\delta \geq 1 - \frac{1}{2\lfloor n \rfloor + 1} = 1 - \frac{1}{2\lfloor \log_2 B \rfloor + 1}$, as claimed. □

Note that the lower bound on δ obtained in the above proof can be improved to $1 - \frac{1}{\log_2 B + 1}$ when B is a power of 2, as in this case all of the sets J_k for $k < n + 1$ are redundant, so eliminating these, the resulting cover has only $n + 1$ sets. This is essentially equivalent to the argument used in the continuous-time lower bound (cf. Section 6.B).

Proof of Theorem 6.4.2

Before proving the theorem, we will first prove an structural lemma analogous to the tightness results Lemmas 6.B.3 and 6.B.6 in the continuous-time setting, which will establish that so long as δ is not too large, the optimal algorithm $\mathbf{p}^{B,\delta,*}$ satisfies the property that $\alpha_\delta^{B,\mathbf{p}^{B,\delta,*}}(i) = \alpha_\delta^{B,*}$ for all $i \in [B]$. In other words, under the algorithm with optimal δ-CR for discrete-time ski rental, the adversary is indifferent to the ski season duration that it chooses. First, we pose an optimization-based formulation of the δ-CR of an arbitrary algorithm $\mathbf{p}$ that will facilitate the analysis.

Lemma 6.C.1. *Let $\delta \in [0, 1)$, and let $\mathbf{p} \in \Delta_B$ be an algorithm for discrete-time ski rental with buying cost B. Then $\alpha_\delta^{B,\mathbf{p}}(i)$, the δ-*CR *when the adversary chooses the true number of skiing days as $i \in [B]$, can be expressed as*

$$\begin{aligned} \alpha_\delta^{B,\mathbf{p}}(i) = \max_{\mathbf{q}\in\mathbb{R}^B} \quad & \frac{1}{1-\delta}\left(\sum_{j=1}^{i} \frac{B+j-1}{i} q_j + \sum_{j=i+1}^{B} q_j\right) \\ \text{s.t.} \quad & \mathbf{0} \leq \mathbf{q} \leq \mathbf{p} \\ & \mathbf{1}^\top \mathbf{q} = 1 - \delta. \end{aligned} \tag{6.45}$$

Proof. This is an immediate consequence of the minimization formulation of CVaR_δ presented in (6.1); in fact, this is the Lagrange dual of that formulation, applied to the definition of δ-CR for continuous-time ski rental. This is also the particular case of the supremum form of CVaR_δ in (6.1) which computes the expected cost on the worst "$(1 - \delta)$-sized subpopulation of the distribution" $\mathbf{p}$ when the loss random variable takes discrete outcomes; the optimal solution $\mathbf{q}$ is obtained by "filling in" $\mathbf{p}$

starting from the indices with highest cost, i.e., starting with i, then $i-1$, through 1, and then starting again with the indices $i+1$ through B, until the probability budget $1-\delta$ has been depleted. This structure of the optimal solution $\mathbf{q}$ can also be obtained from the characterization of CVaR_δ for discrete random variables provided in [238, Proposition 8]. □

We also prove another technical lemma asserting that any if $p_i = 0$ for some index i, this must introduce slack in $\alpha_\delta^{B,\mathbf{p}}(i)$.

Lemma 6.C.2. *Let $\mathbf{p} \in \Delta_B$ be an algorithm for discrete-time ski rental with buying cost B that has δ-CR $\alpha_\delta^{B,\mathbf{p}}$. If $p_i = 0$, then $\alpha_\delta^{B,\mathbf{p}}(i) < \alpha_\delta^{B,\mathbf{p}}$.*

Proof. Suppose $p_1 = 0$; then we may eliminate the variable q_1 and its coefficient in the objective of the maximization form of the δ-CR in (6.45), since q_1 must be zero; then it is clear

$$\begin{aligned}
\alpha_\delta^{B,\mathbf{p}}(1) = \max_{\mathbf{q}\in\mathbb{R}^B} \quad & \frac{1}{1-\delta}\left(\sum_{j=2}^{B} q_j\right) = 1 \\
\text{s.t.} \quad & \mathbf{0} \leq \mathbf{q} \leq \mathbf{p} \\
& \mathbf{1}^\top \mathbf{q} = 1-\delta,
\end{aligned}$$

which is strictly less than $\alpha_\delta^{B,\mathbf{p}}$, since we always have the ordering $\alpha_\delta^{B,\mathbf{p}} \geq \alpha_0^{B,*} = \frac{1}{1-(1-1/B)^B} > 1$.

Alternatively, suppose that $i > 1$ and $p_i = 0$. If it is also the case that $p_j = 0$ for all $j < i$, then the prior argument holds and we once again have $\alpha_\delta^{B,\mathbf{p}}(i) = 1$. Otherwise, at least one $p_j > 0$ for $j < i$. Since $p_i = 0$, we may eliminate the variable q_i and its coefficient in the objective of the maximization form of the δ-CR in (6.45), since q_i

must be zero; thus

$$\begin{aligned}
\alpha_\delta^{B,\mathbf{p}}(i) = \max_{\mathbf{q}\in\mathbb{R}^B} \quad & \frac{1}{1-\delta}\left(\sum_{j=1}^{i-1}\frac{B+j-1}{i}q_j + \sum_{j=i+1}^{B} q_j\right) && (6.46)\\
\text{s.t.} \quad & \mathbf{0} \le \mathbf{q} \le \mathbf{p} \\
& \mathbf{1}^\top \mathbf{q} = 1-\delta \\
< \max_{\mathbf{q}\in\mathbb{R}^B} \quad & \frac{1}{1-\delta}\left(\sum_{j=1}^{i-1}\frac{B+j-1}{i-1}q_j + \sum_{j=i+1}^{B} q_j\right) && (6.47)\\
\text{s.t.} \quad & \mathbf{0} \le \mathbf{q} \le \mathbf{p} \\
& \mathbf{1}^\top \mathbf{q} = 1-\delta \\
= \alpha_\delta^{B,\mathbf{p}}(i-1) &
\end{aligned}$$

where the strict inequality holds due to the fact that, since there is some $p_j > 0$ for $j < i$, the optimal solution of both maximization problems will have some $q_j > 0$, and on this domain the objective of (6.46) is strictly less than that of (6.47). Thus $\alpha_\delta^{B,\mathbf{p}}(i) < \alpha_\delta^{B,\mathbf{p}}(i-1) \le \alpha_\delta^{B,\mathbf{p}}$. □

Using these lemmas, we can now prove the structural result establishing that the optimal algorithm, as long as it has competitive ratio strictly better than $2 - \frac{1}{B}$, has δ-CR independent of the adversary's choice of ski season duration. This is similar in spirit to the "principle of equality" in the expected cost setting (see, e.g., [267]) and our tightness results in continuous time (Lemmas 6.B.3 and 6.B.6).

Lemma 6.C.3. *Let $\delta \in [0,1)$ be such that the optimal δ-CR for ski rental with buying cost B strictly improves on the deterministic optimal, i.e, $\alpha_\delta^{B,*} < 2 - \frac{1}{B}$, and let $\mathbf{p}^{B,\delta,*}$ be an algorithm obtaining this optimal δ-CR. Then*

$$\alpha_\delta^{B,\mathbf{p}^{B,\delta,*}}(i) = \alpha_\delta^{B,*}$$

for all $i \in [B]$.

Proof. We will abbreviate $\mathbf{p}^{B,\delta,*}$ simply as $\mathbf{p}$. By assumption that $\alpha_\delta^{B,*} < 2 - \frac{1}{B}$, it must be the case that $p_i < 1 - \delta$ for all $i \in [B]$.

Suppose for the sake of contradiction that there is some slack in the δ-CR for some adversary decision $i \in [B]$, so that

$$\alpha_\delta^{B,\mathbf{p}}(i) < \alpha_\delta^{B,*}.$$

Similar to the structure of the proof for tightness in the continuous-time setting, we will prove this result in two parts: in part (a), we will show that there exists some other distribution $\hat{\mathbf{p}} \in \Delta_B$ with at least as good δ-CR as $\mathbf{p}$ that has slack at time $i = B$, i.e., $\alpha_\delta^{B,\hat{\mathbf{p}}} \le \alpha_\delta^{B,\mathbf{p}}$ and $\alpha_\delta^{B,\hat{\mathbf{p}}}(B) < \alpha_\delta^{B,*}$. Then, in part (b) we will show that we can redistribute this slack to every other time, i.e., we can construct some other $\tilde{\mathbf{p}} \in \Delta_B$ with $\alpha_\delta^{B,\tilde{\mathbf{p}}} \le \alpha_\delta^{B,\hat{\mathbf{p}}}$ and $\alpha_\delta^{B,\tilde{\mathbf{p}}}(i) < \alpha_\delta^{B,*}$ for all $i \in [B]$, which implies that $\mathbf{p}$ is not optimal.

(a) Let i be the largest element in $[B]$ with the slack property $\alpha_\delta^{B,\mathbf{p}}(i) < \alpha_\delta^{B,*}$; if $i = B$, we may define $\hat{\mathbf{p}} = \mathbf{p}$ and move to part (b). Otherwise, we have $i < B$. Note that since there is no slack for adversary decisions $i+1, \dots, B$, Lemma 6.C.2 implies that $p_j > 0$ for all $j = i+1, \dots, B$. Inspecting the maximization formulation of $\alpha_\delta^{B,\mathbf{p}}(i)$ in (6.45), it is clear that by adding a small constant $\epsilon > 0$ to p_i and subtracting ϵ from p_{i+1}, one can slightly increase $\alpha_\delta^{B,\mathbf{p}}(i)$ while strictly decreasing $\alpha_\delta^{B,\mathbf{p}}(i+1)$, thus introducing slack at $i+1$. This is because increasing p_i (which, recall, must be strictly less than $1-\delta$) by $\epsilon \le 1 - \delta - p_i$ will increase the optimal q_i in the maximization form of $\alpha_\delta^{B,\mathbf{p}}(i)$ by ϵ, since q_i is associated with the largest coefficient in the objective. However, the budget constraint $\mathbf{1}^\top \mathbf{q} = 1 - \delta$ means that some other q_j (or the sum of several q_j) must then decrease by ϵ in the optimal solution, leading to an increase of the optimal value by at most $\epsilon\left(\frac{B+i-1}{i} - 1\right)$. On the other hand, decreasing p_{i+1} by $\epsilon \le p_{i+1}$ will lead to a corresponding decrease by ϵ of the optimal q_{i+1} in the maximization form of $\alpha_\delta^{B,\mathbf{p}}(i+1)$, since q_{i+1} is associated with the largest coefficient in the objective, and the fact $p_{i+1} < 1 - \delta$ means $q_{i+1} = p_{i+1}$ for the optimal $\mathbf{q}$. However, since p_i is increased by ϵ, this decrease in q_{i+1} will be absorbed by q_i (or a combination of multiple q_j, $j \ne i+1$), which is associated with the second largest coefficient in the objective. Altogether, the optimal value of the problem will decrease by at least $\epsilon\left(\frac{B+i}{i+1} - \frac{B+i-1}{i+1}\right)$, meaning that $\alpha_\delta^{B,\mathbf{p}}(i+1)$ has decreased accordingly. Thus if we choose $\epsilon > 0$ satisfying

$$\epsilon \le \min\{1 - \delta - p_i, p_{i+1}\} \qquad \text{and} \qquad \alpha_\delta^{B,\mathbf{p}}(i) + \epsilon\left(\frac{B+i-1}{i} - 1\right) < \alpha_\delta^{B,*},$$

the modified distribution obtained from increasing p_i by ϵ and decreasing p_{i+1} by ϵ will have slack at both i and $i+1$. By similar reasoning, $\alpha_\delta^{B,\mathbf{p}}(j)$ will not be impacted for $j < i$ and will not increase (but might decrease) for $j > i+1$. Repeating this process at $i+1$, $i+2$, and so on, we eventually obtain a distribution $\hat{\mathbf{p}}$ with the properties that $\alpha_\delta^{B,\hat{\mathbf{p}}} \le \alpha_\delta^{B,*}$ and $\alpha_\delta^{B,\hat{\mathbf{p}}}(B) < \alpha_\delta^{B,*}$.

(b) Suppose $\hat{\mathbf{p}}$ is a distribution satisfying the properties $\alpha_\delta^{B,\hat{\mathbf{p}}} \leq \alpha_\delta^{B,*}$ and $\alpha_\delta^{B,\hat{\mathbf{p}}}(B) < \alpha_\delta^{B,*}$, and define a new distribution $\tilde{\mathbf{p}}$ by shifting a small ϵ fraction of the mass on all actions $i < B$ to B:

$$\tilde{p}_i = \begin{cases} (1-\epsilon)\hat{p}_i & \text{for } i < B \\ \hat{p}_B + \epsilon \sum_{i=1}^{B-1} \hat{p}_i & \text{for } i = B. \end{cases}$$

Note that this will always result in a different distribution when $\epsilon > 0$, since otherwise $\hat{\mathbf{p}}$ must place all probability on the action B, violating the assumption of $\alpha_\delta^{B,*} < 2 - \frac{1}{B}$. Similar to the previous case, we evaluate the impact of this change on the δ-CR through inspection of the maximization formula (6.45). Since we add mass at most ϵ to the action B, the optimal q_B, which is associated with the largest coefficient in the objective of $\alpha_\delta^{B,\hat{\mathbf{p}}}(B)$, will increase by at most ϵ, compensated by a decrease in the sum of $q_1, \ldots, q_{B-1}$, each of which has a coefficient at least 1. Thus we will have $\alpha_\delta^{B,\tilde{\mathbf{p}}}(B) \leq \alpha_\delta^{B,\hat{\mathbf{p}}}(B) + \epsilon\left(1 - \frac{1}{B}\right)$. On the other hand, consider $i < B$; if $\hat{p}_i = 0$, Lemma 6.C.2 ensures that there will be slack in $\alpha_\delta^{B,\tilde{\mathbf{p}}}(i)$. If instead $\hat{p}_i > 0$, then by a similar argument to the previous part, it holds that $\alpha_\delta^{B,\tilde{\mathbf{p}}}(i) < \alpha_\delta^{B,\hat{\mathbf{p}}}(i)$, since decreasing all of the nonzero $\hat{p}_i$ entries by a multiplicative factor of $(1-\epsilon)$ will in particular decrease the optimal q_i (and possibly other q_j with $j < i$) by the same factor, while increasing q_j associated with smaller coefficients in the objective of (6.45). Together, these bounds imply that ϵ can be chosen such that $\alpha_\delta^{B,\tilde{\mathbf{p}}}(i) < \alpha_\delta^{B,\hat{\mathbf{p}}}(i) \leq \alpha_\delta^{B,*}$ for all $i \in [B]$, contradicting the assumed optimality of $\mathbf{p}$. □

Note that while we cannot validate the δ-CR condition $\alpha_\delta^{B,*} < 2 - \frac{1}{B}$ of the above lemma *a priori* without knowledge of the δ-CR, by part (ii) of the theorem, we can in general use the sufficient condition $\delta < 1 - \frac{c}{\log(B+1)}$ as a heuristic.

With these lemmas proved, we are now prepared to prove the theorem.

Proof of Theorem 6.4.2. We will abbreviate $\mathbf{p}^{B,\delta,*}$ simply as $\mathbf{p}$. We will prove this result in two parts: first, we show that the proposed $\mathbf{p}$, defined as:

$$p_i = \frac{C}{B}\left(1 - \frac{1}{B}\right)^{B-i}$$

for all $i \in [B]$, obtains the claimed competitive bound of $\frac{C-\delta}{1-\delta}$ for the assumed range of δ, giving an upper bound on the δ-CR in this region. After proving this upper bound, we will then prove that $\mathbf{p}$ is, in fact, optimal for this region of δ.

Observe that p_i is increasing in i; thus, $p_1 = \frac{C}{B}\left(1 - \frac{1}{B}\right)^{B-1}$ is the smallest probability assigned by $\mathbf{p}$ to any action, and moreover, by the assumption on the range of δ in the theorem statement, we have $\delta \leq p_1 \leq p_i$ for any $i \in [B]$. Inspecting the maximization form of CVaR_δ in (6.45), one can easily observe that an optimal $\mathbf{q}$ for $\alpha_\delta^{B,\mathbf{p}}(i)$ when $i < B$ will be $q_j = p_j$ for $j < B$ and $q_B = p_B - \delta$, since when $i < B$, q_B is associated with the (smallest) coefficient 1 in the objective, while if $i = B$, the optimal solution will be $q_1 = p_1 - \delta$ and $q_j = p_j$ for $j > 1$, since q_1 is associated with the (smallest) coefficient 1 in the objective. As a result, the CVaR_δ for this range of δ amounts to subtracting δ from the original competitive ratio and normalizing by $1 - \delta$, i.e.,

$$\alpha_\delta^{B,\mathbf{p}}(i) = \frac{1}{1-\delta}\left(\alpha_0^{B,\mathbf{p}}(i) - \delta\right).$$

Since $\alpha_0^{B,\mathbf{p}}(i) = C = \frac{1}{1-(1-1/B)^B}$ for all $i \in [B]$, we obtain the claimed δ-CR: $\alpha_\delta^{B,\mathbf{p}} = \frac{C-\delta}{1-\delta}$.

Now, we turn to proving that this is the optimal δ-CR for $\delta \leq \frac{C}{B}\left(1 - \frac{1}{B}\right)^{B-1}$; henceforth, $\mathbf{p}^\delta$ will refer to the algorithm with optimal δ-CR, which is presumed to be unknown. It must be the case that $\alpha_\delta^{B,\mathbf{p}^\delta}$ is continuous in δ; this is because $\alpha_{\delta+\epsilon}^{B,\mathbf{p}^\delta} - \alpha_\delta^{B,\mathbf{p}^\delta}$ is bounded by a linear function of ϵ, since there exist optimal $\mathbf{q}^{\delta+\epsilon}, \mathbf{q}^\delta$ in the corresponding optimization formulations (6.45) satisfying $\|\mathbf{q}^{\delta+\epsilon} - \mathbf{q}^\delta\|_1 \leq \epsilon$ due to the structure of the optimal solution (i.e., its "filling in" p_i associated with larger costs first, and indifference between p_j associated with the coefficient 1). This in turn implies that $\mathbf{p}^\delta$ ought to be continuous as a function of δ when $\alpha_\delta^{B,\mathbf{p}^\delta} < 2 - \frac{1}{B}$; if this were not the case, i.e., if $\mathbf{p}^\delta$ were discontinuous at some δ, then letting i be the smallest index such that p_i^δ is discontinuous at δ, it is clear from inspection of the maximization form (6.45) that this would introduce a discontinuity in $\alpha_\delta^{B,\mathbf{p}^\delta}$, yielding a contradiction whenever the δ-CR equality (Lemma 6.C.3) is supposed to hold. Note that when $\delta \leq \frac{C}{B}\left(1 - \frac{1}{B}\right)^{B-1}$, $\alpha_\delta^{B,*} \leq \frac{C-\delta}{1-\delta} < 2$, since $\frac{C-\delta}{1-\delta}$ is increasing in δ and

$$\left.\frac{C-\delta}{1-\delta}\right|_{\delta=\frac{C}{B}\left(1-\frac{1}{B}\right)^{B-1}} = \left(\frac{(B-1)^2}{1+\left(1-\frac{1}{B}\right)^B - B} + B\right)^{-1},$$

which is increasing in B and has limit $\frac{e}{e-1}$ as $B \to \infty$. Thus by Lemma 6.C.3, the tightness property must hold on the specified domain of δ: $\alpha_\delta^{B,\mathbf{p}^\delta}(i) = \alpha_\delta^{B,\mathbf{p}^\delta}$ for all $i \in [B]$.

$\mathbf{p}^\delta$ continuous in particular implies that, for any $\epsilon > 0$, there is some non-degenerate interval $[0, \hat{\delta}]$ with the property that, for all $\delta \in [0, \hat{\delta}]$, $p_i^\delta \geq p_0^0 - \epsilon = \frac{C}{B}\left(1 - \frac{1}{B}\right)^{B-1} - \epsilon$. Thus, if we pick δ sufficiently small, the optimal solution of the maximization form of $\alpha_\delta^{B,\mathbf{p}^\delta}(i)$ in (6.45) will be $\mathbf{q} = \mathbf{p}^\delta - \delta \mathbf{e}_j$, where j is an index associated with the cost 1 in the objective, since this is the lowest cost. Defining the cost matrix $\mathbf{M}$ whose ith row contains the cost coefficients when the true ski season duration is i:

$$M_{ij} = \begin{cases} \frac{B+j-1}{i} & \text{if } j \leq i \\ 1 & \text{otherwise,} \end{cases}$$

we thus have the equation

$$\frac{1}{1-\delta}\left(\mathbf{M}\mathbf{p}^\delta - \delta \cdot \mathbf{1}\right) = \alpha_\delta^{B,\mathbf{p}^\delta} \cdot \mathbf{1}$$

when δ is sufficiently small. Rearranging, we have

$$\mathbf{M}\mathbf{p}^\delta = \left((1-\delta)\alpha_\delta^{B,\mathbf{p}^\delta} + \delta\right) \cdot \mathbf{1}. \tag{6.48}$$

Now, notice that (6.48) is of the form $\mathbf{M}\mathbf{p} = c \cdot \mathbf{1}$; this is exactly the equation that arises in the classical discrete-time ski-rental setting when we seek to obtain the optimal algorithm for the (expected cost) competitive ratio (see, e.g., [267]). Since $\mathbf{M}$ is invertible, the unique solution is $\mathbf{p} = c \cdot \mathbf{M}^{-1}\mathbf{1}$, and c must be chosen as $C = \frac{1}{1-(1-\frac{1}{B})^B}$ to ensure normalization of the resulting probability distribution. However, in our setting, this same reasoning implies that $\mathbf{p}^\delta = \left((1-\delta)\alpha_\delta^{B,\mathbf{p}^\delta} + \delta\right) \cdot \mathbf{M}^{-1}\mathbf{1}$, and $(1-\delta)\alpha_\delta^{B,\mathbf{p}^\delta} + \delta = C$ to ensure $\mathbf{p}^\delta$ is a valid probability distribution. As a result, we obtain $\alpha_\delta^{B,\mathbf{p}^\delta} = \frac{C-\delta}{1-\delta}$ and $\mathbf{p}^\delta = \mathbf{p}^0$, as claimed. Finally, note that the preceding argument was made for δ small enough that the optimal solution of the maximization form of $\alpha_\delta^{B,\mathbf{p}^\delta}(i)$ in (6.45) is of the form $\mathbf{q} = \mathbf{p}^\delta - \delta \mathbf{e}_j$, where j is an index associated with the cost 1 in the objective. Because $\mathbf{p}^\delta$ is constant in δ while this condition holds, this condition is seen to be equivalent to $\delta \leq p_0^0 = \frac{C}{B}\left(1 - \frac{1}{B}\right)^{B-1}$. □

6.D Proofs and Additional Results for Section 6.5

Proof of Lemma 6.5.1

We begin by calculating an expression for the inverse CDF of the profit random variable $L \cdot \mathbb{1}_{X>v} + X \cdot \mathbb{1}_{X \leq v}$ in terms of the inverse CDF of the threshold X. As shorthand, we define the cost function $C(X, v) = L \cdot \mathbb{1}_{X>v} + X \cdot \mathbb{1}_{X \le v}$. Then clearly

$$F_{C(X,v)}(y) = \mathbb{P}(C(X,v) \le y) = \begin{cases} \mathbb{P}(X > v) + \mathbb{P}(X \le y) & \text{if } y \le v \\ 1 & \text{otherwise} \end{cases}$$
$$= \begin{cases} 1 - F_X(v) + F_X(y) & \text{if } y \le v \\ 1 & \text{otherwise} \end{cases}$$

and hence, for $t \in [0, 1]$,

$$\begin{aligned} F^{-1}_{C(X,v)}(t) &= \inf\{y \in [L, U] : F_{C(X,v)}(y) \ge t\} \\ &= \begin{cases} L & \text{if } t \le 1 - F_X(v) \\ \inf\{y \in [L, U] : 1 - F_X(v) + F_X(y) \ge t\} & \text{otherwise} \end{cases} \\ &= \begin{cases} L & \text{if } t \le 1 - F_X(v) \\ F_X^{-1}(t - 1 + F_X(v)) & \text{otherwise.} \end{cases} \end{aligned}$$

Using the definition of CVaR_δ as an integral of the inverse CDF (6.2), we have

$$\begin{aligned} &\mathrm{CVaR}_\delta[C(X, v)] \\ &= \frac{1}{1-\delta} \int_0^{1-\delta} F_X^{-1}(t)\,\mathrm{d}t \\ &= \begin{cases} L & \text{if } 1 - \delta \le 1 - F_X(v) \\ \frac{1}{1-\delta}\left[(1 - F_X(v))L + \int_{1-F_X(v)}^{1-\delta} F_X^{-1}(t - 1 + F_X(v))\,\mathrm{d}t\right] & \text{otherwise} \end{cases} \\ &= \begin{cases} L & \text{if } F_X(v) \le \delta \\ \frac{1}{1-\delta}\left[(1 - F_X(v))L + \int_0^{F_X(v)-\delta} F_X^{-1}(t)\,\mathrm{d}t\right] & \text{otherwise,} \end{cases} \end{aligned}$$

as claimed. □

Proof of Theorem 6.5.2

In this proof, we will suppress the sub- and superscript and simply write $\alpha := \alpha_\delta^\theta$. First, note that when $\delta = 1$, the initial condition $\phi(t) = \alpha L = \sqrt{LU}$ holds over the entire interval $[0, 1]$. This is the inverse CDF of the deterministic optimal strategy that always plays the threshold $\sqrt{LU}$, and is hence $\sqrt{\theta}$-competitive; this is easily seen to match the solution to (6.5). In the following, we will restrict to $\delta < 1$.

Let $\phi(t)$ be the solution to the delay differential equation posed in the theorem statement; If we solve it on the region $[\delta, 1]$ by integration, we have

$$\begin{aligned}
\phi(t) &= \begin{cases} \alpha L & \text{for } t \in [0, \delta] \\ \alpha L + \frac{\alpha_\delta^\theta}{1-\delta} \int_\delta^t \phi(s-\delta) - L \, \mathrm{d}s & \text{for } t \in (\delta, 1] \end{cases} \\
&= \begin{cases} \alpha L & \text{for } t \in [0, \delta] \\ \alpha L - \frac{\alpha_\delta^\theta (t-\delta) L}{1-\delta} + \frac{\alpha_\delta^\theta}{1-\delta} \int_0^{t-\delta} \phi(s) \, \mathrm{d}s & \text{for } t \in (\delta, 1] \end{cases} \\
&= \begin{cases} \alpha L & \text{for } t \in [0, \delta] \\ \frac{\alpha}{1-\delta} \left[(1-t)L + \int_0^{t-\delta} \phi(s) \, \mathrm{d}s \right] & \text{for } t \in (\delta, 1]. \end{cases}
\end{aligned} \tag{6.49}$$

ϕ is clearly continuous on $[0, 1]$, and by construction, $\phi(t)$ is also strictly increasing on $[\delta, 1]$, since in this region $\phi'(t) = \frac{\alpha_\delta^\theta}{1-\delta} [\phi(t-\delta) - L] = \frac{\alpha L}{1-\delta}(\alpha - 1) > 0$ since the δ-CR cannot be 1 unless the problem is trivial ($\theta = 1$). Thus, assuming α is chosen such that $\phi(1) = U$, we have that ϕ is one-to-one on the interval $[\delta, 1]$ and $\phi([\delta, 1]) = \phi([0, 1]) = [\alpha L, U]$.

Now, assume that an algorithm uses ϕ as the inverse CDF of its random threshold X, and suppose the adversary chooses a sequence with maximal price $v_{\max} < \alpha L$. In this case, the algorithm will not accept a price during the sequence, since $\phi([0, 1]) = [\alpha L, U]$ implies that X only takes values within the interval $[\alpha L, U]$. In this case, the algorithm earns (deterministic) profit L during the compulsory trade, so the algorithm's δ-CR is simply $\frac{v_{\max}}{L} < \frac{\alpha L}{L} = \alpha$.

On the other hand, suppose the adversary chooses a sequence with maximal price $v_{\max} \geq \alpha L$. Because ϕ is one-to-one on $[\delta, 1]$, ϕ^{-1} exactly coincides with the CDF of X on the domain $[\alpha L, U]$, and for $\phi(\phi^{-1}(v_{\max})) = v_{\max}$. Defining $t = \phi^{-1}(v_{\max})$, noting that $t \geq \delta$, and applying Lemma 6.5.1, we may compute the δ-CR:

$$\begin{aligned}
\frac{v_{\max}}{\mathrm{CVaR}_\delta[L \cdot \mathbb{1}_{X>v} + X \cdot \mathbb{1}_{X \leq v}]} &= \frac{\phi(t)}{\frac{1}{1-\delta} \left[(1 - \phi^{-1}(v_{\max}))L + \int_0^{\phi^{-1}(v_{\max})-\delta} \phi(s) \, \mathrm{d}s \right]} \\
&= \frac{\phi(t)}{\frac{1}{1-\delta} \left[(1-t)L + \int_0^{t-\delta} \phi(s) \, \mathrm{d}s \right]} \\
&= \alpha,
\end{aligned}$$

where the final equality follows from the integral form of $\phi(t)$ in (6.49). Thus ϕ, when used as an inverse CDF for the random threshold, yields an algorithm with δ-CR α.

Now, let us establish an analytic characterization of ϕ for $\delta \in [0, 1]$. When $\delta = 0$, the delay differential equation simplifies to an ordinary differential equation $\phi'(t) = \alpha(\phi(t) - L)$ with initial condition $\phi(0) = \alpha L$. Solving this initial value problem yields the solution $\phi(t) = L + (\alpha - 1)Le^{\alpha t}$, which is easily seen to coincide with (6.7) in the $\delta \to 0$ limit, as the sum in (6.7) becomes the Taylor series of the exponential $e^{\alpha t}$.

On the other hand, if $\delta \in (0, 1)$, then we may solve the delay differential equation by integrating step-by-step. That is, suppose we know $\phi(t - \delta)$ exactly on the interval $[k\delta, (k + 1)\delta]$ for some $k \in \mathbb{N}$, either by the initial condition or because we have solved for $\phi(t)$ on the previous interval $[(k - 1)\delta, k\delta]$. Then we may treat the delay differential equation as an ordinary differential equation on $[k\delta, (k + 1)\delta]$ and solve for $\phi(t)$ accordingly.

We claim that on the interval $[k\delta, (k + 1)\delta]$, ϕ takes the form

$$\phi(t) = L + (\alpha - 1)L \sum_{j=0}^{k} \frac{\alpha^j (t - j\delta)^j}{(1 - \delta)^j j!}. \tag{6.50}$$

Note that this inductive form of ϕ for any nonnegative integer k is equivalent to the form of ϕ posited in the theorem statement in (6.7), as $t \in [k\delta, (k + 1)\delta]$ causes all terms with $j \geq k + 1$ in (6.7) to vanish.

We establish the validity of this form by induction on k, which is exactly the number of step-by-step integrations that must be performed to obtain the solution ϕ on the interval $[k\delta, (k + 1)\delta]$. As the base case, when $k = 0$, (6.50) simply reduces to the initial condition $\phi(t) = \alpha L$ on the interval $[0, \delta]$. Now suppose that the formula holds for a certain k; expressing $\phi(t)$ as an integral of the delay differential equation starting from $(k+1)\delta$, we have that, for $t \in [(k+1)\delta, (k+2)\delta]$,

$$
\begin{aligned}
\phi(t) &= \phi((k+1)\delta) + \frac{\alpha}{1-\delta}\int_{(k+1)\delta}^{t} \phi(s-\delta) - L \,\mathrm{d}s \\
&= L + (\alpha-1)L\sum_{j=0}^{k} \frac{\alpha^j((k+1)\delta - j\delta)^j}{(1-\delta)^j j!} \\
&\quad + \frac{\alpha}{1-\delta}\left[-(t-(k+1)\delta)L + \int_{k\delta}^{t-\delta} \phi(s)\,\mathrm{d}s\right] \qquad (6.51) \\
&= L + (\alpha-1)L\sum_{j=0}^{k} \frac{\alpha^j((k+1-j)\delta)^j}{(1-\delta)^j j!} \\
&\quad + \frac{\alpha}{1-\delta}\Bigg[-(t-(k+1)\delta)L \\
&\qquad\qquad + \int_{k\delta}^{t-\delta} L + (\alpha-1)L\sum_{j=0}^{k}\frac{\alpha^j(s-j\delta)^j}{(1-\delta)^j j!}\,\mathrm{d}s\Bigg] \qquad (6.52) \\
&= L + (\alpha-1)L\sum_{j=0}^{k} \frac{\alpha^j((k+1-j)\delta)^j}{(1-\delta)^j j!} \\
&\quad + \frac{\alpha}{1-\delta}\left[(\alpha-1)L\sum_{j=0}^{k}\frac{\alpha^j(s-j\delta)^{j+1}}{(1-\delta)^j(j+1)!}\right]_{s=k\delta}^{t-\delta} \\
&= L + (\alpha-1)L\sum_{j=0}^{k} \frac{\alpha^j((k+1-j)\delta)^j}{(1-\delta)^j j!} \\
&\quad + (\alpha-1)L\left[\sum_{j=0}^{k}\frac{\alpha^{j+1}(t-(j+1)\delta)^{j+1}}{(1-\delta)^{j+1}(j+1)!} - \sum_{j=0}^{k}\frac{\alpha^{j+1}((k-j)\delta)^{j+1}}{(1-\delta)^{j+1}(j+1)!}\right] \\
&= L + (\alpha-1)L\sum_{j=0}^{k} \frac{\alpha^j((k+1-j)\delta)^j}{(1-\delta)^j j!} \\
&\quad + (\alpha-1)L\left[\sum_{j=1}^{k+1}\frac{\alpha^j(t-j\delta)^j}{(1-\delta)^j j!} - \sum_{j=1}^{k+1}\frac{\alpha^j((k-(j-1))\delta)^j}{(1-\delta)^j j!}\right] \\
&= L + (\alpha-1)L + (\alpha-1)L\sum_{j=1}^{k+1}\frac{\alpha^j(t-j\delta)^j}{(1-\delta)^j j!} \\
&= L + (\alpha-1)L\sum_{j=0}^{k+1}\frac{\alpha^j(t-j\delta)^j}{(1-\delta)^j j!},
\end{aligned}
$$

where (6.51) and (6.52) follow by the induction hypothesis. Thus, by induction, we have established that (6.7) is the solution to the delay differential equation (6.4).

Let us now turn to analyzing the competitive ratio α. First, note that when $\delta \geq \frac{1}{2}$, we have

$$\phi(t) = \alpha L + \frac{\alpha L}{1-\delta}(\alpha - 1)[t-\delta]^+, \tag{6.53}$$

as all terms in (6.7) with $j > 1$ vanish for $t \leq 1$. As α must be chosen so that $\phi(1) = U$, solving this equation for α yields

$$\alpha L + \alpha(\alpha - 1)L = U \implies \alpha = \sqrt{\theta}.$$

We may also obtain an identical upper bound on α for all $\delta \in [0,1)$ (and in particular, $\delta > \frac{1}{5}$) by simply lower bounding ϕ by (6.53), since every term in the sum in (6.7) is nonnegative:

$$\begin{aligned}
U = \phi(1) &\geq \alpha L + \frac{\alpha L}{1-\delta}(\alpha - 1)(1-\delta) \\
&\implies \alpha \leq \sqrt{\theta}.
\end{aligned}$$

This establishes the $\delta > \frac{1}{5}$ case in the analytic bound (6.6); moreover, this case matches the implicit bound $\overline{r}(\delta)$ defined by (6.5) since $\delta > \frac{1}{5}$ implies $\overline{n}(\delta) = \max\left\{1, \left\lfloor \left(\lfloor \delta^{-1} \rfloor - 1\right)/2 \right\rfloor\right\} = 1$, in which case $\overline{r}(\delta)$ is the positive root of the equation $(\overline{r}(\delta) - 1)(\overline{r}(\delta) + 1) = \theta - 1$, i.e., $\sqrt{\theta}$.

On the other hand, suppose that $\delta \leq \frac{1}{5}$. We have that

$$\begin{aligned}
\theta = \frac{\phi(1)}{L} &= 1 + (\alpha - 1)\sum_{j=0}^{\infty} \frac{\alpha^j([1-j\delta]^+)^j}{(1-\delta)^j j!} \\
&= 1 + (\alpha - 1)\sum_{j=0}^{k} \frac{\alpha^j(1-j\delta)^j}{(1-\delta)^j j!} \qquad \text{where } k := \left\lfloor \delta^{-1} \right\rfloor \qquad (6.54) \\
&\geq 1 + (\alpha - 1)\sum_{j=0}^{k} \frac{\alpha^j(1-\frac{j}{k})^j}{(1-\frac{1}{k})^j j!} \qquad (6.55) \\
&= 1 + (\alpha - 1)\sum_{j=0}^{k} \left(\frac{k-j}{k-1}\right)^j \frac{\alpha^j}{j!} \\
&\geq 1 + (\alpha - 1)\sum_{j=0}^{\lfloor (k-1)/2 \rfloor} \left(\frac{k-j}{k-1}\right)^j \frac{\alpha^j}{j!} \qquad (6.56)
\end{aligned}$$

where (6.54) follows from the fact that $j\delta \geq 1$ for all integers $j \geq \left\lfloor \delta^{-1} \right\rfloor + 1$ and $j\delta \leq 1$ for $j \leq \left\lfloor \delta^{-1} \right\rfloor$, and (6.55) follows due to the fact that $\frac{1-j\delta}{1-\delta}$ is decreasing in δ for $\delta < 1$ and $\delta \leq \left(\left\lfloor \delta^{-1} \right\rfloor\right)^{-1} = k^{-1}$. Now, we claim that for all $j \in \{0, \dots, \lfloor (k-1)/2 \rfloor\}$, the following inequality holds:

$$\left(\frac{k-j}{k-1}\right)^j \frac{1}{j!} \geq \binom{\lfloor (k-1)/2 \rfloor}{j} \frac{1}{\lfloor (k-1)/2 \rfloor^j}. \tag{6.57}$$

Note that, by our assumption that $\delta \leq \frac{1}{5}$, we have $k \geq 5$, so $\lfloor (k-1)/2 \rfloor \geq 1$ and the right-hand side of (6.57) is well-defined. To see that this inequality holds, we will bound the ratio between the right-hand side of (6.57) and the left-hand side above by 1. Calling this ratio $R(j)$, observe that

$$\begin{aligned} R(j) = \frac{\binom{\lfloor (k-1)/2 \rfloor}{j} \frac{1}{\lfloor (k-1)/2 \rfloor^j}}{\left(\frac{k-j}{k-1}\right)^j \frac{1}{j!}} &= \frac{\frac{\lfloor (k-1)/2 \rfloor!}{(\lfloor (k-1)/2 \rfloor - j)! \lfloor (k-1)/2 \rfloor^j}}{\left(\frac{k-j}{k-1}\right)^j} \\ &= \frac{(k-1)^j \prod_{i=0}^{j-1} (\lfloor (k-1)/2 \rfloor - i)}{(k-j)^j \lfloor (k-1)/2 \rfloor^j}, \end{aligned}$$

from which it is clear that $R(0) = 1$, $R(1) = 1$, and

$$R(j) = R(j-1) \cdot \frac{(k-j+1)^{j-1}(k-1)(\lfloor (k-1)/2 \rfloor - j + 1)}{(k-j)^j \lfloor (k-1)/2 \rfloor}$$

for $j \geq 2$. Thus, if we can prove that $\frac{(k-j+1)^{j-1}(k-1)(\lfloor (k-1)/2 \rfloor - j+1)}{(k-j)^j \lfloor (k-1)/2 \rfloor} \leq 1$ for each $j \in \{2, \dots, \lfloor (k-1)/2 \rfloor\}$, induction will yield the desired property that $R(j) \leq 1$ for all such j. Thus we compute:

$$\begin{aligned} &\frac{(k-j+1)^{j-1}(k-1)(\lfloor (k-1)/2 \rfloor - j + 1)}{(k-j)^j \lfloor (k-1)/2 \rfloor} \\ &\quad = \left(1 + \frac{1}{k-j}\right)^{j-1} \frac{k-1}{k-j} \cdot \frac{\lfloor (k-1)/2 \rfloor - j + 1}{\lfloor (k-1)/2 \rfloor} \\ &\quad \leq \frac{1}{1 - \frac{j-1}{k-j}} \cdot \frac{k-1}{k-j} \cdot \frac{\lfloor (k-1)/2 \rfloor - j + 1}{\lfloor (k-1)/2 \rfloor} \qquad (6.58) \\ &\quad = \frac{k-1}{k-2j+1} \cdot \frac{\lfloor (k-1)/2 \rfloor - j + 1}{\lfloor (k-1)/2 \rfloor} \\ &\quad \leq \frac{k-1}{k-2j+1} \cdot \frac{(k-1)/2 - j + 1}{(k-1)/2} \qquad (6.59) \\ &\quad = 1 \end{aligned}$$

where (6.58) follows by applying a version of Bernoulli's inequality (e.g., [268, Chapter 3, (1.2)]) to $\left(1 + \frac{1}{k-j}\right)^{-(j-1)}$ and (6.59) follows from the fact that $1 - \frac{1}{x}$ is increasing in x for $x > 0$. Thus, we have established that $R(j) \leq 1$ for all $j \in \{0, \dots, \lfloor (k-1)/2 \rfloor\}$, and hence (6.57) holds for all such j. Applying this inequality to (6.56), we obtain

$$\begin{aligned}\theta &\geq 1 + (\alpha - 1) \sum_{j=0}^{\lfloor (k-1)/2 \rfloor} \left(\frac{k-j}{k-1} \right)^j \frac{\alpha^j}{j!} \\ &\geq 1 + (\alpha - 1) \sum_{j=0}^{\lfloor (k-1)/2 \rfloor} \binom{\lfloor (k-1)/2 \rfloor}{j} \frac{\alpha^j}{\lfloor (k-1)/2 \rfloor^j} \\ &= 1 + (\alpha - 1) \left(1 + \frac{\alpha}{\lfloor (k-1)/2 \rfloor} \right)^{\lfloor (k-1)/2 \rfloor}, \end{aligned} \tag{6.60}$$

where (6.60) follows from the binomial theorem. Note that, since we have assumed $\delta \leq \frac{1}{5}$ in this case, $\overline{n}(\delta)$ as defined in the theorem is equal to $\lfloor (k-1)/2 \rfloor$, which is always at least 2. Thus, observing that (6.60) is increasing in α when $\alpha > 0$, we immediately obtain that the unique positive solution $\overline{r}(\delta)$ to the equality (6.5) is an upper bound on α.

All that remains to be shown is the $\delta \downarrow 0$ case in the analytic bound (6.6); to facilitate this case, we prove the following lemma characterizing the asymptotic behavior of solutions to equations of the general form (6.5) as δ becomes small.

Lemma 6.D.1. *Let $r(\delta)$ be the unique positive solution to the equation*

$$(r(\delta) - 1) \left(1 + \frac{r(\delta)}{n(\delta)} \right)^{n(\delta)} = \theta - 1, \tag{6.61}$$

where $n(\delta)$ is a function satisfying

$$c_1 \delta \leq \frac{1}{n(\delta)} \leq c_2 \delta$$

for all $\delta \in (0, 1]$, given some $c_2 \geq c_1 > 0$. Then $r(\delta) = 1 + W_0 \left(\frac{\theta - 1}{e} \right) + \Theta(\delta)$, where the asymptotic notation reflects the $\delta \downarrow 0$ regime and omits dependence on θ.

The proof of Lemma 6.D.1 relies on the bounds in the following lemma.

Lemma 6.D.2. *Let $C \geq 1$ and and $n \in \mathbb{R}_{++}$. Then*

$$e^{\frac{\log(1+C)}{C} x} \leq \left(1 + \frac{x}{n} \right)^n \leq e^{x - \frac{C - \log(1+C)}{C^2} \frac{x^2}{n}}$$

for all $x \in [0, nC]$.

Proof. We begin by showing that, for $x \in [0, C]$, the inequalities

$$\frac{\log(1+C)}{C} x \leq \log(1+x) \leq x - \frac{C - \log(1+C)}{C^2} x^2 \tag{6.62}$$

hold. First, note that the first inequality in (6.62) holds due to the fact that $\log(1+x)$ is concave and agrees in value with $\frac{\log(1+C)}{C}x$ at both $x = 0$ and C. For the second inequality, first observe that $\log(1+x)$ and $x - \frac{C-\log(1+C)}{C^2}x^2$ agree in value at $x = 0$ and C. Consider their difference

$$f(x) = x - \frac{C - \log(1+C)}{C^2}x^2 - \log(1+x);$$

we will show that $f(x)$ is nonnegative on $[0, C]$. Computing its derivative, we have

$$f'(x) = x\left(-\frac{2}{C} + \frac{1}{1+x} + \frac{2\log(1+C)}{C^2}\right)$$

which has roots

$$x_1 = 0, \quad x_2 = \frac{C^2 + 2\log(1+C) - 2C}{2(C - \log(1+C))},$$

where we can observe $x_2 > 0$, since $C \geq 1$ implies the well-known bounds $C - \frac{1}{2}C^2 < \log(1+C) < C$. Thus, if we can show that $f'(x) > 0$ for all $x \in (0, x_2)$, this will establish the desired property that $f(x) \geq 0$ for all $x \in [0, C]$. Given that f' is continuous on the nonnegative reals and only has roots at 0 and x_2, it suffices to show that f' is positive for some small $\epsilon > 0$. Noting that $C - \frac{1}{2}C^2 < \log(1+C)$ implies $C - \frac{1}{2}C^2 + \delta = \log(1+C)$ for some $\delta > 0$, it follows that

$$\begin{aligned}
f'(x) &= x\left(-\frac{2}{C} + \frac{1}{1+x} + \frac{2\log(1+C)}{C^2}\right)\\
&= x\left(\frac{2\log(1+C) - 2C}{C^2} + \frac{1}{1+x}\right)\\
&= x\left(\frac{-C^2 + 2\delta}{C^2} + \frac{1}{1+x}\right)\\
&= x\left(-1 + \frac{2\delta}{C^2} + \frac{1}{1+x}\right),
\end{aligned}$$

which can be made strictly positive by choosing $x > 0$ sufficiently small, thus establishing the bound.

Multiplying (6.62) by n, exponentiating, and making the substitution $x \leftarrow \frac{y}{n}$, we obtain the bounds

$$e^{\frac{\log(1+C)}{C}y} \leq \left(1 + \frac{y}{n}\right)^n \leq e^{y - \frac{C-\log(1+C)}{C^2}\frac{y^2}{n}}$$

for $\frac{y}{n} \in [0, C]$, i.e., $y \in [0, nC]$. $\square$

With Lemma 6.D.2 proved, we may now proceed with the proof of Lemma 6.D.1.

Proof of Lemma 6.D.1. We begin by establishing coarse lower and upper bounds on the solution $r(\delta)$ to (6.61). First, note that $\theta = 1$ (the trivial case when all prices are identical) yields the unique positive solution $r(\delta) = 1 = \sqrt{\theta} = 1 + W_0\left(\frac{\theta-1}{e}\right)$, regardless of the value of $n(\delta)$. On the other hand, suppose $\theta > 1$. It is clear that $n(\delta) = 1$ implies $r(\delta) = \sqrt{\theta}$ and the $\delta \downarrow 0$ limit (equivalently $n(\delta) \to \infty$, by the assumed bounds on $n(\delta)$) yields $r(0) = 1 + W_0\left(\frac{\theta-1}{e}\right)$. In addition, observe that the left-hand side of (6.61) is continuous and strictly increasing in both $r(\delta)$ and $n(\delta)$ when $r(\delta) > 1$ and $n(\delta) > 0$. Thus, increasing $n(\delta)$ from 1 must yield a decrease in $r(\delta)$. As a result, we must have $r(\delta) \in \left[1 + W_0\left(\frac{\theta-1}{e}\right), \sqrt{\theta}\right]$ for all $\delta \in [0, 1]$.

Now, we continue on to prove the main result. We break the proof into two parts: the upper bound and the lower bound. In the following, we omit the dependence of $r(\delta)$ and $n(\delta)$ on δ, simply writing r and n, respectively.

Upper bound. Since $r \in \left[1 + W_0\left(\frac{\theta-1}{e}\right), \sqrt{\theta}\right]$, we may apply the lower bound in Lemma 6.D.2 with $C = c_2\delta\sqrt{\theta}$ to (6.61) to obtain

$$\theta - 1 = (r-1)\left(1 + \frac{r}{n}\right)^n$$

$$\geq (r-1)\left(1 + \frac{r}{\frac{1}{c_2\delta}}\right)^{\frac{1}{c_2\delta}} \tag{6.63}$$

$$\geq (r-1)e^{\frac{\log(1+c_2\delta\sqrt{\theta})}{c_2\delta\sqrt{\theta}}r}, \tag{6.64}$$

where (6.63) follows by the monotonicity of the left-hand side of (6.61) in n. Monotonicity of (6.64) in $r > 1$ and the definition of the Lambert W function yields the bound

$$\frac{\log(1 + c_2\delta\sqrt{\theta})}{c_2\delta\sqrt{\theta}}(r-1) \leq W_0\left(\frac{\log(1 + c_2\delta\sqrt{\theta})}{c_2\delta\sqrt{\theta}} \cdot \frac{\theta - 1}{\exp\left(\frac{\log(1+c_2\delta\sqrt{\theta})}{c_2\delta\sqrt{\theta}}\right)}\right),$$

and hence

$$r \leq 1 + \frac{c_2\delta\sqrt{\theta}}{\log(1 + c_2\delta\sqrt{\theta})} \cdot W_0\left(\frac{\log(1 + c_2\delta\sqrt{\theta})}{c_2\delta\sqrt{\theta}} \cdot \frac{\theta - 1}{\exp\left(\frac{\log(1+c_2\delta\sqrt{\theta})}{c_2\delta\sqrt{\theta}}\right)}\right).$$

Taylor expanding about $\delta = 0$ gives:

$$r \leq 1 + W_0\left(\frac{\theta-1}{e}\right) + \frac{c_2\sqrt{\theta} \cdot W_0\left(\frac{\theta-1}{e}\right)}{2}\delta + O(\delta^2),$$

i.e., $r = 1 + W_0\left(\frac{\theta-1}{e}\right) + O(\delta)$ as $\delta \downarrow 0$.

Lower bound. Since $r \in \left[1 + W_0\left(\frac{\theta-1}{e}\right), \sqrt{\theta}\right]$, we may apply the upper bound in Lemma 6.D.2 with $C = c_1\delta\sqrt{\theta}$ to (6.61) to obtain

$$\begin{aligned}
\theta - 1 &= (r-1)\left(1 + \frac{r}{n}\right)^n \\
&\leq (r-1)\left(1 + \frac{r}{\frac{1}{c_1\delta}}\right)^{\frac{1}{c_1\delta}} && (6.65) \\
&\leq (r-1)e^{r - \frac{c_1\delta\sqrt{\theta} - \log\left(1 + c_1\delta\sqrt{\theta}\right)}{(c_1\delta\sqrt{\theta})^2} c_1\delta r^2} \\
&\leq (r-1)e^{r - \frac{c_1\delta\sqrt{\theta} - \log\left(1 + c_1\delta\sqrt{\theta}\right)}{(c_1\delta\sqrt{\theta})^2} c_1\delta\left(1 + W_0\left(\frac{\theta-1}{e}\right)\right)^2} && (6.66)
\end{aligned}$$

where (6.65) follows by the monotonicity of the left-hand side of (6.61) in n and (6.66) results from $r \geq 1 + W_0\left(\frac{\theta-1}{e}\right)$ and the fact that $C - \log(1+C) \geq 0$ for $C \geq 0$. Following the same approach as employed in the upper bound, monotonicity of (6.66) in $r > 1$ and the definition of the Lambert W function yields the lower bound

$$r \geq 1 + W_0\left((\theta - 1)\exp\left[\frac{c_1\delta\sqrt{\theta} - \log\left(1 + c_1\delta\sqrt{\theta}\right)}{(c_1\delta\sqrt{\theta})^2} c_1\delta\left(1 + W_0\left(\frac{\theta-1}{e}\right)\right)^2 - 1\right]\right),$$

and Taylor expanding about $\delta = 0$ gives

$$r \geq 1 + W_0\left(\frac{\theta - 1}{e}\right) + \frac{c_1 \cdot W_0\left(\frac{\theta-1}{e}\right)\left(1 + W_0\left(\frac{\theta-1}{e}\right)\right)}{2}\delta + \Omega(\delta^2),$$

i.e., $r = 1 + W_0\left(\frac{\theta-1}{e}\right) + \Omega(\delta)$ as $\delta \downarrow 0$. □

Having proved Lemma 6.D.1, the $\delta \downarrow 0$ case in the analytic bound (6.6) follows as an immediate consequence of the fact that, when $\delta \in (0, 1]$, $\overline{n}(\delta) = \max\left\{1, \left\lfloor\left(\lfloor\delta^{-1}\rfloor - 1\right)/2\right\rfloor\right\}$ can be upper bounded as

$$\begin{aligned}
\max\left\{1, \left\lfloor\left(\lfloor\delta^{-1}\rfloor - 1\right)/2\right\rfloor\right\} &\leq \max\left\{1, \delta^{-1}\right\} \\
&= \delta^{-1}
\end{aligned}$$

and lower bounded as

$$\max\left\{1, \left\lfloor\left(\lfloor\delta^{-1}\rfloor - 1\right)/2\right\rfloor\right\} = \begin{cases} \left\lfloor\left(\lfloor\delta^{-1}\rfloor - 1\right)/2\right\rfloor & \text{if } \delta \le \frac{1}{5} \\ 1 & \text{otherwise} \end{cases}$$

$$\ge \begin{cases} \lfloor 2\lfloor\delta^{-1}\rfloor/5\rfloor & \text{if } \delta \le \frac{1}{5} \\ 1 & \text{otherwise} \end{cases} \tag{6.67}$$

$$\ge \begin{cases} \left\lfloor\frac{8}{25}\delta^{-1}\right\rfloor & \text{if } \delta \le \frac{1}{5} \\ 1 & \text{otherwise} \end{cases} \tag{6.68}$$

$$\ge \begin{cases} \frac{3}{25}\delta^{-1} & \text{if } \delta \le \frac{1}{5} \\ 1 & \text{otherwise} \end{cases} \tag{6.69}$$

$$\ge \frac{3}{25}\delta^{-1},$$

where (6.67), (6.68), and (6.69) hold since $\delta \le \frac{1}{5}$ implies $-1 \ge -\frac{1}{5}\lfloor\delta^{-1}\rfloor \ge -\frac{1}{5}\delta^{-1}$, which in turn implies $\lfloor\delta^{-1}\rfloor \ge \delta^{-1} - 1 \ge \frac{4}{5}\delta^{-1}$ and $\left\lfloor\frac{8}{25}\delta^{-1}\right\rfloor \ge \frac{8}{25}\delta^{-1} - 1 \ge \frac{3}{25}\delta^{-1}$. This concludes the proof. □

Proof of Theorem 6.5.3

The proof of this lower bound follows by establishing a connection between one-max search with the δ-CR metric and the problem of deterministic *k-max search*, which is a modified form of one-max search in which an agent seeks to sell k units of an item, rather than a single one. [79] gives a threshold-based algorithm for k-max search which uses k distinct, increasing price thresholds, with the agent selling its ith item at the first price surpassing the ith threshold. We construct our lower bound by comparing the quantiles of an arbitrary randomized algorithm for one-max search to the price thresholds of [79] for $\lfloor\delta^{-1}\rfloor$-max search. We will assume without loss of generality that $L = 1$ and $U = \theta$. In the following, we define $k := \underline{n}(\delta) = \max\left\{1, \lceil\delta^{-1}\rceil - 1\right\}$ and write the solution to (6.8) as $r := \underline{r}(\delta)$ for clarity. That is, r is the unique positive solution to the equation

$$(r-1)\left(1 + \frac{r}{k}\right)^k = \theta - 1. \tag{6.70}$$

Note that (6.70) has a unique positive solution r since the left-hand side is strictly increasing in r when $r > 0$. First, in the $\delta = 0$ case, (6.70) becomes

$$(r-1)e^r = \theta - 1,$$

whose solution is exactly $r = 1 + W_0\left(\frac{\theta-1}{e}\right)$. Moreover, note that the $\delta = 0$ case is exactly the standard case of randomized one-max search with expected cost, in which case [78] has shown that the optimal competitive ratio is exactly the unique positive solution to $(r-1)e^r = \theta - 1$, thus establishing the validity of our lower bound for $\delta = 0$.[5]

On the other hand, suppose $\delta = 1$; in this case, $k = 1$, and (6.70) becomes

$$(r-1)(1+r) = \theta - 1,$$

yielding the positive solution $r = \sqrt{\theta}$, which matches the optimal deterministic strategy for one-max search [78]. Since the optimal δ-CR coincides with the optimal deterministic CR when $\delta = 1$, this establishes the validity of our bound for this case.

Now, consider an arbitrary $\delta \in (0,1)$; note that, in this case, k simplifies to $k = \lceil \delta^{-1} \rceil - 1$. Define k price thresholds $p_1, \ldots, p_k$ following [79, Lemma 1]:

$$p_i = 1 + (r-1)\left(1 + \frac{r}{k}\right)^{i-1}$$

for $i \in [k]$. Let $X \sim \mu$ be any random threshold algorithm for one-max search supported on $[1, \theta]$ (recall from Section 6.2 that the restriction to such random threshold algorithms is made without loss of generality). For each $i \in [k]$, define $q_i \in [1, \theta]$ as the ith $(k+1)$-quantile of X:

$$q_i = F_X^{-1}\left(\frac{i}{k+1}\right).$$

By definition of the inverse CDF, for each i we have $\mu[1, q_i] \geq \frac{i}{k+1}$ and $\mu[q_i, \theta] \geq 1 - \frac{i}{k+1}$.

Suppose that $q_i > p_i$ for some $i \in [k]$, and let i^* be the smallest index for which this strict inequality holds. If $i^* = 1$, then we have

$$\mu(p_1, \theta] \geq \mu[q_1, \theta] \geq 1 - \frac{1}{k+1} = 1 - \frac{1}{\lceil \delta^{-1} \rceil} \geq 1 - \delta,$$

so the algorithm assigns a probability mass of at least $1 - \delta$ to thresholds strictly greater than p_1. Thus its δ-CR is lower bounded as

$$\alpha_\delta^{\theta,\mu}(p_1) = \frac{p_1}{\mathrm{CVaR}_\delta[1 \cdot \mathbb{1}_{X > p_1} + X \cdot \mathbb{1}_{X \leq p_1}]} \geq \frac{p_1}{1} = r.$$

[5] [78] specifically shows that the solution r to the equation $(r-1)e^r = \theta - 1$ is the optimal competitive ratio for a fractional version of one-max search known as *one-way-trading*, and that randomized one-max search (with expected cost) is equivalent to this fractional version, in the sense that any algorithm for one can be transformed into an algorithm for the other with identical competitive ratio.

Otherwise, if $i^* > 1$, then we have $q_{i^*} > p_{i^*}$ and $q_j \le p_j$ for all $j \in [i^* - 1]$. We define a modified version of the inverse CDF of X as

$$\hat{F}_X^{-1}(t) = \begin{cases} 1 & \text{if } t = 0 \\ q_j & \text{if } t \in (\frac{j-1}{k+1}, \frac{j}{k+1}] \text{ for } j \in [i^* - 1] \\ F_X^{-1}(t) & \text{otherwise,} \end{cases}$$

which is the inverse CDF of the modified random variable $\hat{X}$ obtained by moving all the probability mass between q_{j-1} and q_j to q_j for each $j \in [i^* - 1]$, leaving the rest of the distribution alone. Clearly $F_X^{-1}(t) \le \hat{F}_X^{-1}(t)$ for all $t \in [0, 1]$, since inverse CDFs are increasing and we define $\hat{F}_X^{-1}$ by increasing the value of F_X^{-1} on $(\frac{j-1}{k+1}, \frac{j}{k+1}]$ to $q_j = F_X^{-1}\left(\frac{j}{k+1}\right)$ for $j \in [i^* - 1]$.

Now, suppose the adversary chooses the maximum price as p_{i^*}; then the CVaR_δ of the algorithm's profit is upper bounded as:

$$\begin{aligned}
&\mathrm{CVaR}_\delta[1 \cdot \mathbb{1}_{X > p_{i^*}} + X \cdot \mathbb{1}_{X \le p_{i^*}}] \\
&\quad\le \mathrm{CVaR}_{\frac{1}{k+1}}[1 \cdot \mathbb{1}_{X > p_{i^*}} + X \cdot \mathbb{1}_{X \le p_{i^*}}] && (6.71) \\
&\quad= \frac{1}{1 - \frac{1}{k+1}} \left[1 - F_X(p_{i^*}) + \int_0^{F_X(p_{i^*}) - \frac{1}{k+1}} F_X^{-1}(t)\,\mathrm{d}t\right] && (6.72) \\
&\quad\le \frac{k+1}{k} \left[1 - F_X(p_{i^*}) + \int_0^{F_X(p_{i^*}) - \frac{1}{k+1}} \hat{F}_X^{-1}(t)\,\mathrm{d}t\right] && (6.73) \\
&\quad= \frac{k+1}{k} \left[1 - F_X(p_{i^*}) + \frac{1}{k+1} \sum_{j=1}^{i^*-2} q_j + \left(F_X(p_{i^*}) - \frac{i^* - 1}{k+1}\right) q_{i^*-1}\right] && (6.74) \\
&\quad\le \frac{k+1}{k} \left[1 - \frac{i^*}{k+1} + \frac{1}{k+1} \sum_{j=1}^{i^*-1} q_j\right] && (6.75) \\
&\quad\le \frac{1}{k} \left[k + 1 - i^* + \sum_{j=1}^{i^*-1} p_j\right], && (6.76)
\end{aligned}$$

where (6.71) holds due to the property that CVaR_δ (in the maximization setting) is decreasing in δ [239, Proposition 3.4] and $\delta \ge (\lceil \delta^{-1} \rceil)^{-1} = (k+1)^{-1}$; (6.72) follows from Lemma 6.5.1 and the fact that $i^* > 1$ implies $F(p_{i^*}) \ge F(q_{i^*-1}) \ge \frac{1}{k+1}$; (6.73) follows from $F_X^{-1} \le \hat{F}_X^{-1}$; (6.74) follows by the definition of $\hat{F}_X^{-1}$ and the fact that $p_{i^*} \in [q_{i^*-1}, q_{i^*})$ implies $F_X(p_{i^*}) - \frac{1}{k+1} \in \left[\frac{i^*-2}{k+1}, \frac{i^*-1}{k+1}\right)$; (6.75) is a result of $F_X(p_{i^*}) \le \frac{i^*}{k+1}$ and $q_{i^*-1} \ge 1$; and (6.76) follows by the assumption that, in this case, $q_j \le p_j$ for all $j \in [i^* - 1]$. Substituting the definition of p_j into (6.76) and simplifying the sum, we thus obtain the lower bound

$$\begin{aligned}\alpha_\delta^{\theta,\mu}(p_{i^*}) &= \frac{p_{i^*}}{\mathrm{CVaR}_\delta[1 \cdot \mathbb{1}_{X>p_{i^*}} + X \cdot \mathbb{1}_{X \le p_{i^*}}]} \\ &\ge \frac{k\left(1 + (r-1)\left(1+\frac{r}{k}\right)^{i^*-1}\right)}{k + 1 - i^* + \sum_{j=1}^{i^*-1} 1 + (r-1)\left(1+\frac{r}{k}\right)^{j-1}} \\ &= r.\end{aligned}$$

Finally, consider the case that $q_i \le p_i$ for all $i \in [k]$. Defining

$$\hat{F}_X^{-1}(t) = \begin{cases} 1 & \text{if } t = 0 \\ q_j & \text{if } t \in (\frac{j-1}{k+1}, \frac{j}{k+1}] \text{ for } j \in [k] \\ F_X^{-1}(t) & \text{otherwise,} \end{cases}$$

it is straightforward to see that an argument identical to the previous case gives the following upper bound when the adversary chooses a maximum price of θ:

$$\mathrm{CVaR}_\delta[1 \cdot \mathbb{1}_{X>\theta} + X \cdot \mathbb{1}_{X \le \theta}] \le \frac{1}{k}\sum_{j=1}^{k} p_j.$$

As a result, the δ-CR in this case is lower bounded as

$$\begin{aligned}\alpha_\delta^{\theta,\mu}(\theta) &= \frac{\theta}{\mathrm{CVaR}_\delta[1 \cdot \mathbb{1}_{X>\theta} + X \cdot \mathbb{1}_{X \le \theta}]} \\ &\ge \frac{k\theta}{\sum_{j=1}^{k} 1 + (r-1)\left(1+\frac{r}{k}\right)^{j-1}} \\ &= \frac{r\theta}{1 + (r-1)\left(1+\frac{r}{k}\right)^{k}} \\ &= r && \text{by (6.70).}\end{aligned}$$

Thus, we have established that any random threshold algorithm (and thus *any* algorithm) for one-max search has δ-CR at least r. For the analytic bounds in (6.9), there are two cases: first, the $\delta \ge \frac{1}{2}$ case follows from the fact that $\delta \ge \frac{1}{2}$ implies $k = \max\left\{1, \lceil \delta^{-1} \rceil - 1\right\} = 1$, in which case $r = \sqrt{\theta}$. Second, the $\delta \downarrow 0$ case follows as an immediate consequence of Lemma 6.D.1 upon noting that, for $\delta \in (0, 1]$, $\underline{n}(\delta) = \max\left\{1, \lceil \delta^{-1} \rceil - 1\right\}$ can be upper bounded as

$$\max\left\{1, \lceil \delta^{-1} \rceil - 1\right\} \le \max\left\{1, \delta^{-1}\right\} = \delta^{-1}$$

and lower bounded as

$$\begin{aligned}\max\left\{1, \lceil \delta^{-1} \rceil - 1\right\} &= \begin{cases} \lceil \delta^{-1} \rceil - 1 & \text{if } \delta < \frac{1}{2} \\ 1 & \text{otherwise} \end{cases} \\ &\geq \begin{cases} \frac{2}{3}\lceil \delta^{-1} \rceil & \text{if } \delta < \frac{1}{2} \\ 1 & \text{otherwise} \end{cases} \qquad (6.77) \\ &\geq \begin{cases} \frac{1}{2}\delta^{-1} & \text{if } \delta < \frac{1}{2} \\ 1 & \text{otherwise} \end{cases} \\ &\geq \frac{1}{2}\delta^{-1},\end{aligned}$$

where (6.77) follows from the fact that $\delta < \frac{1}{2}$ implies $\lceil \delta^{-1} \rceil \geq 3$, so $-1 \geq -\frac{1}{3}\lceil \delta^{-1} \rceil$. This concludes the proof. □

Chapter 7

END-TO-END CONFORMAL CALIBRATION FOR OPTIMIZATION UNDER UNCERTAINTY

In the previous chapter, we considered the design of online algorithms when faced with risk-sensitive objectives. We now turn to a complementary question of how we should best *learn* uncertainty estimates for risk-aware decision-making problems. In particular, machine learning can significantly improve performance for decision-making under uncertainty in a wide range of domains. However, ensuring robustness guarantees requires well-calibrated uncertainty estimates, which can be difficult to achieve with neural networks. Moreover, in high-dimensional settings, there may be many valid uncertainty estimates, each with their own performance profile—i.e., not all uncertainty is equally valuable for downstream decision-making. To address this problem, this chapter develops an end-to-end framework to *learn* uncertainty sets for conditional robust optimization in a way that is informed by the downstream decision-making loss, with robustness and calibration guarantees provided by conformal prediction. In addition, we propose to represent general convex uncertainty sets with partially input-convex neural networks, which are learned as part of our framework. Our approach consistently improves upon two-stage estimate-then-optimize baselines on concrete applications in energy storage arbitrage and portfolio optimization.

This chapter is primarily based on the following paper:

[1] C. Yeh*, N. Christianson*, A. Wu, A. Wierman, and Y. Yue. "End-to-End Conformal Calibration for Optimization Under Uncertainty." arXiv: 2409.20534 [cs, math], [Online]. Available: http://arxiv.org/abs/2409.20534.

7.1 Introduction

Well-calibrated estimates of forecast uncertainty are vital for risk-aware decision-making in many real-world systems. For instance, grid-scale battery operators forecast electricity prices to schedule battery charging/discharging to maximize profit, while they rely on uncertainty estimates to minimize financial or operational risk. Similarly, financial investors use forecasts of asset returns with uncertainty estimates to maximize portfolio returns while minimizing downside risk.

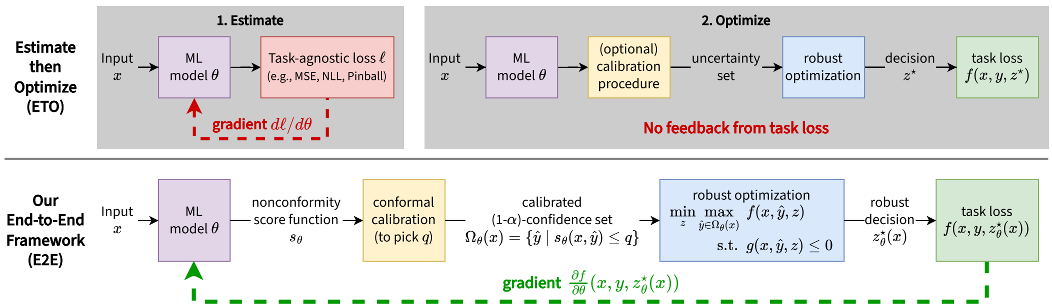


Figure 7.1: Whereas prior "estimate-then-optimize" (ETO, top) approaches separate the model training from the optimization (decision-making) procedure, we propose a framework for end-to-end (E2E, bottom) conformal calibration for optimization under uncertainty that directly trains the machine learning model using gradients from the task loss.

Historically, approaches for decision-making under uncertainty have often treated the estimation of uncertainty separately from its use for downstream decision-making. This "estimate then optimize" (ETO) paradigm [269] separates the problem into an "estimate" stage, where a predictive model is trained to forecast the uncertain quantity, yielding an uncertainty set estimate, followed by an "optimize" stage, where the forecast uncertainty is used to make a decision. Notably, any cost associated with the downstream decision is usually not provided as feedback to the predictive model.

A recent line of work [269–272] has made steps toward bridging the gap between uncertainty quantification and robust optimization-driven decision-making, where optimization problems take a forecast uncertainty set as a parameter, as is common in energy systems [273, 274] and financial applications [275, 276]. However, existing approaches are suboptimal for several reasons:

1. **The predictive model is not trained with feedback from the downstream objective.** Because the downstream objective is often asymmetric with respect to the forecasting model's error, the trained predictive model, though it provides accurate predictions, may yield significantly worse performance on the true decision-making objective.
2. **For the robust optimization to be tractable, the forecast uncertainty sets have restricted parametric forms.** Common parametric forms include box and ellipsoidal uncertainty sets, limiting the expressivity of uncertainty estimates.
3. **Because neural network models are often poor at estimating their own uncertainty, the forecasts may not be well-calibrated.** Recent approaches such as isotonic regression [92] and conformal prediction [277] have made progress

in providing *calibrated* uncertainty estimates from deep learning models, but such methods are typically applied post-hoc to trained models and are therefore difficult to incorporate into an end-to-end training procedure.

As such, there is as of yet *no* comprehensive methodology for training calibrated uncertainty-aware deep learning models end-to-end with downstream decision-making objectives. In this chapter, we provide the first such methodology. We make three specific contributions corresponding to the three issues identified above:

1. **We develop a framework for training prediction models end-to-end with downstream decision-making objectives and conformal-calibrated uncertainty sets in the context of the *conditional robust optimization* problem.** This framework is illustrated in Figure 7.1 (bottom). By including differentiable conformal calibration in our model during training, we close the loop and ensure that feedback from the uncertainty's impact on the downstream objective is accounted for in the training process, since not all model errors nor uncertainty estimates will result in the same downstream cost. This end-to-end training enables the model to focus its learning capacity on minimizing error and uncertainty on outputs with the largest decision-making cost, with more leeway for outputs that have lower costs.
2. **We propose using partially input-convex neural networks (PICNNs) as the nonconformity score function for conformal prediction, enabling the approximate parametrization of arbitrary compact, convex uncertainty sets in the conditional robust optimization problem.** To the best of our knowledge, no existing works use PICNNs to parametrize such arbitrary convex uncertainty sets. Due to the universal convex function approximation property these networks enjoy [278], this approach enables training much more general representations of uncertainty than prior works have considered, which in turn yields substantial improvements on downstream decision-making performance. Importantly, PICNNs are well-matched to our conditional robust optimization problem: we show that the robust problem resulting from this parametrization can be reformulated as a tractable convex optimization problem.
3. **We propose an exact and computationally efficient method to differentiate through the conformal prediction procedure during training.** Unlike prior work [279], our method gives exact gradients, without using approximate ranking and sorting methods.

Finally, we extensively evaluate the performance of our approach on two applications: an energy storage arbitrage task and a portfolio optimization problem. We demonstrate conclusively that the combination of end-to-end training with the flexibility of the PICNN-based uncertainty sets consistently improves over ETO baseline methods. The performance benefit of our end-to-end method is apparent even under distribution shift. Our code is available on GitHub.

7.2 Problem Statement and Background

Our problem is defined formally as follows: suppose that data $(x, y) \in \mathbb{R}^m \times \mathbb{R}^n$ is sampled i.i.d. from an unknown joint distribution $\mathcal{P}$. Upon observing the input x (but not the label y), an agent makes a decision $z \in \mathbb{R}^p$. After the decision is made, the true label y is revealed, and the agent incurs a *task loss* $f(x, y, z)$, for some known task loss function $f : \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$. In addition, the agent's decision must satisfy a set of joint constraints $g(x, y, z) \leq 0$ coupling x, y, and z.

As an illustrative example, consider an agent who would like to minimize the costs of charging and discharging a battery over 24 hours in a day. The agent may use weather forecasts and historical observations x to predict future energy prices y. Based on the predicted prices, the agent decides on the amount z to charge or discharge the battery. The task loss f is the cost incurred by the agent, and the constraints g include limits on how fast the battery can charge as well as the maximum capacity of the battery. This example is explored in more detail in Section 7.5.

Because the agent does not observe the label y prior to making its decision, ensuring good performance and constraint satisfaction requires that the agent makes decisions z that are *robust* to the various outcomes of y. A common objective is to choose z to robustly minimize the task loss and satisfy the constraints over all realizations of y within a $(1-\alpha)$-confidence region $\Omega(x) \subset \mathbb{R}^n$ of the true conditional distribution $\mathcal{P}(y \mid x)$, where $\alpha \in (0, 1)$ is a fixed risk level based on operational requirements. In this case, the agent's robust decision can be expressed as the optimal solution to the following **conditional robust optimization (CRO)** problem [269]:

$$z^\star(x) := \arg\min_{z \in \mathbb{R}^p} \max_{\hat{y} \in \Omega(x)} f(x, \hat{y}, z) \text{ s.t. } g(x, \hat{y}, z) \leq 0. \tag{7.1}$$

After the agent decides $z^\star(x)$, the true label y is revealed, and the agent incurs the task loss $f(x, y, z^\star(x))$. Thus, the agent seeks to minimize expected task loss

$$\mathbb{E}_{(x,y)\sim\mathcal{P}}\left[f(x, y, z^\star(x))\right]. \tag{7.2}$$

While the joint distribution $\mathcal{P}$ is unknown, we assume that we have a dataset $D = \{(x_i, y_i)\}_{i=1}^N$ of i.i.d. samples from $\mathcal{P}$. Then, our objective is to train a machine learning model to learn an approximate $(1 - \alpha)$-confidence set $\Omega(x)$ of possible y values for each input x. Formally, our learned $\Omega(x)$ should satisfy the following *marginal coverage* guarantee.

Definition 7.2.1 (marginal coverage)**.** *An uncertainty set* $\Omega(x)$ *for the distribution* $\mathcal{P}$ *provides* marginal coverage at level $(1 - \alpha)$ *if* $\mathbb{P}_{(x,y)\sim\mathcal{P}}\left(y \in \Omega(x)\right) \geq 1 - \alpha$.

Comparison to related work. The problem of constructing data-driven and machine-learned uncertainty sets with probabilistic coverage guarantees for use in robust optimization has been widely explored in prior literature (e.g., [280–283]). Chenreddy, Bandi, and Delage [269] first coined the phrase "conditional robust optimization" for the problem (7.1) and considered learning context-dependent uncertainty sets $\Omega(x)$ in this setting. However, their approach results in a mixed integer optimization that is intractable to solve for large-scale problems. Moreover, they follow the "estimate then optimize" (ETO) paradigm [284]. As shown in Figure 7.1 (top), the ETO paradigm separates the machine learning model training from the decision optimization. The lack of feedback from the downstream task loss during model training in ETO generally leads to uncertainty sets $\Omega(x)$ which yield suboptimal results. Several other recent papers follow the ETO paradigm using homoskedastic ellipsoidal uncertainty sets [285], heteroskedastic box and ellipsoidal uncertainty sets [286], and a "union of balls" parametrization of uncertainty [270]. In our experiments (Section 7.5), we demonstrate consistent improvements over the methods of Johnstone and Cox [285] and Sun, Liu, and Li [286].

Closest to our work is an "end-to-end" formulation of the CRO problem posed by Chenreddy and Delage [272], which aims to learn conditional uncertainty sets $\Omega(x)$ using a weighted combination of the downstream task loss along with a "conditional coverage loss" to promote calibrated uncertainty. However, they focus solely on ellipsoidal uncertainty sets, and their conditional coverage loss does not provably ensure coverage for their learned uncertainty sets. In our experiments, we do not compare against Chenreddy and Delage [272] because we only consider other methods with a provable coverage guarantee.[1]

[1] As of the time of writing, the approach of Chenreddy and Delage [272] also suffers from substantial inconsistencies between their code implementation and the equations from their paper. In particular, the conditional coverage loss proposed in their paper is not implementable, as it will (almost surely) yield zero gradients.

A concurrent related work by Wang et al. [271] approaches the problem of learning *unconditional* uncertainty sets for robust optimization, while achieving finite sample robust constraint satisfaction guarantees. While their method also uses an end-to-end task loss, we find their use of the same uncertainty set Ω for every problem instance (i.e., Ω is independent of x) to be highly restrictive and unrealistic. For example, in the context of our battery control problem, this restriction would disallow the use of weather forecasts and historical price data to estimate uncertainty in future energy prices. Moreover, they also use restrictive uncertainty set parametrizations such as box, ellipsoidal, and polyhedral uncertainty.

In contrast, our work overcomes these limitations: we incorporate differentiable conformal calibration *during training* to ensure that uncertainty is learned end-to-end in a manner that is both calibrated and minimizes task loss. We apply split conformal post-hoc calibration during inference for provable guarantees on coverage. Furthermore, we use partially input-convex neural networks [287] to directly parameterize the nonconformity score function in conformal prediction, enabling a general and expressive representation of arbitrary conditional convex uncertainty regions that can vary with x and be used efficiently in robust optimization.

Beyond the above closely related work, this chapter builds upon and contributes to several different areas in machine learning and robust optimization; see Section 7.B for a comprehensive discussion.

7.3 End-to-End Training of Conformally Calibrated Uncertainty Sets

In this section, we describe our proposed methodological framework for end-to-end task-aware training of predictive models with conformally calibrated uncertainty for the conditional robust optimization problem (7.1). Our overarching goal is to learn uncertainty sets $\Omega(x)$ which provide $(1-\alpha)$ coverage for any choice of $\alpha \in (0,1)$, and which offer the lowest possible task loss (7.2). To this end, we must consider three primary questions:

1. How should the family of uncertainty sets $\Omega(x)$ be parametrized?
2. How can we guarantee that the uncertainty set $\Omega(x)$ provides coverage at level $1-\alpha$?
3. How can the uncertainty set $\Omega(x)$ be learned to minimize expected task loss?

Figure 7.1 (bottom) illustrates the key parts of our framework to answer these questions. First, we use a machine learning model to parametrize a nonconformity score function s_θ, and we define the uncertainty set $\Omega(x)$ to be a q-sublevel set of $s_\theta(x, \cdot)$. Second, we use conformal calibration to pick q to enforce marginal coverage. Third, we backpropagate gradients through both the robust optimization and conformal calibration steps to update the machine learning model, thereby enabling end-to-end learning. Section 7.3 describe each of these parts in detail, and Algorithm 12 shows pseudocode for both training and inference.

For the rest of the chapter, we make the following assumptions on the functions f and g to ensure tractability of the resulting optimization problem.

Assumption 7.1. *We assume the task loss has the form* $f(x, y, z) = y^\top F(x, z) + \tilde{f}(x, z)$*, where* $F(x, z)$ *is an affine function of* z *and* $\tilde{f}(x, z)$ *is convex in* z*. Furthermore, we assume that* $g(x, y, z) = g(x, z)$ *does not depend on* y *and that* g *is convex in* z*.*

Representations of the uncertainty set

We consider convex uncertainty sets of the form

$$\Omega_\theta(x) = \{\hat{y} \in \mathbb{R}^n \mid s_\theta(x, \hat{y}) \leq q\}, \tag{7.3}$$

where $s_\theta : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ is an arbitrary *nonconformity score function* that is convex in $\hat{y}$, q is a scalar, and θ collects the parameters of a model that we will seek to learn. Note that this representation loses no generality; *any* family of convex sets $\Omega(x)$ can be represented as such a collection of sublevel sets of a partially input-convex function $s(x, \hat{y})$. This particular representation is chosen due to the ease of calibrating sets of this form via conformal prediction to ensure marginal coverage, as we will describe in Section 7.3.

In choosing a particular score function s_θ, one must balance two considerations: first, the *generality* of the sets $\Omega_\theta(x)$ that s_θ can represent, and second, the *tractability* of the resulting robust optimization problem (7.1). We will now show that our representation (7.3) generalizes commonly-used box and ellipsoidal uncertainty sets, which are known to have tractable robust problems; later, in Section 7.4, we will propose to approximate more general convex uncertainty sets using partially input-convex neural networks.

Box uncertainty sets. A simple uncertainty representation is box uncertainty where $\Omega(x) = [\underline{y}(x), \overline{y}(x)]$ is an n-dimensional box whose lower and upper bounds depend on x. Let $h_\theta : \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^n$ be a neural network that estimates lower and upper bounds: $h_\theta(x) = (h_\theta^{\text{lo}}(x), h_\theta^{\text{hi}}(x))$. To represent a box uncertainty set in the form (7.3), we define a nonconformity score function that generalizes scalar conformalized quantile regression [288]:

$$s_\theta(x, y) = \max(\|h_\theta^{\text{lo}}(x) - y\|_\infty, \|y - h_\theta^{\text{hi}}(x)\|_\infty).$$

Then, the uncertainty set (7.3) becomes

$$\Omega_\theta(x) = \left[h_\theta^{\text{lo}}(x) - q\mathbf{1},\ h_\theta^{\text{hi}}(x) + q\mathbf{1}\right] =: \left[\underline{y}(x),\ \overline{y}(x)\right].$$

Given a box uncertainty set $\Omega_\theta(x)$, we can take the dual of the inner maximization problem (see Section 7.C) to transform the robust optimization problem (7.1) into an equivalent form that is convex, and hence tractable, under Assumption 7.1:

$$\begin{aligned} \operatorname*{arg\,min}_{z \in \mathbb{R}^p} \min_{\nu \in \mathbb{R}^n} \quad & (\overline{y}(x) - \underline{y}(x))^\top \nu + \underline{y}(x)^\top F(x, z) + \tilde{f}(x, z) \\ \text{s.t.} \quad & \nu \geq \mathbf{0},\ \nu - F(x, z) \geq \mathbf{0},\ g(x, z) \leq 0. \end{aligned} \tag{7.4}$$

Ellipsoidal uncertainty sets. Another common form of uncertainty set is ellipsoidal uncertainty. Suppose a neural network model $h_\theta : \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{S}_+^n$ predicts mean and covariance parameters $h_\theta(x) = (\mu_\theta(x), \Sigma_\theta(x))$, so that $\hat{\mathcal{P}}(y \mid x; \theta) = \mathcal{N}(y \mid \mu_\theta(x), \Sigma_\theta(x))$ denotes a predicted conditional density, where $\mathcal{N}(\cdot \mid \mu, \Sigma)$ is the multivariate normal density function. In this case, we define the nonconformity score function based on the squared Mahalanobis distance [285, 286]

$$s_\theta(x, y) = (y - \mu_\theta(x))^\top (\Sigma_\theta(x))^{-1} (y - \mu_\theta(x)),$$

which yields uncertainty sets (7.3) that are ellipsoidal:

$$\Omega_\theta(x) = \{\hat{y} \mid (\hat{y} - \mu_\theta(x))^\top (\Sigma_\theta(x))^{-1} (\hat{y} - \mu_\theta(x)) \leq q\}.$$

Let $L_\theta(x)$ denote the unique lower-triangular Cholesky factor of $\Sigma_\theta(x)$ (i.e., $\Sigma_\theta(x) = L_\theta(x) L_\theta(x)^\top$). Taking the dual of the inner maximization problem and invoking strong duality (see Section 7.C), we transform the robust optimization problem (7.1) into an equivalent form that is convex, and hence tractable, under Assumption 7.1:

$$\begin{aligned} \operatorname*{arg\,min}_{z \in \mathbb{R}^p} \quad & \sqrt{q} \|L_\theta(x)^\top F(x, z)\|_2 + \mu_\theta(x)^\top F(x, z) + \tilde{f}(x, z) \\ \text{s.t.} \quad & g(x, z) \leq 0. \end{aligned} \tag{7.5}$$

Conformal uncertainty set calibration

As long as the uncertainty set $\Omega_\theta(x)$ can be expressed in the form (7.3), we can use the split conformal prediction procedure at inference time to choose a value q

Algorithm 12: End-to-end conformal calibration for robust decisions under uncertainty

function $\textsc{Train}$(*training data* $D = \{(x_i, y_i)\}_{i=1}^N$*, uncertainty level* α*, initial model parameters* θ)**:**
  foreach mini-batch $B \subset \{1, \ldots, N\}$ **do**
    Randomly split batch: $B = (B_{\text{cal}}, B_{\text{pred}})$
    Compute $q = \textsc{Quantile}(\{s_\theta(x_i, y_i)\}_{i \in B_{\text{cal}}},\ 1 - \alpha)$
    foreach $i \in B_{\text{pred}}$ **do**
      Solve for robust decision $z_\theta^\star(x_i)$ using (7.4), (7.5), or (7.7)
      Compute gradient of task loss: $d\theta_i = \partial f(x_i, y_i, z_\theta^\star(x_i))/\partial\theta$
    end
    Update θ using gradients $\sum_{i \in B_{\text{pred}}} d\theta_i$
  end

function $\textsc{Inference}$(*model parameters* θ*, calibration data* $D_{cal} = \{(x_i, y_i)\}_{i=1}^M$*, uncertainty level* α*, input* x)**:**
  Compute $q = \textsc{Quantile}(\{s_\theta(\tilde{x}, \tilde{y})\}_{(\tilde{x}, \tilde{y}) \in D_{\text{cal}}},\ 1 - \alpha)$
  return robust decision $z_\theta^\star(x)$ using (7.4), (7.5), or (7.7)

function $\textsc{Quantile}$(*scores* $S = \{s_i\}_{i=1}^M$*, level* β)**:**
  $s_{(1)}, \ldots, s_{(M+1)} = \textsc{SortAscending}(S \cup \{+\infty\})$
  return $s_{(\lceil (M+1)\beta \rceil)}$

that ensures $\Omega_\theta(x)$ provides marginal coverage (Theorem 7.2.1) at any confidence level $1 - \alpha$. The split conformal procedure assumes access to a calibration dataset $D_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^M$ drawn exchangeably from $\mathcal{P}$. We refer readers to Angelopoulos and Bates [90] for details on this procedure.

Lemma 7.3.1 (from Angelopoulos and Bates [90], Appendix D)**.** *Let* $D_{cal} = \{(x_i, y_i)\}_{i=1}^M$ *be a calibration dataset drawn exchangeably (e.g., i.i.d.) from* $\mathcal{P}$*, and let* $s_i = s_\theta(x_i, y_i)$*. If* $q = \textsc{Quantile}(\{s_i\}_{i=1}^M,\ 1 - \alpha)$ *(see Algorithm 12) is the empirical* $\frac{\lceil (M+1)(1-\alpha) \rceil}{M}$*-quantile of the set* $\{s_i\}_{i=1}^M$ *and* (x, y) *is drawn exchangeably with* D_{cal}*, then* $\Omega_\theta(x)$ *has the marginal coverage guarantee*

$$1 - \alpha \leq \mathbb{P}_{x,y,D_{cal}}(y \in \Omega_\theta(x)) \leq 1 - \alpha + \frac{1}{M+1}.$$

We use split conformal prediction, rather than full conformal prediction, both for computational tractability and to avoid the problem of nonconvex uncertainty sets that can arise from the full conformal approach, as noted in Johnstone and Cox [285]. For the rest of this chapter, we assume $\alpha \in [\frac{1}{M+1}, 1)$ so that $q = \textsc{Quantile}(\{s_i\}_{i=1}^M,\ 1 - \alpha) < \infty$ is finite. Thus, for appropriate choices of the score function s_θ, the uncertainty set $\Omega_\theta(x)$ is not unbounded.

While the split conformal prediction procedure in Theorem 7.3.1 ensures that the uncertainty set $\Omega_\theta(x)$ satisfies $(1-\alpha)$ coverage at inference time, this process does not address the question of *training* the uncertainty set $\Omega_\theta(x)$ (via the score function s_θ) to ensure optimal task performance while maintaining coverage. In Section 7.3, we propose applying a separate differentiable conformal prediction procedure during training to address this challenge.

End-to-end training and calibration

Thus far, we have discussed how to calibrate an uncertainty set $\Omega_\theta(x)$ of the form (7.3) to ensure coverage, and we described two choices of score function s_θ parametrizing common box and ellipsoidal uncertainty sets. However, to ensure that the uncertainty sets $\Omega_\theta(x)$ *both* guarantee coverage *and* ensure optimal downstream task performance, it is necessary to design an end-to-end training methodology that can incorporate both desiderata in a fully differentiable manner. We propose such a methodology in Algorithm 12.

Our end-to-end training approach minimizes the empirical task loss $\ell(\theta) = \frac{1}{N}\sum_{i=1}^N \ell_i(\theta)$ using minibatch gradient descent, where $\ell_i(\theta) = f(x_i, y_i, z_\theta^\star(x_i))$. This requires differentiating through both the robust optimization problem as well as the conformal prediction step. The gradient of the task loss on a single instance is $\frac{\mathrm{d}\ell_i}{\mathrm{d}\theta} = \frac{\partial f}{\partial z}|_{(x_i,y_i,z_\theta^\star(x_i))} \frac{\partial z_\theta^\star}{\partial \theta}|_{x_i}$, where $\frac{\partial z_\theta^\star}{\partial \theta}|_{x_i}$ is computed by differentiating through the Karush–Kuhn–Tucker (KKT) conditions of the convex reformulation of the optimization problem (7.1) (i.e., the problems (7.4), (7.5)) following the approach of Amos and Kolter [289], under mild assumptions on the differentiability of f and g. Note that the gradient of any convex optimization problem can be computed with respect to its parameters as such [290, Appendix B].

To include calibration during training, we assume that for every (x, y) in our training set, $s_\theta(x, y)$ is differentiable w.r.t. θ almost everywhere; this assumption holds for common nonconformity score functions, including those used in this chapter. We then adopt the conformal training approach [279] in which a separate q is chosen in each minibatch, as shown in Algorithm 12. The chosen q depends on θ (through s_θ), and $z_\theta^\star(x_i)$ depends on the chosen q. Therefore $\frac{\partial z_\theta^\star}{\partial \theta}$ involves calculating $\frac{\partial z_\theta^\star}{\partial q}\frac{\partial q}{\partial \theta}$, where $\frac{\partial q}{\partial \theta}$ requires differentiating through the empirical quantile function. Whereas Stutz et al. [279] uses a smoothed approximate quantile function for calculating q, we find the smoothing unnecessary, as the gradient of the empirical quantile function is unique and well-defined almost everywhere. Importantly, our exact gradient is

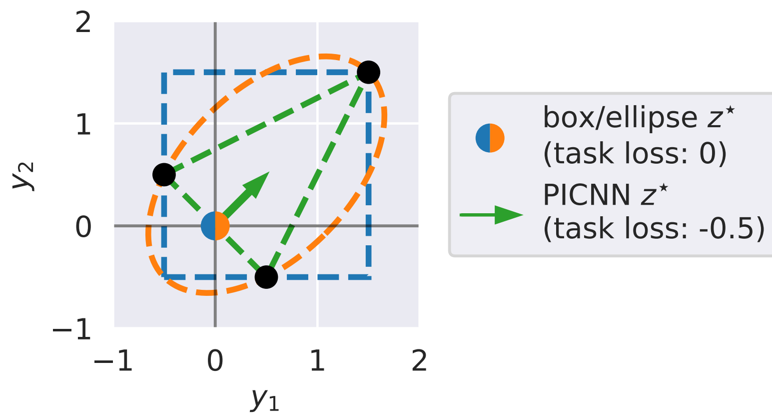


Figure 7.2: Consider a robust portfolio optimization problem with 2 assets, where $y \in \mathbb{R}^2$ is a random vector of asset returns, and the decision $z \in \mathbb{R}^2$ represents portfolio weights: $\max_z \min_{\hat{y}\in\Omega} -z^\top \hat{y}$ s.t. $z \geq 0$, $\mathbf{1}^\top z \leq 1$. Let the distribution of asset returns y be uniform over 3 discrete points (black). The optimal box (blue), ellipse (orange), and PICNN (green) uncertainty sets are shown with their robust decision vectors $z^\star$. The flexibility of the PICNN uncertainty representation allows it to achieve the lowest expected task loss.

both more computationally efficient and simpler to implement than the smoothed approximate quantile approach. See Section 7.D for more details.

After training has concluded and we have performed the final conformal calibration step, the resulting model enjoys the following theoretical guarantee on performance (cf. Sun, Liu, and Li [286, Proposition 1]).

Proposition 7.3.2. *Under the same assumptions as Theorem 7.3.1, the task loss satisfies the following bound with probability at least* $1 - \alpha$ *(over* x, y*, and the calibration set* D_{cal}*):*

$$\begin{aligned} &f(x, y, z_\theta^\star(x)) \\ &\leq \left(\min_{z\in\mathbb{R}^p} \max_{\hat{y}\in\Omega_\theta(x)} f(x, \hat{y}, z) \text{ s.t. } g(x, \hat{y}, z) \leq 0 \right). \end{aligned}$$

Proof. This result is an immediate consequence of the split conformal coverage guarantee of Theorem 7.3.1, which ensures that for the true $(x, y) \sim \mathcal{P}$, $\mathbb{P}_{x,y,D_{\text{cal}}}(y \in \Omega_\theta(x)) \geq 1 - \alpha$, despite the fact that the distribution $\mathcal{P}(y \mid x)$ is unknown. The realized task loss will thus, with probability at least $1 - \alpha$, improve on the optimal value of the robust problem (7.1). □

7.4 Representing General Convex Uncertainty Sets via PICNNs

The previous section discussed how to train calibrated box and ellipsoidal uncertainty sets end-to-end to optimize the downstream task loss. However, both box and ellipsoidal uncertainty sets have restrictive shapes which may yield suboptimal task performance. If $\Omega_\theta(x)$ could represent *any* arbitrary convex uncertainty set, this more expressive class would enable obtaining better task loss. Figure 7.2 illustrates an example where a general convex uncertainty set representation provides a clear advantage over box and ellipsoid uncertainty.

To this end, we propose to directly learn a partially-convex nonconformity score function $s_\theta : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ that is convex only in the second input vector. Fixing x, any q-sublevel set $\{\hat{y} \in \mathbb{R}^n \mid s_\theta(x, \hat{y}) \leq q\}$ of s_θ is a convex set, and likewise every family of convex sets can be expressed as the q-sublevel sets of some partially-convex function. To implement this approach, we are faced with two questions.

1. How should we parametrize the score function s_θ so $\Omega_\theta(x)$ can approximate arbitrary *convex sets?* A natural answer is to parametrize s_θ with a partially input-convex neural network (PICNN) [287], which can efficiently approximate any partially-convex function [278]. We consider a PICNN defined as $s_\theta(x, y) = W_L \sigma_L + V_L y + b_L$, where

$$\begin{aligned} \sigma_0 &= \mathbf{0}, \quad u_0 = x, \quad W_l = \bar{W}_l \operatorname{diag}([\hat{W}_l u_l + w_l]_+) \\ u_{l+1} &= \operatorname{ReLU}(R_l u_l + r_l), \quad V_l = \bar{V}_l \operatorname{diag}(\hat{V}_l u_l + v_l) \\ \sigma_{l+1} &= \operatorname{ReLU}(W_l \sigma_l + V_l y + b_l), \quad b_l = \bar{B}_l u_l + \bar{b}_l, \end{aligned} \tag{7.6}$$

with weights $\theta = (R_l, r_l, \bar{W}_l, \hat{W}_l, w_l, \bar{V}_l, \hat{V}_l, v_l, \bar{B}_l, \bar{b}_l)_{l=0}^L$. The matrices $\bar{W}_l$ are constrained to be entrywise nonnegative to ensure convexity of s_θ with respect to y. For ease of notation, we assume all hidden layers $\sigma_1, \ldots, \sigma_L$ have the same dimension d.

2. Does the chosen parametrization of $\Omega_\theta(x)$ (via PICNNs) yield a tractable reformulation of the CRO problem (7.1)*?* Fortunately, we show in the following theorem that the answer is yes.

Theorem 7.4.1. *Let $\Omega_\theta(x) = \{\hat{y} \in \mathbb{R}^n \mid s_\theta(x, \hat{y}) \leq q\}$, where s_θ is a PICNN as defined in* (7.6)*. Then, under Assumption 7.1, the CRO problem* (7.1) *with uncertainty set $\Omega_\theta(x)$ is equivalent to the following convex (and hence tractable) minimization problem:*

$$\begin{aligned} \arg\min_{z \in \mathbb{R}^p} \min_{\nu \in \mathbb{R}^{2Ld+1}} \quad & b(\theta, q)^\top \nu + \tilde{f}(x, z) \\ \text{s.t.} \quad & A(\theta)^\top \nu = \begin{bmatrix} F(x, z) \\ \mathbf{0} \end{bmatrix}, \ \nu \geq \mathbf{0}, \ g(x, z) \leq 0 \end{aligned} \tag{7.7}$$

where $A(\theta) \in \mathbb{R}^{(2Ld+1)\times(n+Ld)}$ *and* $b(\theta, q) \in \mathbb{R}^{2Ld+1}$ *are constructed from the weights* θ *of the PICNN* (7.6)*, and* b *also depends on* q*.*

We prove Theorem 7.4.1 in Section 7.C; the main idea is that when $\Omega_\theta(x)$ is a sublevel set of a PICNN, we can equivalently reformulate the inner maximization problem in (7.1) as a linear program and take the dual to yield a tractable minimization problem.

Since the PICNN uncertainty sets are of the same form as (7.3) and yield a tractable convex reformulation (7.7) of the CRO problem (7.1), we can apply the split conformal procedure detailed in Section 7.3 to choose $q \in \mathbb{R}$ and obtain coverage guarantees on $\Omega_\theta(x)$, and we can employ the same end-to-end training methodology from Section 7.3 to train calibrated uncertainties end-to-end using the downstream task loss. In some cases during training, the inner maximization problem of (7.1) with PICNN-parametrized uncertainty set may be unbounded (if $\Omega_\theta(x)$ is not compact) or infeasible (if the chosen q is too small causing $\Omega_\theta(x)$ to be empty). This will lead, respectively, to an infeasible or unbounded equivalent problem (7.7). We can avoid this concern by adjusting the PICNN architecture to ensure its sublevel sets are compact and by suitably increasing q when needed to ensure $\Omega_\theta(x)$ is never empty. Such modifications do not change the general form of the problem (7.7) and preserve the marginal coverage guarantee for the uncertainty set $\Omega_\theta(x)$; see Section 7.C for details.

7.5 Experiments

In this section, we present experimental results for our E2E method against several ETO baselines. Code to reproduce our results are provided in the supplementary materials.

Problem descriptions

We consider two tasks: price forecasting for battery storage operation and portfolio optimization. Their task loss functions and constraints satisfy Assumption 7.1.

Price forecasting for battery storage. This problem comes from Donti, Amos, and Kolter [291], where a grid-scale battery operator predicts electricity prices $y \in \mathbb{R}^T$ over a T-step horizon and uses the predicted prices to decide a battery charge/discharge schedule for price arbitrage. The input features x include the past day's prices and temperature, the next day's energy load forecast and temperature forecast, binary indicators of weekends or holidays, and yearly sinusoidal features. The operator decides how much to charge ($z^{\text{in}} \in \mathbb{R}^T$) or discharge ($z^{\text{out}} \in \mathbb{R}^T$) the

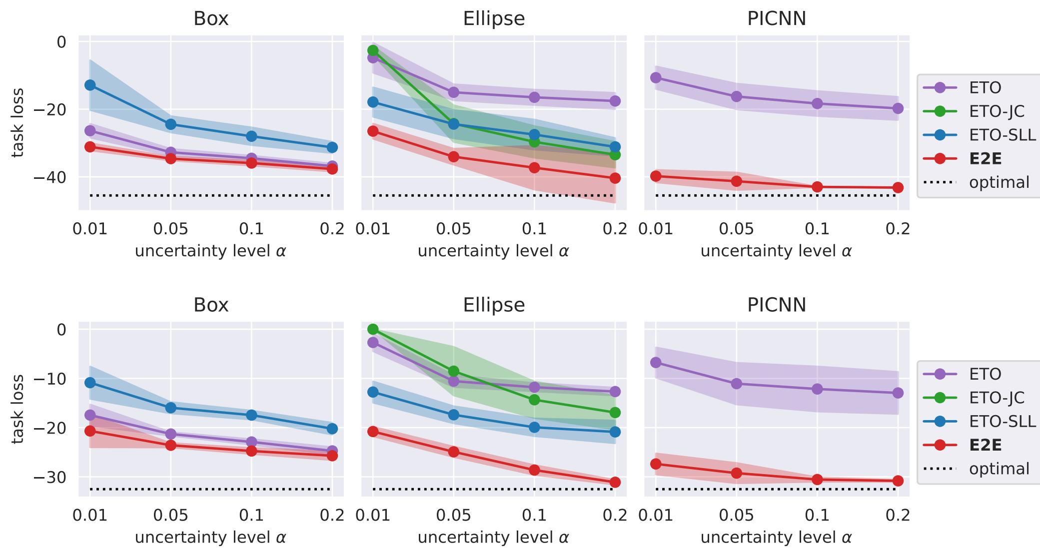


Figure 7.3: Task loss performance (mean ±1 stddev across 10 runs) for the battery storage problem with no distribution shift (top) and with distribution shift (bottom). Lower values are better.

battery, which changes the battery's state of charge ($z^{\text{state}} \in \mathbb{R}^T$). The battery has capacity B, charging efficiency γ, and maximum charging/discharging rates c^{in} and c^{out}. The task loss function represents the multiple objectives of maximizing profit, flexibility to participate in other markets by keeping the battery near half its capacity (with weight λ), and battery health by discouraging rapid charging/discharging (with weight ϵ):

$$f(y, z) = \sum_{t=1}^{T} y_t (z^{\text{in}} - z^{\text{out}})_t + \lambda \|z^{\text{state}} - \frac{B}{2}\mathbf{1}\|^2 + \epsilon \|z^{\text{in}}\|^2 + \epsilon \|z^{\text{out}}\|^2.$$

The constraints are, for all $t = 1, \dots, T$,

$$\begin{aligned} &z_0^{\text{state}} = B/2, \quad z_t^{\text{state}} = z_{t-1}^{\text{state}} - z_t^{\text{out}} + \gamma z_t^{\text{in}}, \\ &0 \le z^{\text{in}} \le c^{\text{in}}, \quad 0 \le z^{\text{out}} \le c^{\text{out}}, \quad 0 \le z_t^{\text{state}} \le B. \end{aligned}$$

Following Donti, Amos, and Kolter [291], we set $T = 24$, $B = 1$, $\gamma = 0.9$, $c^{\text{in}} = 0.5$, $c^{\text{out}} = 0.2$, $\lambda = 0.1$, and $\epsilon = 0.05$.

Portfolio optimization. We adopt the portfolio optimization setting and synthetic dataset from Chenreddy and Delage [272], where the prediction targets $y \in \mathbb{R}^n$ are the returns of a set of n securities, and the decision $z \in \mathbb{R}^n$ sets portfolio weights. The task loss is $f(y, z) = -y^\top z$, with constraints $z \ge 0$, $\mathbf{1}^\top z = 1$. The synthetic dataset consists of $(x, y) \in \mathbb{R}^{2\times 2}$ drawn from a mixture of three 4-D multivariate normal distributions. For these experiments, we provide details about the data in

Section 7.E and experimental results in Section 7.A. The results are similar to those for battery storage, except that portfolio optimization is a lower dimensional and easier problem.

Baseline methods

We implemented several "estimate-then-optimize" (ETO) baselines, listed below, to compare against our end-to-end (E2E) method. These two-stage ETO baselines are trained using task-agnostic losses such as pinball loss or negative log-likelihood (NLL). To ensure a fair comparison against our E2E method, we also apply conformal calibration to each ETO method after training to satisfy coverage.

- **ETO** denotes models with identical neural network architectures to our E2E models, differing only in the loss function during training. The box uncertainty **ETO** model is trained with pinball loss to predict the $\frac{\alpha}{2}$ and $1 - \frac{\alpha}{2}$ quantiles. The ellipsoidal uncertainty **ETO** model is trained with a multivariate normal negative log-likelihood loss. For the PICNN **ETO** model, we train s_θ using a negative log-likelihood loss by interpreting s_θ as an energy function—i.e., $\hat{\mathcal{P}}_\theta(y \mid x) \propto \exp(-s_\theta(x, y))$—yielding the loss $\mathrm{NLL}(\theta) = \ln s_\theta(x, y) + \ln Z_\theta(x)$, where $Z_\theta(x) = \int_{\tilde{y} \in \mathbb{R}^n} \exp(-s_\theta(x, \tilde{y}))\, \mathrm{d}\tilde{y}$, following the approach of Lin and Ba [292]. More details of the **ETO** models are given in Section 7.E.
- **ETO-SLL** is our implementation of the box and ellipsoid uncertainty ETO methods from Sun, Liu, and Li [286]. Unlike **ETO**, **ETO-SLL** first trains a point estimate model (without uncertainty) with mean-squared error loss. Then, **ETO-SLL** box and ellipsoidal uncertainty sets are derived from training a separate quantile regressor using pinball loss to predict the $(1-\alpha)$-quantiles of absolute residuals or ℓ_2-norm of residuals of the point estimate. Unlike **ETO** which can learn ellipsoidal uncertainty sets with different covariance matrices for each input x, the **ETO-SLL** ellipsoidal uncertainty sets all share the same covariance matrix (up to scale).
- **ETO-JC** is our implementation of the ellipsoid uncertainty ETO method by Johnstone and Cox [285]. Like **ETO-SLL**, **ETO-JC** also first trains a point estimate model (without uncertainty) with mean-squared error loss. **ETO-JC** uses the same covariance matrix (with the same scale) for each input x.

Battery storage problem results

Figure 7.3 (top) compares task loss performance for different uncertainty levels ($\alpha \in \{.01, .05, .1, .2\}$) and the different uncertainty set representations for the ETO baselines against our proposed E2E methodology on the battery storage problem with no distribution shift. Our E2E approach consistently yields improved performance over all ETO baslines, for all three uncertainty set parametrizations, and over all tested uncertainty levels α. Moreover, the PICNN uncertainty representation, when trained end-to-end, provides up to 42% relative improvement in performance over the best ETO box uncertainty set and up to 209% relative improvement over the best ETO ellipse uncertainty set. We additionally show the corresponding coverage obtained by the learned uncertainty sets in Figure 7.4 (top); all models obtain coverage close to the target level, confirming that the improvements in task loss performance from our E2E approach do not come at the cost of worse coverage.

Performance under distribution shift

The aforementioned results were produced without distribution shift, where our training and test sets were sampled uniformly at random, thus ensuring exchangeability and guaranteeing marginal coverage. In this section, we evaluate our method on the more realistic setting with distribution shift by splitting our data temporally; our models are trained on the first 80% of days and evaluated on the last 20% of days. Figures 7.3 (bottom) and 7.4 (bottom) mirror Figures 7.3 (top) and 7.4 (top), except that there is now distribution shift. We again find that our E2E approach consistently yields improved performance over all ETO baselines, for all three uncertainty set parametrizations, and for all tested uncertainty levels α. Likewise, the PICNN uncertainties, when trained end-to-end, improve on the performance offered by box and ellipsoidal uncertainty. We find unsurprisingly that, under distribution shift, the models do not provide the same level of coverage guaranteed in the i.i.d. case, as the exchangeability assumption needed for conformal prediction no longer holds. The ellipsoidal and PICNN models tend to provide worse coverage than the box uncertainty, which we believe reflects how ellipsoidal and PICNN uncertainty sets offer greater representational power, and thus might be fitting too closely to the pre-shift distribution, which impacts robustness under distribution shift. Devising methods to anticipate distribution shift when training these more expressive models, and in particular the PICNN-based uncertainty, remains an interesting avenue for future work.

7.6 Conclusion

In this chapter, we develop the first end-to-end methodology for training predictive models with uncertainty estimates (with calibration enforced differentiably throughout training) that are utilized in downstream conditional robust optimization problems. We demonstrate an approach utilizing partially input-convex neural networks (PICNNs) to represent general convex uncertainty regions, and we perform extensive experiments on a battery storage application and a portfolio optimization task. Whereas prior works on two-stage estimate-then-optimize approaches emphasized "the convenience brought by the disentanglement of the prediction and the uncertainty calibration" [286], our results highlight that such "convenience" comes at a substantial cost; our end-to-end approach, combined with the expressiveness of the PICNN representation, has clear performance gains over the traditional two-stage methods.

A number of interesting directions for future work on learning decision-aware uncertainty in an end-to-end manner remain. First, while our PICNN-based uncertainty set representation allows the parametrization of general convex uncertainty sets, future work may explore *nonconvex* uncertainty regions. Doing so may require eschewing the analytical methods for differentiating through convex optimization problems and instead use, e.g., policy gradient methods for passing gradients through general stochastic and robust optimization problems. Second, developing end-to-end methods to target conditional calibration (as opposed to marginal calibration) remains an active research direction. Finally, one may explore other types of constraints besides uncertainty set-based robustness, such as value-at-risk (VaR) or conditional value-at-risk (CVaR) constraints.

Appendix

In these appendix sections, we present additional experimental results and descriptions, we discuss our contributions in the context of related work, and we provide theoretical details and proofs underlying our proposed methodology.

7.A Additional Experimental Results

Experimental results: Battery storage

The optimal task losses shown in black dotted lines in Figure 7.3 are the lowest average achievable task loss on the test set given perfect knowledge of the target y. The optimal task loss is calculated for each example (x, y) in the test set as $f(x, y, z^\star_{\text{opt}})$ where

$$z^\star_{\text{opt}} = \operatorname*{arg\,min}_{z \in \mathbb{R}^p} f(x, y, z) \text{ s.t. } g(x, y, z) \leq 0.$$

Figure 7.4 plots the marginal coverage of the different uncertainty sets across four levels of α. As discussed in Section 7.5, coverage levels stay consistent between the

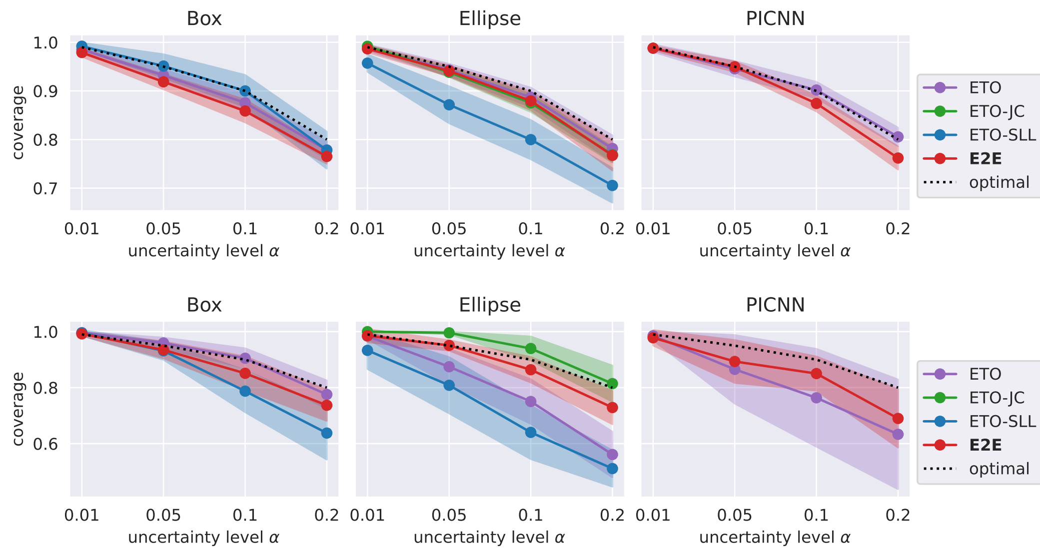


Figure 7.4: Coverage (mean ±1 stddev across 10 runs) for the battery storage problem with no distribution shift (top) and with distribution shift (bottom). The dotted black line indicates the target coverage level $1 - \alpha$. Our E2E models achieve similar coverage to the ETO baselines, confirming that the lower task loss of our E2E models does not come at the expense of worse coverage.

ETO baselines and our E2E models, confirming that the lower task loss of our E2E models does not come at the expense of worse coverage.

Experimental results: Portfolio optimization

Tables 7.1 and 7.2 show the task loss and coverage results for the portfolio optimization problem. We again find that our E2E approach generally improves upon the ETO baselines at all uncertainty levels α, with the exception of box uncertainty where all the methods achieve similar performance. Our PICNN-based uncertainty representation, when learned end-to-end, performs better than box uncertainty and comparably with ellipse uncertainty. The similarity in performance between E2E ellipsoidal uncertainty and E2E PICNN uncertainty is likely due to the underlying aleatoric uncertainty (i.e., the uncertainty in $\mathcal{P}(y \mid x)$) generally taking an ellipsoidal shape—the conditional distribution $\mathcal{P}(y \mid x)$ is a Gaussian mixture model, and it tends to have a dominant mode (see, e.g., Figure 7.5). In terms of coverage, we find that all the models and training methodologies obtain coverage very close to the target level, confirming that the improvements in task loss performance from our E2E approach do not come at the cost of worse coverage.

Because the conditional distribution $\mathcal{P}(y \mid x)$ for the portfolio optimization problem is 2-dimensional, we can visualize the conditional distribution as well as the

Table 7.1: Task loss performance (mean ± 1 stddev across 10 runs) for the portfolio optimization problem. Lower values are better, and the best performance for each uncertainty-level α is highlighted. The results show that our E2E methods consistently outperform the ETO baselines.

		uncertainty level α 0.01	0.05	0.1	0.2
ETO	Box	-1.16 ± 0.42	-1.37 ± 0.12	-1.39 ± 0.13	-1.41 ± 0.12
ETO	Ellipse	-1.09 ± 0.12	-1.24 ± 0.11	-1.29 ± 0.10	-1.33 ± 0.10
ETO	PICNN	-0.95 ± 0.24	-1.11 ± 0.24	-1.20 ± 0.22	-1.31 ± 0.16
ETO-SLL	Box	-1.41 ± 0.13	-1.42 ± 0.12	-1.42 ± 0.12	-1.44 ± 0.11
ETO-SLL	Ellipse	-1.12 ± 0.22	-1.37 ± 0.12	-1.40 ± 0.12	-1.43 ± 0.12
ETO-JC	Ellipse	-1.16 ± 0.17	-1.40 ± 0.11	-1.42 ± 0.11	-1.44 ± 0.11
E2E	Box	-1.21 ± 0.44	-1.40 ± 0.14	-1.43 ± 0.11	-1.43 ± 0.10
E2E	Ellipse	**-1.48** ± 0.12	-1.47 ± 0.11	**-1.48** ± 0.11	**-1.47** ± 0.11
E2E	PICNN	-1.45 ± 0.14	**-1.48** ± 0.10	**-1.48** ± 0.10	**-1.47** ± 0.11

uncertainty sets estimated by our models. Figure 7.5 plots the conditional density for input $x = \begin{bmatrix}-1.167 & 0.024\end{bmatrix}^\top$, along with the $\alpha = 0.1$ uncertainty sets $\Omega_\theta(x)$ and the resulting decision vectors $z^\star_\theta(x)$ for each uncertainty set parametrization. Uncertainty sets and decision vectors from both **ETO** and E2E models are shown in different colors. The key takeaway from this figure is that smaller uncertainty sets (which is what ETO training tends to produce) do not always result in lower task loss. Furthermore, the more flexible parametrization of the PICNN allows it to learn uncertainty set shapes that may be more amenable to the downstream robust decision task than box or ellipsoidal uncertainty, even if the resulting uncertainty set has a larger or odder shape.

7.B Related Work

Task-based learning. The notion of "task-based" end-to-end model learning was introduced by Donti, Amos, and Kolter [291], which proposed to train machine learning models end-to-end in a manner capturing a downstream stochastic optimization task. To achieve this, the authors backpropagate gradients through a stochastic optimization problem, which is made possible for various types of convex optimization problems via the implicit function theory [289–291]. However, Donti, Amos, and Kolter [291] does not train the model to estimate uncertainty and thereby does not provide any explicit guarantees on robustness on their decisions to uncertainty. Our framework improves upon this baseline by yielding calibrated uncertainty sets which can then be used to obtain robust decisions.

Table 7.2: Coverage (mean ±1 stddev across 10 runs) for the portfolio optimization problem. Our E2E models achieve similar coverage to the ETO baselines, confirming that the lower task loss of our E2E models does not come at the expense of worse coverage.

		uncertainty level α			
		0.01	0.05	0.1	0.2
`ETO`	Box	.984 ± .007	.947 ± .017	.902 ± .017	.786 ± .020
`ETO`	Ellipse	.988 ± .004	.944 ± .020	.894 ± .022	.794 ± .027
`ETO`	PICNN	.989 ± .006	.949 ± .014	.901 ± .019	.801 ± .034
`ETO-SLL`	Box	.985 ± .012	.945 ± .021	.885 ± .030	.796 ± .029
`ETO-SLL`	Ellipse	.989 ± .011	.945 ± .024	.885 ± .039	.795 ± .030
`ETO-JC`	Ellipse	.991 ± .006	.953 ± .017	.902 ± .026	.796 ± .026
E2E	Box	.989 ± .006	.949 ± .012	.903 ± .016	.785 ± .019
E2E	Ellipse	.992 ± .006	.954 ± .010	.903 ± .022	.798 ± .022
E2E	PICNN	.993 ± .002	.953 ± .010	.912 ± .017	.798 ± .024

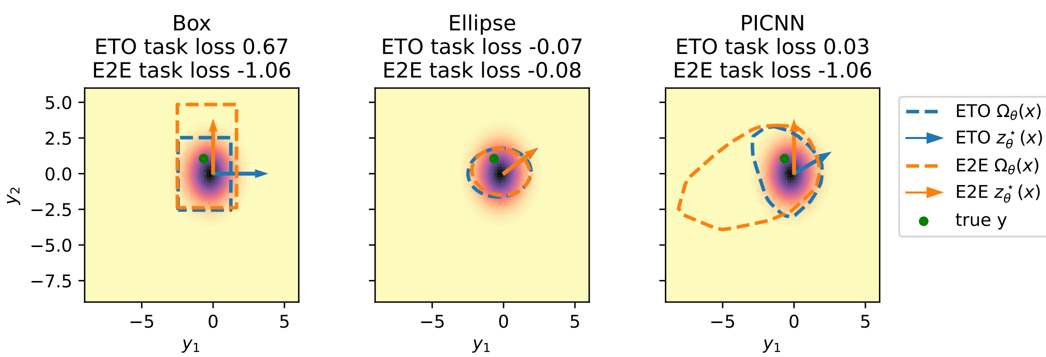


Figure 7.5: This figure plots the density of the conditional distribution $\mathcal{P}(y \mid x)$ for $x = \begin{bmatrix}-1.167 & 0.024\end{bmatrix}^\top$ from the portfolio optimization problem, with darker colors indicating higher density. Also plotted are the $\alpha = 0.1$ uncertainty sets $\Omega_\theta(x)$ (dashed lines) and the resulting decision vectors $z^\star_\theta(x)$ (arrows) for each uncertainty set parametrization. Results for **`ETO`** models are shown in blue, whereas results for E2E are shown in orange. The "true" y sampled from $\mathcal{P}(y \mid x)$ is drawn in green, and the task loss for this example is computed using this y. The decision vectors have been artificially scaled larger to be easier to see.

Uncertainty Quantification. Various designs for deep learning regression models that provide uncertainty estimates have been proposed in the literature, including Bayesian neural networks [293, 294], Gaussian process regression and deep kernel learning [295–297], ensembles of models [298], and quantile regression [288], among other techniques. These methods typically only provide heuristic uncertainty estimates that are not necessarily well-calibrated [299].

Post-hoc methods such as isotonic regression [92] or conformal prediction [277] may be used to calibrate the uncertainty outputs of deep learning models. These calibration methods generally treat the model as a black box and scale predicted uncertainty levels so that they are calibrated on a held-out calibration set. Isotonic regression guarantees calibrated outputs in the limit of infinite data, whereas conformal methods provide probabilistic, finite-sample calibration guarantees when the calibration set is exchangeable (e.g., drawn i.i.d. from the same distribution) with test data. These calibration methods are generally not included in the model training procedure because they involve non-differentiable operators, such as sorting. However, recent works have proposed differentiable losses [279, 300] that approximate the conformal prediction procedure during training and thus allow end-to-end training of models to output more calibrated uncertainty. As approximations, these methods lose the marginal coverage guarantees that true conformal methods provide. However, such guarantees can be recovered at test time by replacing the approximations with true conformal prediction.

Robust and stochastic optimization. The optimization community has proposed a number of techniques over the years to improve robust decision-making under uncertainty, including stochastic, risk-sensitive, chance-constrained, distributionally robust, and robust optimization (e.g., [260, 301–303]). These techniques have been applied to a wide range of applications, including energy systems operation [29, 41, 42, 45, 46, 48, 304, 305] and portfolio optimization [275, 276, 281]. In these works, the robust and stochastic optimization methods enable selecting decisions (grid resource dispatches or portfolio allocations) in a manner that is aware of uncertainty, e.g., so an energy system operator can ensure that sufficient generation is available to meet demand even on a cloudy day without much solar generation. Typically, however, the construction of uncertainty sets, estimated probability distributions over uncertain parameters, or ambiguity sets over distributions takes place offline and is unconnected to the eventual decision-making task. Thus, our proposed end-to-end approach allows for simultaneous calibration of uncertainty sets with optimal decision-making.

7.C Maximizing Over the Uncertainty Set

We consider robust optimization problems of the form

$$\min_{z \in \mathbb{R}^p} \max_{\hat{y} \in \mathbb{R}^n} \hat{y}^\top F z + \tilde{f}(x, z) \qquad \text{s.t.} \qquad \hat{y} \in \Omega(x), \quad g(x, z) \leq 0.$$

For fixed z, the inner maximization problem is

$$\max_{\hat{y}\in\mathbb{R}^n} \hat{y}^\top F z \qquad \text{s.t.} \qquad \hat{y} \in \Omega(x),$$

which we analyze in the more abstract form

$$\max_{y\in\mathbb{R}^n} c^\top y \qquad \text{s.t.} \qquad y \in \Omega$$

for arbitrary $c \in \mathbb{R}^n \setminus \{0\}$. The subsections of this appendix derive the dual form of this maximization problem for specific representations of the uncertainty set Ω.

Suppose y is standardized or whitened by an affine transformation with $\mu \in \mathbb{R}^n$ and invertible matrix $W \in \mathbb{R}^{n\times n}$

$$y_{\text{transformed}} = W^{-1}(y - \mu)$$

so that Ω is an uncertainty set on the transformed $y_{\text{transformed}}$. Then, the original primal objective can be recovered as

$$c^\top y = c^\top (W y_{\text{transformed}} + \mu) = (Wc)^\top y_{\text{transformed}} + c^\top \mu.$$

In our experiments, we use element-wise standardization of y by setting $W = \operatorname{diag}(y_{\text{std}})$, where $y_{\text{std}} \in \mathbb{R}^n$ is the element-wise standard-deviation of y.

Maximizing over a box constraint

Let $[\underline{y}, \overline{y}] \subset \mathbb{R}^n$ be a box uncertainty set for $y \in \mathbb{R}^n$. Then, for any vector $c \in \mathbb{R}^n$, the primal linear program

$$\max_{y\in\mathbb{R}^n} c^\top y \qquad \text{s.t.} \qquad \underline{y} \le y \le \overline{y}$$

has dual problem

$$\min_{\nu\in\mathbb{R}^{2n}} \begin{bmatrix} \overline{y}^\top & -\underline{y}^\top \end{bmatrix} \nu \qquad \text{s.t.} \qquad \begin{bmatrix} I_n & -I_n \end{bmatrix} \nu = c, \quad \nu \ge \mathbf{0},$$

which can also be equivalently written as

$$\min_{\nu\in\mathbb{R}^n} (\overline{y} - \underline{y})^\top \nu + \underline{y}^\top c \qquad \text{s.t.} \qquad \nu \ge \mathbf{0}, \quad \nu - c \ge \mathbf{0}.$$

Since strong duality always holds for linear programs, the optimal values of the primal and dual problems will be equal so long as one of the problems is feasible, e.g., so long as the box $[\underline{y}, \overline{y}]$ is nonempty. We can thus incorporate this dual problem into the outer minimization of (7.1) to yield the non-robust form (7.4).

Maximizing over an ellipsoid

For any $c \in \mathbb{R}^n \setminus \{0\}$, $\Sigma \in \mathbb{S}^n_{++}$, and $q > 0$, the primal quadratically constrained linear program (QCLP)

$$\max_{y \in \mathbb{R}^n} c^\top y \qquad \text{s.t.} \qquad (y - \mu)^\top \Sigma^{-1} (y - \mu) \leq q$$

has dual problem

$$\min_{\nu \in \mathbb{R}} \frac{1}{4\nu} c^\top \Sigma c + \mu^\top c + \nu q \qquad \text{s.t.} \quad \nu \geq 0.$$

By Slater's condition, strong duality holds by virtue of the assumption that $q > 0$ (which implies strict feasibility of the primal problem), and thus the primal and dual problems have the same optimal value. Moreover, since Σ is positive definite and $q > 0$, this problem has a unique optimal solution at $\nu^\star = \frac{1}{2\sqrt{q}} \|L^\top c\|_2$, where L is the unique lower-triangular Cholesky factor of Σ (i.e., $\Sigma = LL^\top$). Substituting $\nu^\star$ into the dual problem yields

$$\sqrt{q} \|L^\top c\|_2 + \mu^\top c.$$

Plugging this into (7.1) yields the non-robust form (7.5).

We write the dual objective in terms of the Cholesky factor L because our predictive models for ellipsoidal uncertainty directly output the entries of L (see Section 7.E). Note, however, that the dual problem solution can be equivalently written in terms of the square-root of Σ, because

$$\|L^\top c\|_2^2 = c^\top L L^\top c = c^\top \Sigma c = c^\top \Sigma^{1/2} \Sigma^{1/2} c = \|\Sigma^{1/2} c\|_2^2.$$

Proof of Theorem 7.4.1: Maximizing over the sublevel set of a PICNN

Let $s_\theta : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ be a partially input-convex neural network (PICNN) with ReLU activations as described in (7.6), so that $s_\theta(x, y)$ is convex in y. Suppose that all the hidden layers have the same dimension d (i.e., $\forall l = 0, \ldots, L-1$: $W_l \in \mathbb{R}^{d \times d}$, $V_l \in \mathbb{R}^{d \times n}$, $b_l \in \mathbb{R}^d$), and the final layer L has $W_L \in \mathbb{R}^{1 \times d}$, $V_L \in \mathbb{R}^{1 \times n}$, $b_L \in \mathbb{R}$. Let $c \in \mathbb{R}^n$ be any vector. Then, the optimization problem

$$\max_{y \in \mathbb{R}^n} c^\top y \qquad \text{s.t.} \qquad s_\theta(x, y) \leq q \tag{7.8}$$

can be equivalently written as

$$
\begin{aligned}
\max_{y\in\mathbb{R}^n,\ \sigma_1,\dots,\sigma_L\in\mathbb{R}^d} \quad & c^\top y && && \text{(7.9a)}\\
\text{s.t.} \quad & \sigma_l \geq \mathbf{0}_d && \forall l = 1,\dots,L && \text{(7.9b)}\\
& \sigma_{l+1} \geq W_l\sigma_l + V_l y + b_l && \forall l = 0,\dots,L-1 && \text{(7.9c)}\\
& W_L\sigma_L + V_L y + b_L \leq q. && && \text{(7.9d)}
\end{aligned}
$$

To see that this is the case, first note that (7.9) is a relaxed form of (7.8), obtained by replacing the equalities $\sigma_{l+1} = \mathrm{ReLU}\left(W_l\sigma_l + V_l y + b_l\right)$ in the definition of the PICNN (7.6) with the two separate inequalities $\sigma_{l+1} \geq \mathbf{0}_d$ and $\sigma_{l+1} \geq W_l\sigma_l + V_l y + b_l$ for each $l = 0,\dots,L-1$. As such, the optimal value of (7.9) is no less than that of (7.8). However, given an optimal solution $y, \sigma_1,\dots,\sigma_L$ to (7.9), it is possible to obtain another feasible solution $y, \hat{\sigma}_1,\dots,\hat{\sigma}_L$ with the same optimal objective value by iteratively decreasing each component of σ_l until one of the two inequality constraints (7.9b), (7.9c) is tight, beginning at $l = 1$ and incrementing l once all entries of σ_l cannot be decreased further. This procedure of decreasing the entries in each σ_l will maintain problem feasibility, since the weight matrices W_l are all assumed to be entrywise nonnegative in the PICNN construction; in particular, this procedure will not increase the left-hand side of (7.9d). Moreover, since one of the two constraints (7.9b), (7.9c) will hold for each entry of each $\hat{\sigma}_l$, this immediately implies that y is feasible for the unrelaxed problem (7.8), and so (7.8) and (7.9) must have the same optimal value.

Having shown that we may replace the convex program (7.8) with a linear equivalent (7.9), we can write this latter problem in the matrix form

$$
\max_{y\in\mathbb{R}^n,\ \sigma_1,\dots,\sigma_L\in\mathbb{R}^d} c^\top y \qquad \text{s.t.} \qquad A\begin{bmatrix} y \\ \sigma_1 \\ \vdots \\ \sigma_L \end{bmatrix} \leq b
$$

where

$$A = \begin{bmatrix} & -I_d & & \\ & & \ddots & \\ & & & -I_d \\ V_0 & -I_d & & \\ \vdots & W_1 & \ddots & \\ \vdots & & \ddots & -I_d \\ V_L & & & W_L \end{bmatrix} \in \mathbb{R}^{(2Ld+1)\times(n+Ld)}, \qquad b = \begin{bmatrix} \mathbf{0}_d \\ \vdots \\ \mathbf{0}_d \\ -b_0 \\ \vdots \\ -b_{L-1} \\ q - b_L \end{bmatrix} \in \mathbb{R}^{2Ld+1}. \tag{7.10}$$

By strong duality, if this linear program has an optimal solution, its optimal value is equal to the optimal value of its dual problem:

$$\min_{\nu\in\mathbb{R}^{2Ld+1}} b^\top \nu \qquad \text{s.t.} \qquad A^\top \nu = \begin{bmatrix} c \\ \mathbf{0}_{Ld} \end{bmatrix}, \quad \nu \geq 0. \tag{7.11}$$

We can incorporate this dual problem (7.11) into the outer minimization of (7.1) to yield the non-robust form (7.7). For a more interpretable form of this dual problem, let $\nu^{(i)}$ denote the portion of the dual vector ν corresponding to the i-th block-row of matrix A, indexed from 0. That is, $\nu^{(i)} = \nu_{id+1:(i+1)d}$ for $i = 0, \dots, 2L-1$. Furthermore, let $\mu = \nu_{2Ld+1}$ be the last entry of ν. Written out, the dual problem (7.11) becomes

$$\begin{aligned} \min_{\nu^{(0)},\dots,\nu^{(2L-1)}\in\mathbb{R}^d,\, \mu\in\mathbb{R}} \quad & \mu(q - b_L) - \sum_{l=0}^{L} b_l^\top \nu^{(L+l)} \\ \text{s.t.} \quad & \begin{bmatrix} V_0^\top & \cdots & V_L^\top \end{bmatrix} \nu_{Ld+1:} = c \\ & W_{l+1}^\top \nu^{(L+l+1)} - \nu^{(L+l)} - \nu^{(l)} = \mathbf{0}_d \qquad \forall l = 0, \dots, L-1 \\ & \nu \geq 0. \end{aligned}$$

Ensuring feasibility of the PICNN maximization problem

As noted at the end of Section 7.3, it may sometimes be the case that the inner maximization problem of (7.1) is unbounded or infeasible when $\Omega_\theta(x)$ is parametrized by a PICNN, since in general, the sublevel sets of the PICNN might be unbounded, or the q selected by the split conformal procedure detailed in Section 7.3 may be sufficiently small that $\Omega_\theta(x) = \{\hat{y} \in \mathbb{R}^n \mid s_\theta(x, \hat{y}) \leq q\}$ is empty for certain inputs x. We can address each of these concerns using separate techniques.

Ensuring compact sublevel sets To ensure that the PICNN-parametrized score function $s_\theta(x, y)$ has compact sublevel sets in y, we can redefine the output layer by setting $V_L = \mathbf{0}_{1\times n}$ and adding a small ℓ^∞ norm term penalizing growth in y:

$$s_\theta(x, y) = W_L \sigma_L + \epsilon \|y\|_\infty + b_L, \tag{7.12}$$

where $\epsilon \geq 0$ is a small penalty term, and where all the remaining parameters and layers remain identical to their definition in (7.6). This modification ensures that, for any fixed x, $s_\theta(x, y)$ has compact sublevel sets, since $\sigma_L \geq 0$ by construction (7.6) and the penalty term $\epsilon\|y\|_\infty$ will grow unboundedly large as y goes to infinity in any direction, so long as $\epsilon > 0$. Moreover, so long as ϵ is chosen sufficiently small and the PICNN is sufficiently deep, this modification should not negatively impact the ability of the PICNN to represent general compact convex uncertainty sets.

Using this modified PICNN (7.12), the maximization problem (7.8) can be written as an equivalent linear program

$$\max_{\substack{y\in\mathbb{R}^n,\ \sigma_1,\ldots,\sigma_L\in\mathbb{R}^d,\\ \kappa\in\mathbb{R}}} \quad c^\top y \tag{7.13a}$$

$$\text{s.t.} \quad \sigma_l \geq \mathbf{0}_d \qquad \forall l = 1, \ldots, L \tag{7.13b}$$

$$\sigma_{l+1} \geq W_l \sigma_l + V_l y + b_l \qquad \forall l = 0, \ldots, L-1 \tag{7.13c}$$

$$\kappa \geq y_i,\ \kappa \geq -y_i \qquad \forall i = 1, \ldots, n \tag{7.13d}$$

$$W_L \sigma_L + \epsilon\kappa + b_L \leq q \tag{7.13e}$$

where the equivalence between (7.8) and (7.13) follows the same argument as that employed in the previous section when showing the equivalence of (7.8) and (7.9). We can thus likewise apply strong duality to obtain an equivalent minimization form of the problem (7.13) and incorporate this into the outer minimization of (7.1) to yield a non-robust problem of the general form (7.7).

Ensuring $\Omega_\theta(x)$ is nonempty If the q chosen by the split conformal procedure is too small such that $\Omega_\theta(x)$ is empty, i.e., $q \leq q_{\min}$ where

$$q_{\min} := \min_{\hat{y}\in\mathbb{R}^n} s_\theta(x, \hat{y}), \tag{7.14}$$

then we simply increase q to $q_{\min}$ so that $\Omega_\theta(x)$ is guaranteed to be nonempty. That is, for input x, we set

$$q = \max\left(\min_{\hat{y}\in\mathbb{R}^n} s_\theta(x, \hat{y}),\ \textsc{Quantile}(\{s_\theta(x_i, y_i)\}_{(x_i,y_i)\in D_{\mathrm{cal}}},\ 1-\alpha)\right).$$

This preserves the marginal coverage guarantee, as increasing q can only result in a larger uncertainty set $\Omega_\theta(x)$.

In theory, $q_{\min}$ varies as a function of θ, and it is possible to differentiate through the optimization problem (7.14) using the methods from Agrawal et al. [290] since the problem is convex and s_θ is assumed to be differentiable w.r.t. θ almost everywhere. However, to avoid this added complexity, in practice, we treat $q_{\min}$ as a constant. In other words, on inputs x where we have to increase q to $q_{\min}$, we treat $\frac{\partial q}{\partial \theta} = 0$.

7.D Exact Differentiable Conformal Prediction

In this section, we prove how to exactly differentiate through the conformal prediction procedure, unlike the approximate derivative first introduced in [279].

Theorem 7.D.1. *Let $\alpha \in (0,1)$ be a risk level, and let $s_i := s_\theta(x_i, y_i)$ denote the scores computed by a score function $s_\theta : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ over data points $\{(x_i, y_i)\}_{i=1}^M$. Suppose $s_\theta(x_i, y_i)$ is differentiable w.r.t. θ for all $i = 1, \dots, M$.*

Define $s_{M+1} := \infty$. Let $\sigma : \{1, \dots, M+1\} \to \{1, \dots, M+1\}$ denote the permutation that sorts the scores in ascending order, such that $s_{\sigma(i)} \leq s_{\sigma(j)}$ for all $i < j$. For simplicity of notation, we may write $s_{(i)} := s_{\sigma(i)}$.

Let $q = \textsc{Quantile}(\{s_i\}_{i=1}^M,\ 1-\alpha)$ where the $\textsc{Quantile}$ function is as defined in Algorithm 12. That is, $q = s_{(k)}$, where $k := \lceil (M+1)(1-\alpha) \rceil \in \{1, \dots, M, M+1\}$. If $s_{(k)}$ is unique, then

$$\frac{\mathrm{d}q}{\mathrm{d}\theta} = \begin{cases} \frac{\mathrm{d}}{\mathrm{d}\theta} s_\theta(x_{\sigma(k)}, y_{\sigma(k)}), & \text{if } \alpha \geq \frac{1}{M+1} \\ 0, & \text{otherwise.} \end{cases}$$

Proof. First, when $\alpha \in (0, \frac{1}{M+1})$, we have $k = M+1$, so $q = \infty$ is constant regardless of the choice of θ. Thus, $\frac{\mathrm{d}q}{\mathrm{d}\theta} = 0$.

Now, suppose $\alpha \geq \frac{1}{M+1}$. The $\textsc{Quantile}$ function returns the k-th largest value of $\{s_i\}_{i=1}^M \cup \{\infty\}$. Since we assume $s_{(k)}$ is unique, we have $\frac{\mathrm{d}q}{\mathrm{d}s_{(i)}} = \mathbf{1}[i = k]$. Finally, we have

$$\frac{\mathrm{d}q}{\mathrm{d}\theta} = \sum_{i=1}^M \frac{\mathrm{d}q}{\mathrm{d}s_i} \frac{\mathrm{d}s_i}{\mathrm{d}\theta} = \sum_{i=1}^M \frac{\mathrm{d}q}{\mathrm{d}s_{(i)}} \frac{\mathrm{d}s_{(i)}}{\mathrm{d}\theta} = \frac{\mathrm{d}s_{(k)}}{\mathrm{d}\theta} = \frac{\mathrm{d}}{\mathrm{d}\theta} s_\theta(x_{\sigma(k)}, y_{\sigma(k)}).$$

□

The two key assumptions in this theorem are that (1) s_θ is differentiable w.r.t. θ, and (2) $s_{(k)}$ is unique. When s_θ is a neural network with a common activation function (e.g., ReLU), (1) holds for inputs $(x, y) \in \mathbb{R}^m \times \mathbb{R}^n$ almost everywhere and θ almost everywhere. Regarding (2), in practice, just as the gradient of the max function is typically implemented without checking whether its inputs have ties, we do not check whether $s_{(k)}$ is unique.

7.E Experiment Details

Our experiments were conducted across a variety of machines, including private servers and Amazon AWS EC2 instances, ranging from 12-core to 128-core machines. Our ETO experiments benefited from GPU acceleration across a combination of NVIDIA GeForce GTX 1080 Ti, Titan RTX, T4, and A100 GPUs. Our E2E experiments did not use GPU acceleration, due to the lack of GPU support in the `cvxpylayers` Python package [290].

In all experiments, we use a batch size of 256 and the Adam optimizer [306]. Models were trained for up to 100 epochs with early stopping if there was no improvement in validation loss for 10 consecutive epochs.

For box and ellipsoid **ETO** baseline models, we performed a hyperparameter grid search over learning rates ($10^{-4.5}$, 10^{-4}, $10^{-3.5}$, 10^{-3}, $10^{-2.5}$, 10^{-2}, $10^{-1.5}$) and L2 weight decay values (0, 10^{-4}, 10^{-3}, 10^{-2}). For PICNN **ETO** models we performed a hyperparameter grid search over learning rates (10^{-4}, 10^{-3}, 10^{-2}) and L2 weight decay values (10^{-4}, 10^{-3}, 10^{-2}).

Uncertainty representation

Box uncertainty Our box uncertainty model uses a neural network h_θ with 3 hidden layers of 256 units each and ReLU activations with batch-normalization. The output layer has dimension $2n$, where dimensions $1 : n$ predict the lower bound. Output dimensions $n + 1 : 2n$, after passing through a softplus to ensure positivity, represents the difference between the upper and lower bounds. That is,

$$\begin{bmatrix} h_\theta^{\mathrm{lo}}(x) \\ h_\theta^{\mathrm{hi}}(x) \end{bmatrix} = \begin{bmatrix} h_\theta(x)_{1:n} \\ h_\theta(x)_{1:n} + \operatorname{softplus}(h_\theta(x)_{n+1:2n}) \end{bmatrix}.$$

This architecture ensures that $h_\theta^{\mathrm{hi}}(x) > h_\theta^{\mathrm{lo}}(x)$.

In the two-stage **ETO** baseline, we first train h_θ to estimate the $\alpha/2$- and $(1-\alpha/2)$-quantiles, so that $[h_\theta^{\mathrm{lo}}(x), h_\theta^{\mathrm{hi}}(x)]$ represents the centered $(1-\alpha)$-confidence region. Quantile regression is a common method for generating uncertainty sets for scalar predictions by estimating quantiles of the conditional distribution $\mathcal{P}(y \mid x)$ [288].

For scalar true label y, quantile regression models are commonly trained to minimize *pinball loss* (a.k.a. *quantile loss*) where β is the quantile level being estimated:

$$\text{pinball}_\beta(\hat{y}, y) = \begin{cases} \beta \cdot (y - \hat{y}), & \text{if } y > \hat{y} \\ (1-\beta) \cdot (\hat{y} - y), & \text{if } y \leq \hat{y}. \end{cases}$$

To generalize the pinball loss to our setting of multi-dimensional $y \in \mathbb{R}^n$, we sum the pinball loss across the dimensions of y: $\text{pinball}_\beta(\hat{y}, y) = \sum_{i=1}^n \text{pinball}_\beta(\hat{y}_i, y_i)$.

Our end-to-end (E2E) box uncertainty models use the same architecture as above, initialized with weights from the the trained **ETO** model. We found it helpful to use a weighted combination of the task loss and pinball loss during training of the E2E models to improve training stability. In our experiments, we used a weight of 0.9 on the task loss and 0.1 on the pinball loss. The E2E models used the best L2 weight decay from the **ETO** models, and the learning rate was tuned across 10^{-2}, 10^{-3}, and 10^{-4}.

Ellipsoidal uncertainty Our ellipsoidal uncertainty model uses a neural network h_θ with 3 hidden layers of 256 units each and ReLU activations with batch-normalization. The output layer has dimension $n + n(n+1)/2$, where dimensions $1 : n$ predict the mean $\mu_\theta(x)$ and the remaining output dimensions are used to construct a lower-triangular Cholesky factor $L_\theta(x)$ of the covariance matrix $\Sigma_\theta(x) = L_\theta(x) L_\theta(x)^\top$. We pass the diagonal entries of $L_\theta(x)$ through a softplus function to ensure strict positivity, which then ensures $\Sigma_\theta(x)$ is positive definite.

For the **ETO** baseline, we trained the model using the negative log-likelihood (NLL) loss

$$\text{NLL}(\theta) = \frac{1}{N} \sum_{(x,y) \in D} -\ln \mathcal{N}(y \mid \mu_\theta(x), \Sigma_\theta(x)),$$

where $\mathcal{N}(\cdot \mid \mu, \Sigma)$ denotes the density of a multivariate normal distribution with mean μ and covariance matrix Σ.

Our end-to-end (E2E) ellipsoidal uncertainty models use the same architecture as above, initialized with weights from the the trained **ETO** model. We found it helpful to use a weighted combination of the task loss and NLL loss during training of the E2E models to improve training stability. In our experiments, we used a weight of 0.9 on the task loss and 0.1 on the NLL loss. The E2E models used the best L2 weight decay from the **ETO** models, and the learning rate was tuned across 10^{-2}, 10^{-3}, and 10^{-4}.

PICNN uncertainty Our PICNN has 2 hidden layers with ReLU activations.

For the battery storage problem, we used 64 units per hidden layer. We did not run into any feasibility issues for the PICNN maximization problem, so we did not restrict V_L as described in Section 7.C, and we set $\epsilon = 0$.

For the portfolio optimization problem, we tried 32, 64, and 128 units per hidden layer, finding that 32 units worked best for the portfolio optimization problem, 128 units performed marginally better for the battery storage problem. We did run into feasibility issues for the PICNN maximization problem, which we resolved by setting $V_L = \mathbf{0}_{1\times n}$ as described in Section 7.C. This change alone was sufficient, and we set $\epsilon = 0$.

For the **ETO** baseline, we take inspiration from the approach by [292] to give probabilistic interpretation to a PICNN model s_θ via the energy-based model $\hat{\mathcal{P}}_\theta(y \mid x) = \frac{1}{Z_\theta(x)} \exp(-s_\theta(x, y))$ where $Z_\theta(x) := \int_{\tilde{y}\in\mathbb{R}^n} \exp(-s_\theta(x, \tilde{y}))\, \mathrm{d}\tilde{y}$ is the normalizing constant. We train our **ETO** PICNN models with an approximation to the true NLL loss based on samples from the Metropolis-Adjusted Langevin Algorithm (MALA), a Markov Chain Monte Carlo (MCMC) method. We refer readers to our code for the specific hyperparameters and implementation details we used.

Note that under this energy-based model, adding a scalar constant c to the PICNN (i.e., $s_\theta(x, y) + c$) does not change the probability distribution. That is, $\exp(-s_\theta(x, y)) \propto \exp(-s_\theta(x, y) + c)$. To regularize the PICNN model, which has a bias term in its output layer, we therefore introduce a regularization loss of $w_{\text{zero}} \cdot s_\theta(x, y)^2$ where w_{zero} is a regularization weight. This regularization loss encourages $s_\theta(x, y)$ to be close to 0, for all examples in the training set. In our experiments, we set $w_{\text{zero}} = 1$.

Our end-to-end (E2E) PICNN uncertainty models use the same architecture as above, initialized with weights from the the trained **ETO** model. Unlike for box and ellipsoidal uncertainty which used a weighted combination of task loss and NLL loss, our E2E PICNN uncertainty models are trained only with the task loss. Similar to the **ETO** PICNN model, we also regularize the E2E PICNN. Here, we add a regularization loss of $w_{\text{q}} \cdot q^2$, where w_{q} is a regularization weight and q is the conformal prediction threshold computed in each minibatch of E2E training. This regularization loss term aims to keep q near 0; without this regularization, we found that q tended to grow dramatically over training epochs with poor task loss. In our experiments, we set $w_{\text{q}} = 0.01$.

The E2E models used the best L2 weight decay from the `ETO` models. For the battery storage problem, we tested learning rates of 10^{-3} and 10^{-4}. For the portfolio optimization problem, we used a learning rate of 5×10^{-3}.

Data

Price forecasting for battery storage We use the same dataset as Donti, Amos, and Kolter [291] in our price forecasting for battery storage problem. In this dataset, the target $y \in \mathbb{R}^{24}$ is the hourly PJM day-ahead system energy price for 2011-2016, for a total of 2189 days. Unlike Donti, Amos, and Kolter [291], though, we do not exclude any days whose electricity prices are too high (>500\$/MWh). Whereas Donti, Amos, and Kolter [291] treated these days as outliers, our conditional robust optimization problem is designed to output robust decisions. For predicting target for a given day, the inputs $x \in \mathbb{R}^{101}$ include the previous day's log-prices, the given day's hourly load forecast, the previous day's hourly temperature, the given day's hourly temperature, and several calendar-based features such as whether the given day is a weekend or a US holiday.

For the setting without distribution shift, we take a random 20% subset of the dataset as the test set; because the test set is selected randomly, it is considered exchangeable with the rest of the dataset. For the setting with distribution shift, we take the chronologically last 20% of the dataset as the test set; because load, electricity prices, and temperature all have distribution shifts over time, the test set is not exchangeable with the rest of the dataset. For each seed, we further use a 80/20 random split of the remaining data for training and calibration.

Portfolio optimization For the portfolio optimization task, we used synthetically generated data. We sample $x \in \mathbb{R}^2$, $y \in \mathbb{R}^2$ from a mixture of three 4-D multivariate Gaussian distributions as used in [272]. Formally,

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim p_a \mathcal{N}(\mu_a, \Sigma_a) + p_b \mathcal{N}(\mu_b, \Sigma_b) + p_c \mathcal{N}(\mu_c, \Sigma_c)$$

where $p_a + p_b + p_c = 1$. Specifically,

$$
\begin{aligned}
p_a &= \phi, & p_b &= \frac{1}{\alpha_{\text{GMM}} + 1}(1 - \phi), & p_c &= \frac{\alpha_{\text{GMM}}}{\alpha_{\text{GMM}} + 1}(1 - \phi), \\
\mu_a &= \mathbf{0}_4, & \mu_b &= \begin{bmatrix} 0 & 5 & 5 & 0 \end{bmatrix}^\top, & \mu_c &= \mu_b, \\
\Sigma_a &= \begin{bmatrix} 1 & 0 & 0.37 & 0 \\ 0 & 1.5 & 0 & 0 \\ 0.37 & 0 & 2 & 0.73 \\ 0 & 0 & 0.73 & 3 \end{bmatrix}, & \Sigma_b &= \alpha_{\text{GMM}} \Sigma_a, & \Sigma_c &= \frac{1}{\alpha_{\text{GMM}}} \Sigma_a
\end{aligned}
$$

for some $\phi \in [0, 1]$ and $\alpha_{\text{GMM}} \in [0, 1]$. In our experiments, we used $\phi = 0.7$ and $\alpha_{\text{GMM}} = 0.9$. (Chenreddy and Delage [272] do not disclose the values of ϕ and α_{GMM} chosen for their experiments.) For each random seed, we generate 2000 samples and use a (train, calibration, test) split of (600, 400, 1000).

Chapter 8

PRICING UNCERTAINTY IN STOCHASTIC MULTI-STAGE ELECTRICITY MARKETS

In the previous chapter, we considered the question of how to learn calibrated uncertainty sets to improve performance in a class of conditional robust optimization problems. However, if we seek to deploy such robust strategies in applications such as electricity market dispatch, a complementary, operational question arises: how should we *price* this uncertainty to ensure efficient market operation? Inspired by this question, this chapter proposes a pricing mechanism for multi-stage electricity markets that does not explicitly depend on the choice of dispatch procedure or optimization method. Our approach is applicable to a wide range of methodologies for the economic dispatch of power systems under uncertainty, including multi-interval dispatch, multi-settlement markets, scenario-based dispatch, and chance-constrained or robust dispatch policies. We prove that our pricing scheme provides both *ex-ante* and *ex-post* dispatch-following incentives by simultaneously supporting per-stage and ex-post competitive equilibria. In numerical experiments on a ramp-constrained test system, we demonstrate the benefits of scheduling under uncertainty and show how our price decomposes into components corresponding to energy, intertemporal coupling, and uncertainty.

This chapter is primarily based on the following paper (© 2023 IEEE):

[1] L. Werner*, N. Christianson*, A. Zocca, A. Wierman, and S. Low, "Pricing Uncertainty in Stochastic Multi-Stage Electricity Markets," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, Dec. 2023, pp. 1580–1587. DOI: 10.1109/CDC49753.2023.10384022. [Online]. Available: https://ieeexplore.ieee.org/document/10384022.

8.1 Introduction

Rapid changes in the composition of the generation mix in power markets is creating several challenges for system operators (SOs). First, increasing renewable penetration from solar and wind is injecting variability and uncertainty into available power supply. Second, there is a lack of suitable market mechanisms tailored to the physical characteristics of distributed energy resources (DERs), such as energy storage, which are seeking to join markets in increasing numbers. Third, electrification of vehicle charging and thermal (heating/cooling) loads is impacting the shape and variability of the demand profile, leading to periods of high, sustained ramping.

These factors have a common theme of uncertainty, and SOs have been rapidly innovating on new market structures and dispatch procedures to handle it. These include multi-interval lookahead dispatch [307], ramping reserves [308], operating reserves [309], capacity markets [310], and multi-stage or intraday markets. Alongside, researchers have been investigating techniques from stochastic optimization to efficiently dispatch the market under uncertainty, including robust optimization [41, 311], chance-constrained optimization [48], scenario optimization [312], and distributionally robust optimization [46].

Uncertainty impacts the stability of pricing signals and can lead to market distortions such as out-of-merit dispatch, ramping shortages, and load shedding. Even with more advanced and accurate forecasts, SOs must still dispatch the system in a way that anticipates forecast uncertainty and the possibility of distribution shift over time. Pricing that incorporates characterizations of uncertainty is necessary to fairly and efficiently compensate different resources for their contributions to a reliable power supply.

The contribution in this chapter is a pricing scheme for multi-stage markets that does not depend on the particular characterization of uncertainty or the method for optimizing over dispatch decisions that account for this uncertainty. Our approach is different from those in several recent works where the construction of the energy price intimately depends on the optimization paradigm (e.g., chance-constrained [46], robust [311], or rolling-window [313, 314]). We show that our proposed prices can be decomposed into components corresponding to the standard locational marginal price (LMP), intertemporal coupling, and uncertainty. Finally, we establish that this price clears the market under profit-maximizing assumptions on the participants and that it supports both *ex-ante* and *ex-post* dispatch-following incentives (see Section 8.3 for definitions).

Related Work

Our work draws on two main lines of inquiry into electricity market mechanism design. The first is dispatching and pricing multi-interval markets in the presence of intertemporal coupling constraints. The second is dispatching and pricing using techniques from robust and stochastic optimization.

Pricing multi-period electricity markets In rolling-window real-time economic dispatch schemes, distribution shift in predicted net demand can lead to lost opportunity cost (LOC) and distorted truthful bidding incentives for generators. Several pricing mechanisms building on standard uniform pricing schemes have been proposed in recent years to mitigate the lack of dispatch-following incentives [307, 315, 316]. A more recent line of work [31, 313, 314] has proposed a non-uniform pricing scheme, Temporal Locational Marginal Pricing (TLMP), and has established a dual definition of dispatch-following incentives. Simultaneously satisfying a "partial equilibrium" (i.e., *ex ante* dispatch-following incentive in every stage) and a general equilibrium (i.e., *ex post*) forms the notion of "strong equilibrium," used in this work.

Our pricing mechanism is distinguished from these works as they do not incorporate uncertainty directly in the lookahead dispatch algorithms, but rather design prices to mitigate incentive misalignment as a result of inaccurate predictions and distribution shift. However, these lookahead algorithms might be infeasible [29, 43], necessitating our development of more general pricing schemes that can incorporate such robust constraints.

Pricing stochastic electricity markets There has been much recent interest in designing electricity markets incorporating robust or stochastic constraints to ensure reliable operation in the face of uncertainty. For example, such dispatch schemes include economic dispatch with robust constraints [41, 311], chance constraints [47, 317, 318], distributionally robust chance constraints [45, 46], and conditional value at risk constraints [48]. However, in the subset of these works that explicitly address the problem of designing price mechanisms for the stochastic dispatch problem, inconsistent notions of *ex ante* dispatch-following incentives are considered which leaves open the need for out-of-market settlements to make up for lost opportunity cost.

This work improves practically upon existing methodologies by combining the temporally-coupled multi-interval dispatch used in practice with stochastic market-clearing mechanisms proposed in the research literature. Our approach can be applied to any formulation of stochastic or robust economic dispatch and ensures zero lost opportunity cost on the part of market participants by considering both *ex ante* and *ex post* dispatch-following incentives in the price specification.

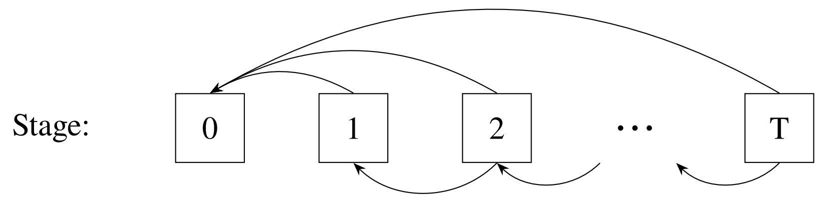


Figure 8.1: Coupling between $T + 1$ stages in DA + RT economic dispatch. A directed edge between two stages indicates that the later-stage decision depends explicitly on the decision committed to in the earlier stage.

8.2 Multi-Stage Dispatch Under Uncertainty

The day-ahead (DA) and real-time (RT) stages of electricity market clearing form a $T + 1$ stage sequential optimization problem, with coupling between the stages and uncertainty from load and renewables realized between each of the T stages. The first stage is the single-shot, DA optimization problem which determines a unit commitment and associated dispatch for the upcoming 24-hour time horizon. This dispatch, although not physically realized, may be financially settled. Subsequently, in real time, a receding-horizon multi-interval optimization is performed. The first interval from each of these T subproblems is financially binding. Between each of the subproblems, the SO utilizes updated forecasts of uncertain demand and renewable generation to improve the efficiency of the dispatch.

The stages of the sequential problem are temporally coupled in the manner depicted in Figure 8.1. The first (DA) stage couples to all of the subsequent stages because it fixes the unit commitment—and therefore the upper/lower generation bounds, ramp limits, etc.—in the T subsequent (RT) stages. Within the RT market, stages are coupled consecutively due to the form of ramping constraints and the battery state-of-charge updates.

Since the $T + 1$ stages are solved and settled sequentially, we consider two groups of stages at a time: the period with no uncertainty, and the set of periods with remaining uncertainty. In the DA stage, the SO seeks to solve a stochastic optimization problem that fixes here-and-now decisions for the unit commitment while selecting policies for the wait-and-see decisions of RT stage 1. The purpose of the policies is to provide realization-dependent recourse in subsequent stages. However, in each of these stages, after uncertainty has been revealed, the multi-interval optimization is solved again for the next stage.

Notation

For each optimization interval indexed by $t \in \{0, \ldots, T\}$, each market participant $i \in \{1, \ldots, N\}$ has a dispatch vector $\mathbf{x}_{i,t} \in \mathbb{R}^{M_{i,t}}$ where $M_{i,t}$ is the dimension of the dispatch vector for i in stage t. The dispatch $\mathbf{x}_{i,t}$ includes all of the quantities associated with participant i in stage t. For conventional generators, this is just their power generation. For storage resources, it includes power generation and state-of-charge. We do not consider discrete variables, such as those needed for unit commitment, in our presentation here. They can be included without impacting our pricing or dispatch results, although the dispatch problem would need to be modified slightly as in [46, 319]. System states, such as nodal power injections, line flows, and voltage angles, can be written in terms of the individual dispatch variables $\mathbf{x}_{i,t}$ and are therefore not explicitly notated. For each t, we collect dispatch vectors into a single decision vector:

$$\mathbf{x}_t := (\mathbf{x}_{1,t}, \ldots, \mathbf{x}_{N,t}) \in \mathbb{R}^{M_t},$$

where $M_t := \sum_i M_{i,t}$. Associated with each dispatch vector is a market price $\boldsymbol{\pi}_t \in \mathbb{R}^{M_t}$. The revenue (or payment) each participant receives over the entire horizon is $\boldsymbol{\pi}_t^\top \mathbf{x}_t$.

For each t we associate a random vector of uncertainty $\boldsymbol{\xi}_t \in \mathbb{R}^{P_t}$. Realizations of $\boldsymbol{\xi}_t$, denoted $\hat{\boldsymbol{\xi}}_t$, are obtained sequentially after the dispatch $\hat{\mathbf{x}}_{t-1}$ has been committed but prior to computing $\mathbf{x}_t$. We also assume that the SO has access to a forecast θ_t that represents their best knowledge at stage t about subsequent uncertainty $\boldsymbol{\xi}_{t+1}, \ldots, \boldsymbol{\xi}_T$. The composition of the forecast depends on what information is accessible. In the simplest case, θ_t is just a point forecast of $\boldsymbol{\xi}_{t+1}, \ldots, \boldsymbol{\xi}_T$. When distributional information is available, θ_t can be a set of parameters describing each forecast distribution and its support. Since stage 0 is the DA/UC stage of the market clearing, which happens when no uncertainty has been realized, $\hat{\boldsymbol{\xi}}_0$ is defined to be a set of forecasts over the subsequent T RT intervals.

In the rest of the chapter, we denote by $\mathbf{a}_{\tau:t}$ the set of vectors $\{\mathbf{a}_j\}_{j=\tau}^t$. If $\tau > t$, we define this to be the empty set. For $\tau, t \in \mathbb{N}$ satisfying $\tau \leq t$, we define $[\tau, t] := \{\tau, \tau + 1, \ldots, t\}$.

Ex-post Dispatch Problem and Prices

If the SO had perfect forecasts of uncertainty, it could solve the following optimization problem (8.1) for all time intervals simultaneously. This is a useful solution because it benchmarks the efficiency of dispatch algorithms and quantifies the impact of uncertainty.

Problem 8.1. *Given an uncertainty realization $\hat{\boldsymbol{\xi}}$, the ex-post dispatch problem for all $T + 1$ stages is:*

$$\min_{\mathbf{x}_0,\ldots,\mathbf{x}_T} \quad \sum_{t=0}^{T}\sum_{i=1}^{N} c_{i,t}(\mathbf{x}_{i,0:t};\hat{\boldsymbol{\xi}}_{0:t}) \tag{8.1a}$$

$$\text{s.t.} \quad f_t(\mathbf{x}_t;\hat{\boldsymbol{\xi}}_{0:t}) \le \mathbf{0} \qquad \forall t \tag{8.1b}$$

$$g_{i,t}(\mathbf{x}_{i,t};\hat{\boldsymbol{\xi}}_{0:t}) \le \mathbf{0} \qquad \forall i,\ \forall t \tag{8.1c}$$

$$h_{i,t}(\mathbf{x}_{i,0:t};\hat{\boldsymbol{\xi}}_{0:t}) \le \mathbf{0} \qquad \forall i,\ \forall t. \tag{8.1d}$$

Our formulation contains three types of constraints: (8.1b) convex system-wide constraints f_t that couple decisions across market participants but *within* each stage (e.g., power balance, line flow limits, zonal constraints, reserve requirements); (8.1c) private constraints $g_{i,t}$ for participant i and stage t (e.g., generation limits, state-of-charge (SOC) limits); and (8.1d) private constraints $h_{i,t}$ for participant i coupling their decisions in stage t to all previous dispatches (e.g., ramping, storage SOC updates, unit commitment-dependent generation limits).

This formulation of economic dispatch incorporates linear power flow equations, network constraints, zonal constraints, reserve constraints, private constraints, and intertemporal constraints for both conventional generators, flexible and inflexible loads, and storage.

Assumption 8.1. *We assume that for each i, t, functions $c_{i,t}$, f_t, $g_{i,t}$, and $h_{i,t}$ are convex w.r.t $\mathbf{x}_t$. We also assume that they are causal, in the sense that they possibly depend on any dispatches and uncertainty realized until time t. Finally, for non-triviality, we assume that Problem 8.1 has a feasible solution.*

If market dispatches $\mathbf{x}_0^*, \ldots, \mathbf{x}_T^*$ are generated by the optimal solution of Problem 8.1, then the market clearing price that supports a competitive equilibrium is just the dual multiplier associated with constraint (8.1b) (cf. [313, 316]).

Sequential Market Dispatch

In practice, solving Problem 8.1 is not a viable procedure for clearing the market due to the combination of uncertain inter-stage coupling constraints. Instead, SOs resort to solving a sequence of market-clearing optimization problems.[1] For each stage, updated forecasts of uncertainty are used as problem parameters, and advisory forward decisions are computed, but only the decision for the current stage is settled.

The market-clearing problem for stage t is presented in Problem 8.2, where the function $V_t : \mathbb{R}^{M_t} \rightarrow \mathbb{R}$ represents the forward cost of dispatch $\mathbf{x}_t$; we refer to this as the *forward value* or *cost-to-go* function. As with the functions in Problem 8.1, V_t may be parameterized by all uncertainty realized up to t, all previous dispatches, as well as forecasts of future uncertainty θ_t that are available at time t:

$$V_t(\mathbf{x}_t; \hat{\mathbf{x}}_{i,0:t-1}, \hat{\boldsymbol{\xi}}_{0:t}, \theta_t).$$

In service of simpler notation, we make this dependence on parameters implicit in the remainder of the manuscript and simply refer to $V_t(\mathbf{x}_t)$, except where an explicit reference to a particular parameter is necessary. In Section 8.2, we remark on how V_t is already incorporated in market dispatch problems in practice as well as on the theoretical benefits of abstracting the forward cost of decisions in this way.

Problem 8.2. *Let $\hat{\mathbf{x}}_{0:t-1}$ be the sequence of dispatches committed prior to stage t and $\hat{\boldsymbol{\xi}}_{0:t}$ the uncertainty realized through stage t. The sequential dispatch problem for interval t is:*

$$\min_{\mathbf{x}_t} \sum_i c_{i,t}(\mathbf{x}_{i,t}, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t}) + V_t(\mathbf{x}_t) \tag{8.2a}$$

$$\text{s.t.} \quad \boldsymbol{\lambda}_t \perp f_t(\mathbf{x}_t; \hat{\boldsymbol{\xi}}_{0:t}) \leq \mathbf{0} \tag{8.2b}$$

$$\boldsymbol{\mu}_{i,t} \perp g_{i,t}(\mathbf{x}_{i,t}; \hat{\boldsymbol{\xi}}_{0:t}) \leq \mathbf{0} \qquad \forall i \tag{8.2c}$$

$$\boldsymbol{\eta}_{i,t} \perp h_{i,t}(\mathbf{x}_{i,t}, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t}) \leq \mathbf{0} \qquad \forall i. \tag{8.2d}$$

The dual multipliers associated with each set of constraints are indicated to the left of each constraint (and followed by "$\perp$"). When $V_t(\mathbf{x}_t)$ is convex with respect to $\mathbf{x}_t$, and the convexity conditions from Assumption 8.1 hold, then (8.2) is a convex optimization problem.

The following algorithm specifies how the system operator clears and settles the market over the multi-stage scheduling horizon. Note that at each stage, the SO requires a scheme for deciding the prices $\boldsymbol{\pi}_t^*$ (see below).

[1] For example, this sequence could be the combination of a day-ahead forward market followed by real-time adjustment market clearings every 15 minutes.

Algorithm 8.1.

1. *The SO generates a DA uncertainty forecast* $\hat{\boldsymbol{\xi}}_0$ *and solves Problem 8.2 for* $t = 0$ *to produce decisions* $\mathbf{x}_0^*$ *and prices* $\boldsymbol{\pi}_0^*$.
2. *For* $t = 1, \ldots, T$*:*
   a) *Nature realizes uncertainty* $\hat{\boldsymbol{\xi}}_t$*;*
   b) *The SO solves Problem 8.2 to produce dispatches* $\mathbf{x}_t^*$ *and prices* $\boldsymbol{\pi}_t^*$*;*
   c) *Each participant realizes dispatch* $\hat{\mathbf{x}}_t \coloneqq \mathbf{x}_t^*$ *and settles with the SO i at* ${\boldsymbol{\pi}_{i,t}^*}^\top \hat{\mathbf{x}}_{i,t}$.

Assumption 8.2. *Solving Problem 8.2 iteratively for* $t = 0, \ldots, T$ *produces a feasible sequence of dispatches. Note that such recursive feasibility is in general not guaranteed and may depend on the choice of* V_t *and* θ*; see [29, 43] for further consideration of these details.*

Specifying the cost-to-go function V_t

Depending on the parameterization of the uncertainty forecast θ_t and the choice of the stochastic optimization model, the function V_t adopts different forms. We show below how several common stochastic paradigms fit into this framework. These encompass the multi-settlement and rolling-window optimization procedures (with and without lookahead) used by SOs in practice as well as stochastic optimization formulations increasingly studied in the research literature.

Rolling dispatch without lookahead. This procedure is the traditional approach to dispatching the DA and RT markets, where each stage (or interval) is optimized without considering the forward consequences of the current dispatch. Thus, $\mathbf{x}_t$ is only coupled intertemporally to $\hat{\mathbf{x}}_{0:t-1}$ through the constraints (8.2d). In this case, $V_t \coloneqq 0$ for all $t = 0, \ldots, T$. As this is convex, Problem 8.2 is, therefore, convex and tractable.

Rolling dispatch with lookahead. To better handle uncertainty, SOs make use of forecasts and advisory decisions over a lookahead horizon of length $h > 1$. Exploiting lookahead predictions can increase the feasibility and ex-post optimality of the overall dispatch sequence since it allows for anticipating future ramp, unit commitment, and storage charge/discharge needs [320]. The forecast is a point forecast $\theta_t = (\tilde{\boldsymbol{\xi}}_{t+1}, \ldots, \tilde{\boldsymbol{\xi}}_{t+h})$, available at time t, of the true uncertainties $\hat{\boldsymbol{\xi}}_{t+1}, \ldots, \hat{\boldsymbol{\xi}}_{t+h}$ to be realized.

$$
\begin{aligned}
V_t(\mathbf{x}_t; \theta_t) := \min_{\mathbf{x}_{t+1},\dots,\mathbf{x}_{t+h}} \quad & \sum_{\tau=t+1}^{t+h} \sum_{i=1}^{N} c_{i,\tau}(\mathbf{x}_{i,t+1:\tau}, \hat{\mathbf{x}}_{i,0:t}; \hat{\boldsymbol{\xi}}_{0:t}, \tilde{\boldsymbol{\xi}}_{t+1:\tau}) && \text{(8.3a)} \\
\text{s.t.} \quad & f_\tau(\mathbf{x}_\tau; \hat{\boldsymbol{\xi}}_{0:t}, \tilde{\boldsymbol{\xi}}_{t+1:\tau}) \le \mathbf{0} \qquad \forall \tau && \text{(8.3b)} \\
& g_{i,\tau}(\mathbf{x}_{i,\tau}; \hat{\boldsymbol{\xi}}_{0:t}, \tilde{\boldsymbol{\xi}}_{t+1:\tau}) \le \mathbf{0} \qquad \forall i, \forall \tau && \text{(8.3c)} \\
& h_{i,\tau}(\mathbf{x}_{i,t+1:\tau}, \hat{\mathbf{x}}_{i,0:t}; \hat{\boldsymbol{\xi}}_{0:t}, \tilde{\boldsymbol{\xi}}_{t+1:\tau}) \le \mathbf{0} \qquad \forall i, \forall \tau. && \text{(8.3d)}
\end{aligned}
$$

In the above, $\forall \tau$ means $\tau \in [t+1, t+h]$. By convention, if (8.3) is infeasible, $V_t = +\infty$. By standard arguments on the convexity of the optimal value of a convex program under affine perturbations (e.g., [214, Section 5.6.1]), we have the following structural result on the cost-to-go V_t.

Proposition 8.2.1. *$V_t(\mathbf{x}_t; \theta_t)$ in (8.3) is convex in $\mathbf{x}_t$.*

Although V_t in (8.3) is convex, it is not possible to write down a closed-form solution in general. However, (8.3) can be incorporated into the formulation of the problem (8.2), recovering the standard lookahead economic dispatch problem studied in [313, 316], which is a tractable convex optimization problem. Note that in a solution $\mathbf{x}_t, \mathbf{x}_{t+1}, \dots, \mathbf{x}_{t+h}$ to (8.2) with V_t defined as (8.3), only the first dispatch $\mathbf{x}_t$ is binding for the purposes of Algorithm 8.1. The remaining dispatches $\mathbf{x}_{t+1}, \dots, \mathbf{x}_{t+h}$ are advisory and are re-computed for each successive interval.

Chance-constrained optimization. Chance-constrained optimization has received increasing interest for its ability to optimize over decisions with constraints involving stochastic uncertainty [46, 317, 321]. The form of V_t presented next enables probabilistic guarantees on the feasibility of the advisory dispatch under a distributional assumption on uncertainty. At time t, we define p_t to be the distribution of future uncertainty $\boldsymbol{\xi}_{t+1:t+h}$ conditioned on all uncertainty realizations through time t. The forecast θ_t collects parameters of this distribution or of the SO's best guess of this distribution. In this case, the risk-neutral chance-constrained lookahead value function is defined as follows:

$$
\begin{aligned}
V(\mathbf{x}_t; \theta_t) &:= \\
\min_{\mathbf{x}_{t+1},\dots,\mathbf{x}_{t+h}} \quad & \mathbb{E}_{\boldsymbol{\xi}_{t+1:t+h} \sim p_t} \left[\sum_{\tau=t+1}^{t+h} \sum_{i=1}^{N} c_{i,\tau}(\mathbf{x}_{i,\tau}; \boldsymbol{\xi}_\tau) \right] && \text{(8.4a)} \\
\text{s.t.} \quad & \mathbb{P}_{\boldsymbol{\xi}_{t+1:\tau}}[f_\tau(\mathbf{x}_\tau; \hat{\boldsymbol{\xi}}_{0:t}, \boldsymbol{\xi}_{t+1:\tau}) \le \mathbf{0}] \ge 1 - \varepsilon_\tau^f \quad \forall \tau && \text{(8.4b)} \\
& \mathbb{P}_{\boldsymbol{\xi}_{t+1:\tau}}[g_{i,\tau}(\mathbf{x}_{i,\tau}; \hat{\boldsymbol{\xi}}_{0:t}, \boldsymbol{\xi}_{t+1:\tau}) \le \mathbf{0}] \ge 1 - \varepsilon_{i,\tau}^g \quad \forall i, \forall \tau && \text{(8.4c)} \\
& \mathbb{P}_{\boldsymbol{\xi}_{t+1:\tau}}[h_{i,\tau}(\mathbf{x}_{i,t+1:\tau}, \hat{\mathbf{x}}_{i,0:t}; \hat{\boldsymbol{\xi}}_{0:t}, \boldsymbol{\xi}_{t+1:\tau}) \le \mathbf{0}] \ge 1 - \varepsilon_{i,\tau}^h \quad \forall i, \forall \tau. && \text{(8.4d)}
\end{aligned}
$$

In the above, $\forall \tau$ means $\tau \in [t+1, t+h]$. By convention, if (8.3) is infeasible, $V_t = +\infty$. The hyperparameter ε's can be tuned by the SO to adjust the permissible probability of a constraint violation.

In general, (8.4) is intractable due to the difficulty in computing probabilities and expectations over arbitrary distributions p_t. In particular, the feasible set defined by the constraints may be nonconvex even if the constraint functions $f_\tau, g_{i,\tau}, h_{i,\tau}$ are convex. The structure of the constraints may also make the problem infeasible, e.g., a fixed advisory decision will generally be insufficient to guarantee feasibility under any demand realization, and uncertainty-dependent recourse may be necessary. However, by introducing suitable assumptions on the structure of the problem such as linearity of $c_{i,\tau}, f_\tau, g_{i,\tau}, h_{i,\tau}$, Gaussianity of p_t, separating joint chance constraints into individual chance constraints, and replacing advisory decisions with advisory uncertainty-dependent affine policies, a tractable, convex counterpart to (8.4) can be formed. For details on such a transformation, we refer the reader to recent literature on chance-constrained optimization and economic dispatch [317, 322]. Alternatively, given samples from the underlying distribution of uncertainty, a robust form of this chance-constrained problem could be formulated using an approach similar to the conformal uncertainty set methodology proposed in Chapter 7.

Other stochastic formulations. The procedure we have been following in this section to formulate the sequential dispatch problem in the form (8.2) be applied to other stochastic optimization settings, including scenario-based optimization, robust optimization, and distributionally robust optimization, where there is an extensive literature on convex, tractable reformulations [29, 41, 45, 46, 312].

In fact, although all of these approaches to defining V_t rely on constructing a tractable optimization problem, this is not necessary for Problem 8.2. As long as V_t is convex and it is possible to obtain gradients of V_t for any input $\mathbf{x}_t$, then optimization (8.2) can be solved using gradient-based methods. And, as we will show in the next section, the price formation also depends only on being able to compute gradients of V_t for the market dispatch.

8.3 Pricing Multi-Stage Uncertainty

In this section, we define the market clearing price and prove that it supports a competitive market clearing solution under *ex-ante* and *ex-post* definitions of dispatch-following incentives.

Model of market participation

In order to establish the properties of a competitive equilibrium, we first present the participant's model of market behavior. We assume that the agents are price-takers, in that they do not bid strategically to impact the price. Further, we assume that they optimize for the current stage of the optimization problem and do not price future decisions into the bid for the current interval. We express the agent's profit-maximizing behavior precisely through the following problem.

Problem 8.3. *Under a given price $\boldsymbol{\pi}_{i,t}$, agent i's profit-maximizing schedule in interval t is:*

$$\begin{aligned}
\arg\max_{\overline{\mathbf{x}}_{i,t}} \quad & \boldsymbol{\pi}_{i,t}^\top \overline{\mathbf{x}}_{i,t} - c_{i,t}(\overline{\mathbf{x}}_{i,t}, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t}) && (8.5a)\\
\text{s.t.} \quad & \overline{\boldsymbol{\mu}}_{i,t} \perp g_{i,t}(\overline{\mathbf{x}}_{i,t}; \hat{\boldsymbol{\xi}}_{0:t}) \leq \mathbf{0} && (8.5b)\\
& \overline{\boldsymbol{\eta}}_{i,t} \perp h_{i,t}(\overline{\mathbf{x}}_{i,t}; \hat{\mathbf{x}}_{i,0:t-1}, \hat{\boldsymbol{\xi}}_{0:t}) \leq \mathbf{0}. && (8.5c)
\end{aligned}$$

Equilibrium Concepts

We are interested in pricing mechanisms that the SO can implement to promote *dispatch-following incentives*. These incentives come in two varieties: *ex-ante*, which apply before uncertainty realization and dispatch, and *ex-post*, which apply after uncertainty has been realized and dispatches have been committed. Adopting terminology from [313, 314], we now present equilibrium notions that will encourage both ex-ante and ex-post dispatch following incentives.

Definition 8.3.1. *Let $\mathbf{x}_0, \ldots, \mathbf{x}_T$ be a dispatch sequence and $\boldsymbol{\pi}_0, \ldots, \boldsymbol{\pi}_T$ be a price sequence, and let $\hat{\boldsymbol{\xi}}$ be a realization of uncertainty. This pair of sequences supports a **general equilibrium** over the entire scheduling horizon $t = 0, \ldots, T$ if and only if the following conditions hold:*

1. *Market Clearing Condition. The dispatch sequence satisfies the system-wide constraints at all times:*

$$f_t(\mathbf{x}_t, \hat{\boldsymbol{\xi}}_{0:t}) \leq \mathbf{0} \qquad \forall t \in [0, T].$$

2. *Incentive Compatibility. For each participant i, $\mathbf{x}_{i,0}, \ldots, \mathbf{x}_{i,T}$ is an optimal solution of the participant's* ex post *problem:*

$$\underset{\overline{\mathbf{x}}_{i,0},\ldots,\overline{\mathbf{x}}_{i,T}}{\arg\max} \quad \sum_{t=0}^{T} \boldsymbol{\pi}_{i,t}{}^{\top}\overline{\mathbf{x}}_{i,t} - c_{i,t}(\overline{\mathbf{x}}_{i,t}, \overline{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t}) \tag{8.6a}$$

$$\text{s.t.} \quad \overline{\boldsymbol{\mu}}_{i,t} \perp g_{i,t}(\overline{\mathbf{x}}_{i,t}; \hat{\boldsymbol{\xi}}_{0:t}) \leq \mathbf{0} \qquad \forall t \in [0, T] \tag{8.6b}$$

$$\overline{\boldsymbol{\eta}}_{i,t} \perp h_{i,t}(\overline{\mathbf{x}}_{i,t}, \overline{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t}) \leq \mathbf{0} \qquad \forall t \in [0, T]. \tag{8.6c}$$

That is, the dispatch sequence minimizes the lost opportunity cost of each participant i over the entire scheduling horizon.

A dispatch and price sequence that supports a general equilibrium supports ex-post dispatch-following incentives. However, when the SO schedules in the presence of uncertainty, e.g., in the case of multi-interval lookahead or stochastic dispatch, a missing payments problem can arise due to distribution shift. The works [313, 314] discuss this issue extensively in the lookahead setting and further show how this missing payment problem arises even when there are perfect forecasts (but a truncated lookahead horizon). To address this, they introduce an additional notion of *partial equilibrium* at each dispatch stage which may be viewed as a condition on *ex-ante* dispatch-following incentives.

Definition 8.3.2. *Let $\mathbf{x}_t$ be the dispatch and $\boldsymbol{\pi}_t$ be the price from interval t, and let $\hat{\boldsymbol{\xi}}_{0:t}$ be a realization of uncertainty up through t. This pair supports a* ***partial equilibrium*** *for stage t if and only if the following conditions hold:*

1. *Market Clearing Condition:*

$$f_t(\mathbf{x}_t, \hat{\boldsymbol{\xi}}_{0:t}) \leq \mathbf{0}.$$

2. *Incentive Compatibility: For each i, the subvector $\mathbf{x}_{i,t}$ of $\mathbf{x}_t$ is the optimal solution of (8.5) under price $\boldsymbol{\pi}_{i,t}$.*

The work in [313, 314] also adopts a dual notion of equilibrium that combines partial and general equilibrium.

Definition 8.3.3. *A dispatch sequence* $\mathbf{x}_0, \ldots, \mathbf{x}_T$ *and price sequence* $\boldsymbol{\pi}_0, \ldots, \boldsymbol{\pi}_T$ *support a* ***strong equilibrium*** *under sequentially realized uncertainty* $\hat{\boldsymbol{\xi}}_1, \ldots, \hat{\boldsymbol{\xi}}_T$ *if and only if they support both a general equilibrium and a partial equilibrium for each* t.

By employing this stronger notion of equilibrium, both ex-ante and ex-post incentive alignment can be guaranteed in the lookahead dispatch setting. We adopt this notion of strong equilibrium in our work to enable pricing that guarantees dispatch-following incentives in the case of general lookahead value function V_t, such as those in the case of stochastic optimization formulations of the market dispatch problem.

Pricing a strong equilibrium

In each interval, the market operator solves (8.2) to generate a dispatch for that interval for each participant $\mathbf{x}^*_{i,t}$ along with a price vector $\boldsymbol{\pi}^*_{i,t}$ defined as

$$\begin{aligned} \boldsymbol{\pi}^*_{i,t} := & \underbrace{-\mathsf{D}_{\mathbf{x}_{i,t}} f_t(\mathbf{x}^*_t; \hat{\boldsymbol{\xi}}_{0:t})^\top \boldsymbol{\lambda}^*_t}_{\text{Locational marginal price}} \; \underbrace{-\; \nabla_{\mathbf{x}_{i,t}} V_t(\mathbf{x}^*_{i,t}; \theta_t)}_{\text{Price of uncertainty}} \\ & \underbrace{-\mathsf{D}_{\mathbf{x}_{i,t}} h_{i,t}(\mathbf{x}^*_{i,t}, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t})^\top \boldsymbol{\eta}^*_{i,t}}_{\text{Price of intertemporal coupling}} . \end{aligned} \tag{8.7}$$

This price is defined in terms of optimal dual variables and derivatives of objective/constraint functions at the optimal point. The notation $\mathsf{D}_{\mathbf{x}_{i,t}} f_t(\mathbf{x}^*_t; \hat{\boldsymbol{\xi}}_t)$ represents the Jacobian of the function f_t with respect to variable $\mathbf{x}_{i,t}$ evaluated at $\mathbf{x}_{i,t} = \mathbf{x}^*_{i,t}$.

Our price admits a straightforward decomposition into several functional parts. The first component of the price is the standard locational marginal price (LMP). The second term prices the cost of scheduling under uncertainty. The magnitude of this term is determined both by the particular choice of V_t as well as the quality of the uncertainty parameterization in θ_t. The last component is a price on the intertemporal coupling between decisions. The price of ramping presented in [313] is a special case of this term; our formulation admits other intertemporal couplings, such as from storage state-of-charge [31]. This price is discriminatory, in that each participant may see a different price. The necessity of such price discrimination when there are intertemporal coupling constraints on generators is proven in [313].

We now establish the equilibrium properties of this price. Given the prior convexity assumptions on $c_{i,t}$, f_t, $g_{i,t}$, and $h_{i,t}$, problems (8.5) and (8.6) are convex.

Theorem 8.3.4. *Fix a $t \in [0, T]$ and let $\mathbf{x}_t^*$ be the dispatch produced by the optimal solution of (8.2) and let $\boldsymbol{\pi}_t^*$ be the price as defined in (8.7) using optimal primal/dual variables from (8.2). This dispatch-price pair forms a partial equilibrium for interval t.*

Proof. For an interval t, we have realized uncertainty $\hat{\boldsymbol{\xi}}_t$ and a previous dispatch sequence $\hat{\mathbf{x}}_{0:t-1}$. Assume that problem (8.2) has been solved to optimality yielding optimal primal/dual solutions (not necessarily unique) $\mathbf{x}_t^*$, $\boldsymbol{\lambda}_t^*$, $\boldsymbol{\mu}_{i,t}^*$, $\boldsymbol{\eta}_{i,t}^*$ $\forall i$.

The market clearing condition in Definition 8.3.2 is satisfied by primal feasibility of the optimal solution $\mathbf{x}_t^*$. Without loss of generality, the rest of the proof will be shown for a particular i. To show incentive compatibility, we write down the Lagrangian of (8.2) for a given t:

$$\begin{aligned}\mathcal{L}_t =& \sum_{i=1}^{N} c_{i,t}(\mathbf{x}_{i,t}, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t}) + V_t(\mathbf{x}_t; \theta_t) + \boldsymbol{\lambda}_t^\top f_t(\mathbf{x}_t; \hat{\boldsymbol{\xi}}_{0:t}) \\ &+ \sum_{i=1}^{N} \boldsymbol{\mu}_{i,t}^\top g_{i,t}(\mathbf{x}_{i,t}; \hat{\boldsymbol{\xi}}_{0:t}) + \sum_{i=1}^{N} \boldsymbol{\eta}_{i,t}^\top h_{i,t}(\mathbf{x}_{i,t}, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t}).\end{aligned}$$

The stationarity conditions hold at optimality:

$$\begin{aligned}\mathbf{0} =& \nabla_{\mathbf{x}_{i,t}} c_{i,t}(\mathbf{x}_{i,t}^*, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t}) + \nabla_{\mathbf{x}_{i,t}} V(\mathbf{x}_t^*; \theta_t) \\ &+ \mathsf{D}_{\mathbf{x}_{i,t}} f_t(\mathbf{x}_t^*; \hat{\boldsymbol{\xi}}_{0:t})^\top \boldsymbol{\lambda}_t^* + \mathsf{D}_{\mathbf{x}_{i,t}} g_{i,t}(\mathbf{x}_{i,t}^*; \hat{\boldsymbol{\xi}}_{0:t})^\top \boldsymbol{\mu}_{i,t}^* \\ &+ \mathsf{D}_{\mathbf{x}_{i,t}} h_{i,t}(\mathbf{x}_{i,t}^*, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t})^\top \boldsymbol{\eta}_{i,t}^*.\end{aligned} \tag{8.8}$$

The argument uses the convex KKT theorem. We construct primal-dual solutions that satisfy the KKT optimality conditions (primal/dual feasibility, complementary slackness, and stationarity) of problem (8.5). Because (8.5) is convex, the constructed primal-dual solution is also optimal. Define

$$\overline{\mathbf{x}}_{i,t} := \mathbf{x}_{i,t}^*, \tag{8.9a}$$

$$\overline{\boldsymbol{\mu}}_{i,t} := \boldsymbol{\mu}_{i,t}^*, \tag{8.9b}$$

$$\overline{\boldsymbol{\eta}}_{i,t} := \mathbf{0}. \tag{8.9c}$$

$\overline{\mathbf{x}}_{i,t}$ satisfies primal feasibility of (8.5) because $\mathbf{x}_{i,t}^*$ is primal feasible for (8.2). $\overline{\boldsymbol{\mu}}_{i,t}$ and $\overline{\boldsymbol{\eta}}_{i,t}$ are dual feasible because both are non-negative by construction. Complementary slackness holds for $\overline{\boldsymbol{\mu}}_{i,t}$ because $\boldsymbol{\mu}_{i,t}^*$ is optimal for (8.2), and holds for $\overline{\boldsymbol{\eta}}_{i,t}$ trivially.

The Lagrangian of (8.5) is

$$\begin{aligned}\mathcal{L}_{i,t} =& -{\boldsymbol{\pi}_{i,t}^*}^\top \overline{\mathbf{x}}_{i,t} + c_{i,t}(\overline{\mathbf{x}}_{i,t}, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t}) \\ &+ \overline{\boldsymbol{\mu}}_{i,t}^\top g_{i,t}(\overline{\mathbf{x}}_{i,t}; \hat{\boldsymbol{\xi}}_{0:t}) + \overline{\boldsymbol{\eta}}_{i,t}^\top h_{i,t}(\overline{\mathbf{x}}_{i,t}, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_{0:t}).\end{aligned} \tag{8.10}$$

Now to check the stationarity condition,

$$\begin{aligned}\nabla_{\overline{\mathbf{x}}_{i,t}} \mathcal{L}_{i,t} &= -\boldsymbol{\pi}_{i,t}^* + \nabla_{\overline{\mathbf{x}}_{i,t}} c_{i,t}(\mathbf{x}_{i,t}^*, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_t) \\ &\quad + \mathsf{D}_{\overline{\mathbf{x}}_{i,t}} g_{i,t}(\mathbf{x}_{i,t}^*; \hat{\boldsymbol{\xi}}_{0:t})^\top \boldsymbol{\mu}_{i,t}^* + \mathbf{0} \\ &= \mathsf{D}_{\mathbf{x}_{i,t}} f_t(\mathbf{x}_t^*; \hat{\boldsymbol{\xi}}_{0:t})^\top \boldsymbol{\lambda}_t^* + \nabla_{\mathbf{x}_{i,t}} V(\mathbf{x}_t^*; \theta_t) \\ &\quad + \mathsf{D}_{\mathbf{x}_{i,t}} h_{i,t}(\mathbf{x}_{i,t}^*; \hat{\mathbf{x}}_{i,t-1}, \hat{\boldsymbol{\xi}}_t)^\top \boldsymbol{\eta}_{i,t}^* \\ &\quad + \nabla_{\overline{\mathbf{x}}_{i,t}} c_{i,t}(\mathbf{x}_{i,t}^*, \hat{\mathbf{x}}_{i,0:t-1}; \hat{\boldsymbol{\xi}}_t) \\ &\quad + \mathsf{D}_{\overline{\mathbf{x}}_{i,t}} g_{i,t}(\mathbf{x}_{i,t}^*; \hat{\boldsymbol{\xi}}_{0:t})^\top \boldsymbol{\mu}_{i,t}^* \\ &= \mathbf{0},\end{aligned}$$

where the first equality comes by from plugging (8.9) into (8.10) and the second equality comes from plugging in the price defined in (8.7). The third equality holds because the expression is identical to (8.8). □

Theorem 8.3.5. *The sequences of dispatches $\mathbf{x}_0^*, \ldots, \mathbf{x}_T^*$ and prices $\boldsymbol{\pi}_0^*, \ldots, \boldsymbol{\pi}_T^*$ produced by Algorithm 8.1 over the entire scheduling horizon form a general equilibrium.*

Proof. This result uses the same approach as for Theorem 8.3.4. We construct a primal-dual solution for the individual participant's ex-post LOC problem (8.6) from the primal-dual variables computed over the scheduling horizon with Algorithm 8.1 and then show that this solution is optimal. The intertemporal coupling and uncertainty terms allow the Lagrangian of (8.6) to decouple across intervals and thus the optimality conditions of (8.2) apply simultaneously.

First, the market clearing condition is satisfied because the constraint (8.2b) holds for every t.

The Lagrangian of the individual participant's ex-post LOC problem (8.6) is

$$\begin{aligned}\mathcal{L}_{i,t} = \sum_{t=0}^{T} & -\boldsymbol{\pi}_{i,t}^{*\top} \overline{\mathbf{x}}_{i,t} + c_{i,t}(\mathbf{x}_{i,t}^*; \hat{\boldsymbol{\xi}}_t) + \overline{\boldsymbol{\eta}}_{i,t}^\top g_{i,t}(\overline{\mathbf{x}}_{i,t}; \hat{\boldsymbol{\xi}}_t) \\ & + \overline{\boldsymbol{\nu}}_{i,t}^\top h_{i,t}(\overline{\mathbf{x}}_{i,t}, \overline{\mathbf{x}}_{i,t-1}; \hat{\boldsymbol{\xi}}_t).\end{aligned} \tag{8.11}$$

Define

$$\overline{\mathbf{x}}_{i,t} := \mathbf{x}_{i,t}^* \qquad \forall t, \tag{8.12a}$$

$$\overline{\boldsymbol{\eta}}_{i,t} := \boldsymbol{\eta}_{i,t}^* \qquad \forall t, \tag{8.12b}$$

$$\overline{\boldsymbol{\nu}}_{i,t} := \mathbf{0} \qquad \forall t. \tag{8.12c}$$

Primal/dual feasibility and complementary slackness follow from the same argument as in Theorem 8.3.4. The individual's stationarity condition, which must hold across the entire time horizon, follows from

$$\begin{aligned}
\nabla_{\bar{\mathbf{x}}_{i,t}} \mathcal{L}_{i,t} &= \sum_{t=0}^{T} -{\boldsymbol{\pi}_{i,t}^{*}}^{\top} + \nabla_{\bar{\mathbf{x}}_{i,t}} c_{i,t}(\mathbf{x}_{i,t}^{*}; \hat{\boldsymbol{\xi}}_t) \\
&\quad + \mathsf{D}_{\bar{\mathbf{x}}_{i,t}} g_{i,t}(\mathbf{x}_{i,t}^{*}; \hat{\boldsymbol{\xi}}_t)^{\top} \boldsymbol{\eta}_{i,t}^{*} \\
&\quad + \mathsf{D}_{\bar{\mathbf{x}}_{i,t}} h_{i,t}(\mathbf{x}_{i,t}^{*}; \hat{\mathbf{x}}_{i,t-1}, \hat{\boldsymbol{\xi}}_t)^{\top} \boldsymbol{\nu}_{i,t}^{*} \\
&= \sum_{t=0}^{T} {A_t^{(i)}}^{\top} \boldsymbol{\lambda}_t^{*} + \mathsf{D}_{\mathbf{x}_{i,t}} f_t(\mathbf{x}_t^{*}; \hat{\boldsymbol{\xi}}_t)^{\top} \boldsymbol{\mu}_t^{*} + \nabla_{\mathbf{x}_{i,t}} V(\mathbf{x}_t^{*}; \theta_t) \\
&\quad + \nabla_{\bar{\mathbf{x}}_{i,t}} c_{i,t}(\mathbf{x}_{i,t}^{*}; \hat{\boldsymbol{\xi}}_t) + \mathsf{D}_{\bar{\mathbf{x}}_{i,t}} g_{i,t}(\mathbf{x}_{i,t}^{*}; \hat{\boldsymbol{\xi}}_t)^{\top} \boldsymbol{\eta}_{i,t}^{*} \\
&\quad + \mathsf{D}_{\mathbf{x}_{i,t}} h_{i,t}(\mathbf{x}_{i,t}^{*}; \hat{\mathbf{x}}_{i,t-1}, \hat{\boldsymbol{\xi}}_t)^{\top} \boldsymbol{\nu}_{i,t}^{*} \\
&= \mathbf{0}.
\end{aligned}$$

□

The last equality holds because the equality (8.8) holds for each t. This reveals the motivation of the price construction in (8.7). Including a term for the intertemporal coupling constraints $h_{i,t}$ allows the pricing problem to decouple across intervals. Thus, the participant could leave the market after any interval and their LOC would be 0.

The result in Theorem 8.3.5 shows that the price (8.7) guarantees that each participant has zero lost opportunity cost at the end of the scheduling horizon. The intertemporal coupling term compensates participants for any lost opportunity cost due to binding intertemporal constraints (e.g., ramping) whereas the uncertainty term compensates participants for any lost opportunity cost due to the system operator's uncertainty-aware scheduling procedure.

The following corollary holds from Theorems 8.3.4 and 8.3.5:

Corollary 8.3.6. *The sequences of dispatches $\mathbf{x}_0^*, \ldots, \mathbf{x}_T^*$ and prices $\boldsymbol{\pi}_0^*, \ldots, \boldsymbol{\pi}_T^*$ produced by Algorithm 8.1 over the entire scheduling horizon support a strong equilibrium.*

A strong equilibrium is a desirable property of a market-clearing price because it provides dispatch-following incentives during each stage of scheduling horizon while also correcting the missing payment problem that arises ex-post.

Table 8.1: Generator parameters for the test case

Generator	Pmin (MW)	Pmax (MW)	Ramp Rate (% Pmax/hour)	Cost ($/MWh)
Gas C.C.	350	550	25%	50
Gas Peaker	100	120	200%	70
Solar	0	250	NA	0
Wind	0	350	NA	0

8.4 Experiments

We explore how uncertainty affects dispatch efficiency and pricing under our mechanism through a test case similar to that presented in [314]. We consider a power system with a gas combined-cycle (C.C.) plant, a gas peaker plant, solar, wind, and load in a single bus network. The gas plants are ramp constrained whereas the renewables are not. Cost functions are linear and are parameterized by their marginal cost. All parameters for the generators are given in Table I.

We obtain 24-hr load and renewable generation profiles from CAISO from Sep. 9, 2021 [323]. These include both forecast day-ahead trajectories and the actual, realized real-time trajectories, all of which were normalized to 1000MW peak demand. Sample realizations of the true trajectories are simulated by adding correlated zero-mean Gaussian noise to the actual values.

Algorithm 8.1 was implemented to clear the market in a rolling fashion. The dispatch horizon for a single run of the market is 24 hours, consisting of 289 individual stages: one DA dispatch and RT dispatches every 5 minutes. The first stage ($t = 0$) is the DA unit commitment problem. The unit commitment problem makes use of a 24-hour ahead hourly DA forecast in the CAISO data.[2] The subsequent RT stages take the unit commitment as fixed.

We implement the three mechanisms from Section 8.2 for dispatching in RT. First is myopic scheduling, where only the current interval's cost and constraints are optimized but generator ramping constraints bind the current decision to the realized dispatch from the previous interval. This is a deterministic problem, as demand and renewable generation are assumed to be known, and does not account for the cost of future decisions in the scheduling horizon. Second is multi-interval lookahead scheduling with a 3-hour lookahead horizon. A lookahead forecast is

[2]In North American ISOs, there is often a financial settlement in the DA market. Although our formulations accommodate a financially settled DA market, we do not empirically analyze the DA market settlement in this work, as intertemporal coupling and uncertainty do not arise in the formation of the DA prices.

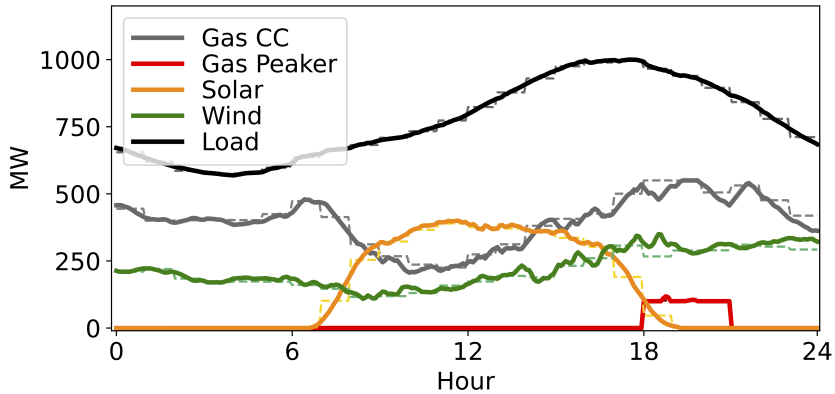


Figure 8.2: DA (dashed line) and optimal RT (solid line) dispatch trajectories for generators and load over a 24 hour scheduling horizon.

computed by taking the mean of a small subset of random trajectories. Third is a multi-interval chance-constrained lookahead problem, where the constraints for the advisory periods hold probabilistically and the objective function is the expected cost for the advisory periods. The distribution parameters of the trajectory forecasts are obtained from the set of randomly sampled trajectories.

Figure 8.2 shows the dispatch trajectories for each of the generators in the system under optimal *ex-post* scheduling. Note that due to its high cost relative to the other generators, the gas peaker is only active during the peak demand hours when the ramp needs of the system exceed available capacity. Lookahead dispatch with point forecasts results in dispatching the peaker less often for binding ramping constraints but more during other intervals due to the cost of uncertain dispatch. Chance-constrained lookahead dispatch is able to avoid most of the binding ramping constraints at the expense of more precautionary dispatches due to uncertainty.

Figure 8.3 presents the benefits of scheduling with lookahead and stochastic forward cost policies. When the forecast error is zero, accounting for forward cost results in more costly dispatch than myopic scheduling. This is due to the inherent conservatism and robustness that these policies provide. However, as uncertainty increases, myopic scheduling becomes more costly than uncertainty-aware scheduling due to load shedding actions and sub-optimal dispatch of higher cost generators.

Finally, we show how our proposed market clearing price (8.7) decomposes into its constituent components in Figure 8.4. The largest component of the price is the uniform shadow price of the power balance constraint. However, for the ramp-constrained gas generator, there are additional terms that compensate for the opportunity cost of the system operator's imperfect scheduling under uncertainty.

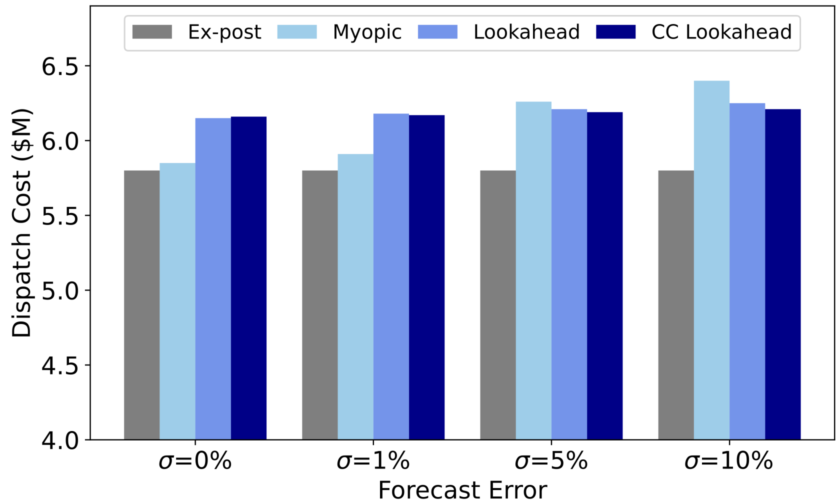


Figure 8.3: Total dispatch cost of the different pricing schemes under increasing forecast error. Forecast error is defined as the mean absolute percentage deviation from the true trajectory realization.

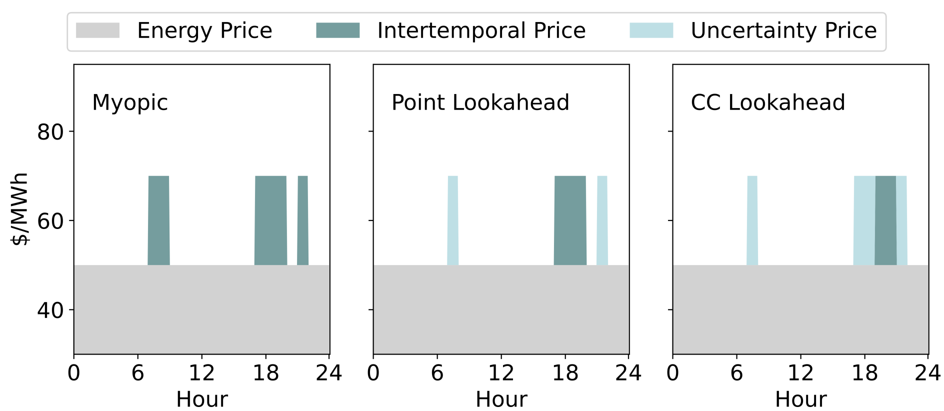


Figure 8.4: Price trajectory $\boldsymbol{\pi}^*_{i,t}$ for the gas combined-cycle generator under $\sigma = 10\%$ forecast uncertainty for different real-time forecast methodologies.

The additional complexity of computing price (8.7) is negligible compared to the standard LMP and T-LMP in [313], as it is also defined in terms of optimal dual variables and cost function gradients. The complexity of the dispatch problem depends on the choice of procedure (e.g., change-constrained, robust, scenario).

8.5 Conclusion

In this chapter, we present a unified mechanism for pricing uncertainty in a multi-stage dispatch setting, incorporating both standard deterministic lookahead dispatch and stochastic market clearing approaches (e.g., chance-constrained, robust) within the same pricing framework. We prove that our price provides dispatch following incentives as well as zero lost opportunity costs for generators. Potential areas of future work include a detailed empirical comparison with other pricing methodologies, such as the standard LMP and the R-TLMP [313], analyzing the system operator's merchandizing surplus under this pricing mechanism, and comparing multi-settlement and single-settlement pricing methodologies.

Chapter 9

FAST AND RELIABLE $N-k$ CONTINGENCY SCREENING WITH INPUT-CONVEX NEURAL NETWORKS

In the last few chapters, we considered questions pertaining to risk and uncertainty in online decision-making, learning, and pricing. However, there are many other notions of reliability—besides risk- and uncertainty-awareness—that we might wish to incorporate into the training and deployment of machine learning models. In this final chapter of the thesis, we consider a setting where a similar conceptual toolkit to the end-to-end learning approach in Chapter 7 can be developed into a framework to enforce reliability—or controlled false negative rate—of machine learning classifiers for the problem of contingency screening in power grids.

Power system operators must ensure that dispatch decisions remain feasible in case of grid outages or contingencies to prevent cascading failures and ensure reliable operation. However, checking the feasibility of all $N-k$ contingencies—every possible simultaneous failure of k grid components—is computationally intractable even for small k, requiring system operators to resort to heuristic screening methods. Because of the increase in uncertainty and changes in system behaviors, heuristic lists might not include all relevant contingencies, generating false negatives in which unsafe scenarios are misclassified as safe. In this work, we propose to use input-convex neural networks (ICNNs) for contingency screening. We show that ICNN reliability can be determined by solving a convex optimization problem, and by scaling model weights using this problem as a differentiable optimization layer during training, we can learn an ICNN classifier that is both data-driven and has provably guaranteed reliability. That is, our method can ensure a zero false negative rate. We empirically validate this methodology in a case study on the IEEE 39-bus test network, observing that it yields substantial (10-20$\times$) speedups while having excellent classification accuracy.

This chapter is primarily based on the following paper:

[1] N. Christianson, W. Cui, S. Low, W. Yang, and B. Zhang, "Fast and Reliable $N-k$ Contingency Screening with Input-Convex Neural Networks," in *Proceedings of the 7th Annual Learning for Dynamics & Control Con-*

ference, PMLR, Jun. 2025, pp. 527–539. [Online]. Available: https://proceedings.mlr.press/v283/christianson25a.html.

9.1 Introduction

Power systems face increasing uncertainty due to increasing variable renewable generation and environmental factors such as extreme weather events and wildfires. To ensure reliable operation in the face of this growing uncertainty, power system operators must dispatch generation resources in a manner that anticipates and is robust to potential asset outages, such as the failure of a transmission line [53, 324]. Failing to anticipate and prepare for such outages can lead to cascading failures that may necessitate load shedding, as occurred in the Texas blackouts in 2021 [325].

To assess and plan for the impacts of potential asset failures before they occur, system operators must perform contingency analysis to identify which failures will result in a post-failure operating point that is infeasible [27, Chapter 3]. In particular, NERC regulations mandate that US power systems remain stable for all $N-1$ contingencies—contingencies involving the loss of a single asset—and that system operators plan for the multi-element contingencies with the most severe consequences [326]. Assessing the security of and planning for such $N-k$ contingencies—simultaneous losses of $k > 1$ assets—is crucial for reliable system operation, as such correlated failures can cause severe blackouts, such as the 2003 Northeast blackout [327]. However, the complexity of contingency analysis grows exponentially in the number of simultaneous failures k that is considered: in a system with N components, the number of such contingencies is $\Omega(N^k)$, which quickly becomes intractable for $k > 1$ in large-scale power systems.

To combat this intractability and enable the efficient screening of $N-k$ contingencies for $k > 1$, a number of approaches have been developed in the recent literature. These methods fall into one of two categories: (1) heuristic approaches using, e.g., machine learning to predict contingency feasibility, and (2) exact methods that reduce computational expense by certifying feasibility of a collection of contingencies using "representative" constraints. However, these methods fall short on two fronts. The heuristic approaches (1) come with no rigorous guarantees on prediction accuracy or reliability; thus, they might misclassify a critical contingency as feasible, causing system outages. On the other hand, the exact methods (2), while reliable,

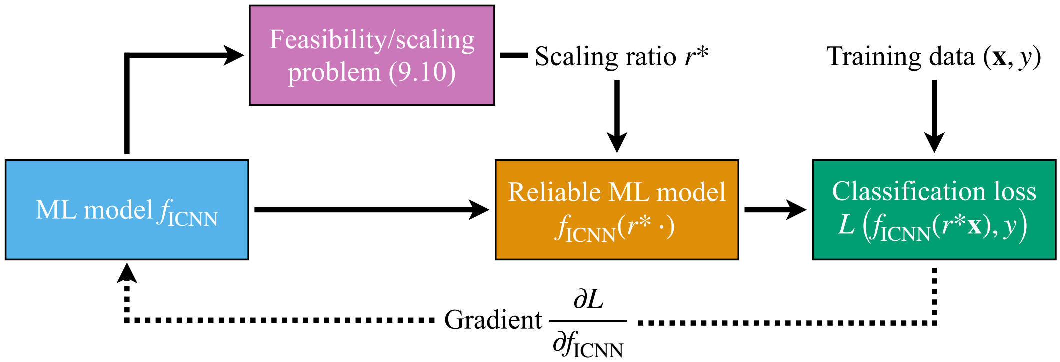


Figure 9.1: A schematic of our proposed methodology for training reliable classifiers for contingency screening; see Algorithm 13 for a full description. Note that the scaling ratio r^* is computed using a differentiable convex optimization layer, so the gradient $\partial L/\partial f_{\text{ICNN}}$ is aware of this scaling step.

are typically hand-designed rules which cannot take advantage of historical data to accelerate contingency analysis by focusing on the most relevant or likely contingencies for a particular power system. To enable reliable and efficient screening of higher-order $N-k$ contingencies in modern power systems, new approaches are needed to bridge the data-driven paradigm of machine learning with the strong reliability guarantees of exact methods.

Contributions

In this work, we confront this challenge, proposing a machine learning approach to screening $N-k$ contingencies that is fast, data-driven, and comes with *provable* guarantees on reliability. In particular, we propose to use *input-convex neural networks* (ICNNs) to screen arbitrary collections of contingencies for infeasibility. We define a *reliable* classifier as one that never misclassifies an infeasible contingency as feasible—that is, one that makes no false negative predictions—and we show that ICNN reliability can be certified by solving a collection of convex optimization problems (Proposition 9.3.1). Furthermore, we show that an unreliable ICNN can be transformed into a reliable one with zero false negative rate by suitably scaling model parameters by the solution to a convex optimization problem (Theorem 9.3.2), and we propose a training methodology that enables learning over the restricted set of *provably reliable ICNNs* by applying this scaling during training via a differentiable convex optimization layer (Theorem 9.4.1, Algorithm 13). This fully differentiable approach ensures that the scaling procedure and its dependence on model parameters are accounted for during gradient descent updates; see Figure 9.1 for a diagram outlining this approach.

Our approach allows for **trading off the online computational burden of contingency screening for an offline one**: it requires a significant computational investment during the training procedure to guarantee model reliability, but at deployment time, screening for contingencies only requires a single feedforward pass of the ICNN. We test our approach in a case study on the IEEE 39-bus test system, finding that it yields significant (10-20 $\times$) speedups in runtime while ensuring zero false negative rate and excellent (2-5%) false positive rate (Section 9.5). In addition, our approach yields an ICNN parametrizing an inner approximation to the set of network injections that are feasible across contingencies, which enables 10 $\times$ faster preventive dispatch via security-constrained optimal power flow (Section 9.5). We anticipate that our proposed approach to learning efficient data-driven inner approximations to complex feasible sets using ICNNs could be of broader interest for other applications in energy systems and control.

Related Work

Our work contributes to four areas in the power systems and machine learning literature.

Power system contingency analysis. The problem of assessing the feasibility of contingencies has been studied in the power systems community for decades as a foundational part of secure grid operation [324]. Much work in recent years has sought to develop faster methods for contingency analysis, including exact methods that do not sacrifice reliability [328–330] as well as heuristic and machine learning approaches that achieve faster speeds at the expense of accuracy [52, 53, 331]. Closest to our work is that of [332], which proposes an approach using "representative constraints" to reduce the number of contingencies that must be considered; these representative constraints constitute an inner approximation of the set of all injections that are feasible across contingencies, just as our approach yields an ICNN-parametrized inner approximation to this set. In contrast to all prior approaches, our approach is both data-driven—using ICNN models, which have substantial representational efficiency [278, Theorem 2], to learn from system data—and ensures rigorous guarantees on reliability, enabling fast and accurate contingency screening without any false negative predictions.

Convex inner approximations in power systems. The design of tractable, convex inner approximations to complicated convex or nonconvex sets is a widely studied problem in power systems and control, with applications to problems such as AC-optimal power flow [333–335] and aggregate flexibility of electric vehicles [336]. When the set one wishes to approximate is convex, our approach could be adapted to enable learning such inner approximations in a data-driven manner, yielding greater efficiency and a better approximation due to the representational capacity of ICNNs.

Machine learning in power systems. Machine learning techniques have been applied to a wide range of problems in power systems, including contingency analysis [52, 53], optimal power flow [56, 57], and security-constrained optimal power flow [54, 337]. Of particular note in this direction are the papers [55, 338, 339], which specifically apply *ICNNs* to the problems of voltage control and optimal power flow. While some of the works applying machine learning to optimal power flow obtain generalization guarantees or provable constraint satisfaction for their methods, these guarantees hold specifically for the dispatch problem and cannot be extended to yield faster reliable approaches for contingency analysis. Thus, we give the first machine learning approach to contingency analysis with *provable* guarantees on model accuracy.

Robust and reliable machine learning. A number of approaches have been developed to train machine learning models that are reliable in some sense, including methods to control the false positive/negative rates of a classifier [340, 341] and neural network verification and transformation techniques [342–344]. Recently, the field of *learning-augmented* algorithms [69, 70] has developed new approaches to incorporate untrusted or "black-box" machine learning predictions into decision-making problems, including a number of energy-related problems [71, 74, 133, 193]. In contrast to these prior approaches, our methodology enables learning data-driven ICNN models for contingency classification that are *reliable by design*, with zero false negative rate enforced during training via a differentiable convex optimization layer.

Notation

Let $\mathbb{R}_+$ denote the nonnegative reals. Given a vector $\mathbf{x} \in \mathbb{R}^n$, we denote its ith entry x_i; similarly, given a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, its ith row is denoted $\mathbf{m}_i$ and its ijth entry is denoted m_{ij}. Given $n \in \mathbb{N}$, we define $[n] = \{1, \dots, n\}$, and given a set $\mathcal{X}$, we define $\mathcal{P}(\mathcal{X})$ as its power set. Given a set $\mathcal{A} \subseteq \mathbb{R}^n$, $\operatorname{int}\mathcal{A}$ denotes its interior and $\operatorname{vol}(\mathcal{A})$ denotes its volume.

9.2 Model and Preliminaries

We begin by reviewing power network economic dispatch via the DC-optimal power flow problem and the problem of screening for infeasible contingencies. We then describe our classification approach to contingency screening and the *input-convex* neural networks we employ to this end.

DC-OPF and Contingency Screening

Consider a power network with topology represented by a graph $G = (V, E)$, where V is the set of nodes/buses and E is the set of edges/transmission lines. Let $n = |V|$ be the number of buses and $m = |E|$ be the number of lines. Without loss of generality, we will assume that each bus $i \in [n]$ has a single generator.

To dispatch generation while minimizing cost and satisfying demand and other constraints in large-scale transmission networks, system operators typically solve the DC-optimal power flow (OPF) problem, which considers a linearized model of power flow [345]. In this problem, the system operator is faced with a known vector $\mathbf{d} \in \mathbb{R}^n$ of (net) demands across buses, and in response chooses generator dispatches $\mathbf{p} \in \mathbb{R}^n$ to minimize cost while satisfying several operational constraints:

$$\min_{\mathbf{p}\in\mathbb{R}^n} \quad \sum_{i\in[n]} c_i(p_i) \tag{9.1a}$$

$$\text{s.t.} \quad \underline{\mathbf{p}} \le \mathbf{p} \le \overline{\mathbf{p}} \tag{9.1b}$$

$$\mathbf{1}^\top(\mathbf{p} - \mathbf{d}) = 0 \tag{9.1c}$$

$$\underline{\mathbf{f}} \le \mathbf{H}(\mathbf{p} - \mathbf{d}) \le \overline{\mathbf{f}}. \tag{9.1d}$$

Here, $c_i(p_i)$ is the cost for the generation decision p_i on generator i, the constraint (9.1b) enforces lower and upper capacity limits $\underline{\mathbf{p}}, \overline{\mathbf{p}} \in \mathbb{R}^n$ on generation, (9.1c) enforces supply-demand balance, and (9.1d) enforces the lower and upper bounds $\underline{\mathbf{f}}, \overline{\mathbf{f}} \in \mathbb{R}^m$ on line power flows given the nodal net injection vector $\mathbf{p}-\mathbf{d}$. The matrix $\mathbf{H}$ mapping from nodal net power injections to line power flows is specifically defined as $\mathbf{H} := \mathbf{B}\mathbf{C}^\top\mathbf{L}^\dagger$, where $\mathbf{B} \in \mathbb{R}^{m\times m}$ is the diagonal matrix of line admittances, $\mathbf{C} \in \mathbb{R}^{n\times m}$ is a bus-by-line directed incidence matrix with entries defined as

$$c_{jl} = \begin{cases} +1 & \text{if line } l = j \to k \text{ for some } k \in V \\ -1 & \text{if line } l = i \to j \text{ for some } i \in V \\ 0 & \text{otherwise,} \end{cases}$$

for some arbitrary orientation on the lines E, and $\mathbf{L} = \mathbf{C}\mathbf{B}\mathbf{C}^\top$ is the admittance-weighted network Laplacian.

In the DC-OPF problem (9.1), the system operator solves for a feasible dispatch vector given a nominal network topology G. In practice, however, after a dispatch decision is chosen and the net nodal power injections $\mathbf{x} := \mathbf{p} - \mathbf{d}$ are fixed, the network topology might change due to the failure of one or more lines. As a result of this contingency, the matrix $\mathbf{H}$ mapping net power injections to line power flows will change, causing the line flows to redistribute and potentially violate the line flow limits (9.1d). Such violations may cause further lines to trip, causing a cascade of failures [27, Chapter 4]. Thus, to ensure continued feasible and reliable operation, the system operator must determine which contingencies are infeasible and must be planned for. This is the *contingency analysis* problem, which is defined formally as follows.[1]

Problem 9.0 (**Contingency Analysis**). *Let $C \subseteq \mathcal{P}([m])$ be a set of contingencies of interest, where each $c \in C$ represents a set of failed lines, and let $\mathbf{x} = \mathbf{p} - \mathbf{d} \in \mathbb{R}^n$ be a vector of nodal net power injections. In the* ***contingency analysis*** *problem, the system operator seeks to determine whether the net injection $\mathbf{x}$ yields feasible line flows for each contingency $c \in C$—that is, whether*

$$\underline{\mathbf{f}} \leq \mathbf{H}_c \mathbf{x} \leq \overline{\mathbf{f}}$$

for each $c \in C$, where $\mathbf{H}_c = \mathbf{B}_c \mathbf{C}_c^\top \mathbf{L}_c^\dagger$ is defined for the post-contingency network topology with lines $E \setminus c$.

A standard choice for the set of reference contingencies C is the collection of all $N - k$ contingencies, or the set of all possible simultaneous failures of up to k lines; in this case,

$$C = \{c \subseteq [m] : 1 \leq |c| \leq k\}.$$

In practice, however, it is impractical to check the feasibility of all possible $N - k$ contingencies in real time for even moderately small k: in a network with m lines, there are $\Omega(m^k)$ such possible contingencies, and so the complexity of $N - k$ contingency analysis grows exponentially with k. Instead, system operators typically only consider the $N - 1$ case, augmented with a small number of representative or problematic higher-order contingencies selected via heuristic methods. Such

[1] Given a change in network topology resulting from a contingency, infeasibility could arise in either the line flow limits (9.1d) or the supply-demand balance constraint (9.1c); the latter is possible only in the case of *islanding* contingencies which disconnect the network into multiple connected components. Because the set of islanding contingencies can be determined in advance, in this work we will restrict our focus only to the set of non-islanding contingencies and the feasibility of the line limits (9.1d).

heuristics work well most of the time, since typically only a small number of contingencies are likely either to occur or to cause system infeasibility. However, they give no guarantees on system (in)feasibility for the broader set of possible $N-k$ contingencies for $k > 1$.

In this work, we seek to develop methods that can efficiently check whether a net injection $\mathbf{x}$ is feasible for *all* contingencies in some general, large reference set C, such as the set of all $N - k$ contingencies for $k > 1$. To this end, we introduce the *contingency screening* problem as a coarse-grained version of the contingency analysis problem.

Problem 9.1 (Contingency Screening). *Let $C \subseteq \mathcal{P}([m])$ be a set of contingencies of interest, and let $\mathbf{x} \in \mathbb{R}^n$ be a vector of nodal net power injections. In the* ***contingency screening*** *problem, the system operator seeks to determine whether the net injection $\mathbf{x}$ is feasible for* ***all*** *contingencies $c \in C$—that is, whether $\mathbf{x}$ is in the* ***feasible region***

$$\mathcal{F}_C \coloneqq \left\{\mathbf{y} \in \mathbb{R}^n : \underline{\mathbf{f}} \leq \mathbf{H}_c\mathbf{y} \leq \overline{\mathbf{f}} \quad \forall c \in C\right\}, \tag{9.2}$$

where each $\mathbf{H}_c = \mathbf{B}_c\mathbf{C}_c^\top\mathbf{L}_c^\dagger$ is defined for the post-contingency network topology with lines $E \setminus c$.

The (true) feasible region $\mathcal{F}_C$ defined above is the set of all net injections which remain feasible under any contingency in the set C. For notational convenience, in the rest of the chapter we write this set abstractly as

$$\mathcal{F}_C \coloneqq \{\mathbf{y} \in \mathbb{R}^n : \mathbf{A}\mathbf{y} \leq \mathbf{b}\}, \tag{9.3}$$

where $\mathbf{A} \in \mathbb{R}^{2m|C|\times n}$ and $\mathbf{b} \in \mathbb{R}^{2m|C|}$ collect all of the constraints $\underline{\mathbf{f}} \leq \mathbf{H}_c\mathbf{y} \leq \overline{\mathbf{f}}$ in (9.2). We will assume that $\mathbf{A}$ contains no zero rows, since these would encode vacuous constraints. We will also make the following mild assumptions on the structure of $\mathcal{F}_C$.

Assumption 9.1. *$\mathcal{F}_C$ is a strict subset of $\mathbb{R}^n$ whose interior contains the origin: $\mathcal{F}_C \subsetneq \mathbb{R}^n$ and $\mathbf{0} \in \operatorname{int}\mathcal{F}_C$. Equivalently, $\mathbf{A}$ has at least one row and $\mathbf{A}\mathbf{0} < \mathbf{b}$.*

Note that these assumptions are reasonable: the first simply means that $\mathcal{F}_C$ encodes *some* constraint; if it does not, then there is no need to perform contingency screening. The second assumption amounts to the condition that the lower and upper line limits $\underline{\mathbf{f}}, \overline{\mathbf{f}}$ are bounded away from zero, which should hold in practice.

The contingency *screening* problem differs from the problem of contingency *analysis* in that it focuses on feasibility across the entire reference set of contingencies $\mathcal{C}$, rather than the feasibility of individual contingencies. We can thus frame contingency screening as a binary classification problem where one seeks to classify a net injection vector $\mathbf{x} \in \mathbb{R}^n$ as feasible or infeasible, and true labels are given by the indicator function $f_{\mathcal{C}}$ defined as

$$f_{\mathcal{C}}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \mathcal{F}_{\mathcal{C}} \text{ (feasible)} \\ 1 & \text{otherwise (infeasible).} \end{cases}$$

While at first glance this might appear to be a simpler problem than the full contingency analysis problem, determining whether some injection $\mathbf{x} \in \mathcal{F}_{\mathcal{C}}$ (i.e., computing the label $f_{\mathcal{C}}(\mathbf{x})$) still has complexity $\Omega(m|\mathcal{C}|)$ in general, as it requires checking the feasibility of each contingency in $\mathcal{C}$. This feasibility verification is tractable given sufficient time and computational resources, but it will generally take too long for real-time operation when the network and contingency set are large. If approximations suffice, we could instead use techniques from machine learning to learn a more computationally efficient approximation of the function $f_{\mathcal{C}}$ in a data-driven fashion using, e.g., neural networks; however, this computational speedup will typically come at the expense of reduced classification accuracy. In particular, a generic machine learning classifier might suffer *false negatives*, where it classifies injections as feasible when they are not. While false positives (misclassifying a feasible injection as infeasible) may simply cause increased caution, false negatives pose a serious threat to reliable power system operation, since an infeasible injection that is not identified as such could lead to a cascade of failures.

While the machine learning literature has developed a number of techniques to reduce the incidence of false negatives in classification, such as increasing the loss weight of examples in the positive class, none of these techniques can yield *provably* guaranteed control over the false negative rate. To confidently deploy machine learning methods to contingency screening, they should ideally avoid any false negative predictions; we call such a classifier *reliable*.

Definition 9.2.1. *A classifier $f : \mathbb{R}^n \to \{0, 1\}$ for the contingency screening problem (Problem 9.1) is said to be **reliable** if it has zero false negative rate, i.e., if*

$$f(\mathbf{x}) = 0 \textit{ implies } \mathbf{x} \in \mathcal{F}_{\mathcal{C}}$$

for any $\mathbf{x} \in \mathbb{R}^n$.

Note that a reliable classifier f is exactly one whose *predicted feasible region* $\{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) = 0\}$ is contained inside the true feasible region $\mathcal{F}_C$; that is, the predicted feasible region should be an inner approximation of the true feasible region. Our goal in this work is to develop an approach for training reliable ML classifiers for contingency screening that satisfy this property. For general machine learning models and classification problems, determining whether this containment property holds is not typically tractable. However, as we will see in Section 9.3, the convex polyhedral structure of the true feasible region $\mathcal{F}_C$ enables tractably answering this question when we restrict to a class of *convex* neural networks.

Input-Convex Neural Networks

Input-convex neural networks (ICNNs) [287] are a restricted class of neural networks that parametrize convex functions. We consider feed-forward ICNNs $f_{\text{ICNN}} : \mathbf{x} \mapsto \mathbf{y}$ with k hidden layers of the form

$$\begin{aligned}
\mathbf{z}_1 &= \operatorname{ReLU}\left(\mathbf{D}_1\mathbf{x} + \mathbf{b}_1\right) \\
\mathbf{z}_i &= \operatorname{ReLU}\left(\mathbf{W}_{i-1}\mathbf{z}_{i-1} + \mathbf{D}_i\mathbf{x} + \mathbf{b}_i\right) \qquad \text{for } i = 2, \ldots, k \\
\mathbf{y} &= \mathbf{W}_k\mathbf{z}_k + \mathbf{D}_{k+1}\mathbf{x} + \mathbf{b}_{k+1},
\end{aligned} \tag{9.4}$$

where $\mathbf{z}_i$ is the ith hidden layer, the intermediate activation function is $\operatorname{ReLU}(x) = \max\{x, 0\}$, and the the weight matrices $\mathbf{W}_i$ are all assumed to have nonnegative entries (the weights $\mathbf{D}_i$ can have arbitrary entries). It is relatively straightforward to see that, under these assumptions (and more generally in the case of convex, nondecreasing activation functions), $f_{\text{ICNN}}(\mathbf{x})$ is convex in $\mathbf{x}$ [287, Proposition 1]. Moreover, given sufficient depth and width, ICNNs can approximate *any* convex function arbitrarily well [278, Theorem 1].

In the remainder of this work, for our application to the contingency screening problem, we will consider ICNNs with input dimension n and output dimension 1. When using an ICNN to classify the feasibility of an injection $\mathbf{x}$, we will take its prediction to be $\sigma(f_{\text{ICNN}}(\mathbf{x}))$, where $\sigma(x) = (1 + e^{-x})^{-1}$ is a sigmoidal activation applied to the output of the ICNN. In this case, predictions less than 0.5 will correspond to a "feasible" classification (0), and those strictly greater than 0.5 will correspond to "infeasible" (1). With this setup, one readily observes that the predicted feasible region of an ICNN is exactly its 0-sublevel set:

$$\{\mathbf{x} \in \mathbb{R}^n : \sigma(f_{\text{ICNN}}(\mathbf{x})) \leq 0.5\} = \{\mathbf{x} \in \mathbb{R}^n : f_{\text{ICNN}}(\mathbf{x}) \leq 0\}.$$

Note that the universal convex function approximation property enjoyed by ICNNs implies that *any* convex set can be approximated arbitrarily well by the 0-sublevel set of an ICNN. Thus, ICNNs are well-matched to the task of approximating the true feasible region $\mathcal{F}_C$ for contingency screening, which is itself a convex set.

Following Definition 9.2.1, a reliable ICNN classifier is one whose predicted feasible region is contained inside the true feasible region $\mathcal{F}_C$. In the next section, we will discuss how the convex structure of an ICNN enables both (a) tractably determining whether this containment property holds and (b) scaling an ICNN's parameters to guarantee its reliability.

9.3 Certifying and Enforcing Reliability for ICNN Contingency Classifiers

As discussed in Section 9.2, a *reliable* classifier for the contingency screening problem is one that makes no false negative predictions, i.e., whose predicted feasible region is fully contained inside the true feasible region $\mathcal{F}_C$ (9.2). For an ICNN classifier f_{ICNN}, this amounts to the property that its 0-sublevel set is contained in $\mathcal{F}_C$. An immediate question that arises is whether it is possible to certify that a given classifier f_{ICNN} satisfies this reliability criterion. Conveniently, we can show that certifying this property reduces to solving a collection of convex optimization problems.

Proposition 9.3.1. *An ICNN classifier for the contingency screening problem is reliable—i.e., has zero false negative rate—if and only if*

$$\left\{ \begin{array}{ll} \max\limits_{\mathbf{x}\in\mathbb{R}^n} & \mathbf{a}_j^\top \mathbf{x} \\ \text{s.t.} & f_{\text{ICNN}}(\mathbf{x}) \leq 0 \end{array} \right\} \leq b_j \tag{9.5}$$

for all $j \in [2m|C|]$, where $\mathbf{a}_j$ is the jth row of $\mathbf{A}$.

Proof. We first observe that, since f_{ICNN} is a convex function, the optimization problem in (9.5) is a convex problem, and thus can be solved tractably. Given this convexity, the fact that containment of the 0-sublevel set of f_{ICNN} inside the polyhedron $\mathcal{F}_C$ can be determined by solving a collection of convex optimization problems of the form (9.5) is a standard result in convex optimization (see, e.g., [346]). For the sake of completeness, we briefly describe the proof here.

For the forward direction, suppose that containment holds, i.e., $\{\mathbf{x} \in \mathbb{R}^n : f_{\text{ICNN}}(\mathbf{x}) \leq 0\} \subseteq \mathcal{F}_C$. This means that $f_{\text{ICNN}}(\mathbf{x}) \leq 0$ implies $\mathbf{A}\mathbf{x} \leq \mathbf{b}$; thus any feasible solution $\mathbf{x}$

to the problem in (9.5) will satisfy the inequality $\mathbf{a}_j^\top \mathbf{x} \leq b_j$, and hence this inequality will hold at optimality.

For the reverse direction, suppose that (9.5) holds for all j. If there were some $\mathbf{x} \in \mathbb{R}^n$ which was not feasible ($\mathbf{x} \notin \mathcal{F}_C$) and yet was predicted feasible by the ICNN ($f_{\text{ICNN}}(\mathbf{x}) \leq 0$), this would imply the existence of some j such that $\mathbf{a}_j^\top \mathbf{x} > b_j$, yielding a contradiction. □

The previous proposition provides a way of tractably certifying whether a given ICNN classifier is reliable, but it does not give a means of transforming an unreliable classifier into a reliable one. Since reliability of a classifier is exactly containment of its predicted feasible set inside the true feasible set, a natural approach to enforcing reliability would be to transform the classifier to translate and scale its predicted feasible set into the interior of $\mathcal{F}_C$. In general, the problem of scaling a convex set $\mathcal{A}$ to be contained in another convex set $\mathcal{B}$ can be tractably cast as a convex optimization problem in certain special cases, such as when both $\mathcal{A}$ and $\mathcal{B}$ are polyhedra given in halfspace description (see the foundational work of Eaves and Freund [346]). However, the set we are concerned with scaling is the 0-sublevel set of an ICNN, which has not been considered in prior work on set containment, and which is more complex due to the multilayer structure and substantial representational efficiency of ICNNs [278, Theorem 2].

Nonetheless, as we show in the following theorem, it is possible to perform such a scaling efficiently by solving a collection of convex optimization problems, yielding a reliable classifier.

Theorem 9.3.2. *Let r^* and $\mathbf{v}^*$ be the optimal solutions to the optimization problem*

$$\min_{r \in \mathbb{R}_+, \mathbf{v} \in \mathbb{R}^n} \quad r \tag{9.6a}$$

$$\text{s.t.} \quad z_j^* \leq \mathbf{a}_j^\top \mathbf{v} + b_j r \quad \forall j \in [2m|C|]\,, \tag{9.6b}$$

where

$$\begin{aligned} z_j^* := \max_{\mathbf{x} \in \mathbb{R}^n} \quad & \mathbf{a}_j^\top \mathbf{x} \\ \text{s.t.} \quad & f_{\text{ICNN}}(\mathbf{x}) \leq 0 \end{aligned} \tag{9.7}$$

for each $j \in [2m|C|]$. Then the transformed ICNN classifier $\hat{f}_{\text{ICNN}}$ defined as

$$\hat{f}_{\text{ICNN}}(\mathbf{x}) := f_{\text{ICNN}}(r^* \mathbf{x} + \mathbf{v}^*)$$

has zero false negative rate. Moreover, (9.6) *has a feasible solution as long as the original predicted feasible set $\{\mathbf{x} \in \mathbb{R}^n : f_{\text{ICNN}}(\mathbf{x}) \leq 0\}$ is bounded.*

Before proving Theorem 9.3.2, we first make four brief comments. First, note that the boundedness assumption on the predicted feasible set $\{\mathbf{x} \in \mathbb{R}^n : f_{\mathrm{ICNN}}(\mathbf{x}) \leq 0\}$ can be easily enforced by, e.g., adding an indicator function to f_{ICNN} that is 0 for all $\|\mathbf{x}\| \leq D$ and $+\infty$ otherwise, where D is some large constant.

Second, note that the transformed classifier $\hat{f}_{\mathrm{ICNN}}$ can be obtained from f_{ICNN} (as defined in (9.4)) by multiplying the weights $\mathbf{D}_i$ by r^* and adding $\mathbf{D}_i\mathbf{v}^*$ to the biases. Its predicted feasible set is a transformed version of f_{ICNN}'s, obtained by translating by $-\mathbf{v}^*$ and scaling down by a factor of r^*. As long as r^* is not infinite—that is, if (9.6) is feasible—then the predicted feasible set of $\hat{f}_{\mathrm{ICNN}}$ will be nonempty (assuming that f_{ICNN}'s predicted feasible set is nonempty). We thus seek to minimize r to maximize the volume of $\hat{f}_{\mathrm{ICNN}}$'s predicted feasible set, which will ensure reliability while minimizing the conservativeness of this classifier as an inner approximation of the true feasible set $\mathcal{F}_C$.

Note that the resulting classifier might still be relatively conservative and suffer poor prediction accuracy on the negative class, i.e., a large false positive rate. In Section 9.4 we will propose a methodology to reduce this conservativeness and enforce classifier reliability *during training* by incorporating a version of the scaling problem (9.6) into the training process as a differentiable layer.

Third, observe that computing r^* and $\mathbf{v}^*$ requires solving a collection of $2m|C|$ optimization problems (9.7) followed by a linear program (9.6) with just as many constraints. One might question, thus, the benefit of our scaling approach over exhaustive checking of contingencies, which has a similar dependence on $|C|$ in its complexity. However, our approach has a substantial benefit: this scaling must only be performed *once* to obtain a classifier that is provably reliable for *any* net injection, and all subsequent feasibility predictions only require an efficient feedforward pass of the ICNN. In contrast, exhaustively checking contingencies must be done separately for *every* net injection. Thus, our approach yields significantly improved efficiency at deployment time by moving the computational burden of ensuring reliability from the *online*, real-time setting to an *offline* preprocessing step.

Finally, we note that it is possible to transform the problems (9.6), (9.7) into a single linear program by taking the Lagrange dual of each maximization problem (9.7) [347] (see Chapter 7.C), similar to the approach for polyhedra in [346]. However, our multi-problem formulation is more efficient, as it lends itself to a distributed solution approach where we solve each of the smaller, independent optimization problems (9.7) in parallel before using their optimal solutions to solve the linear program (9.6).

We now present a proof of Theorem 9.3.2.

Proof of Theorem 9.3.2. Consider the optimization problem

$$\max_{r\in\mathbb{R}_+,\mathbf{v}\in\mathbb{R}^n} \quad \mathrm{vol}\left(\{\mathbf{x}\in\mathbb{R}^n : f_{\mathrm{ICNN}}(r\mathbf{x}+\mathbf{v})\le 0\}\right) \tag{9.8a}$$

$$\text{s.t.} \quad \left\{\begin{aligned} &\max_{\mathbf{x}\in\mathbb{R}^n} \ \mathbf{a}_j^\top \mathbf{x} \\ &\ \text{s.t. } f_{\mathrm{ICNN}}(r\mathbf{x}+\mathbf{v})\le 0 \end{aligned}\right\} \le b_j \quad \forall j\in[2m|C|] \tag{9.8b}$$

where we seek to maximize the volume of the predicted feasible set of f_{ICNN} (to minimize conservativeness) after scaling and translating it by r and $\mathbf{v}$, subject to the constraint that this transformed set is contained in the true feasible set $\mathcal{F}_C$. First, note that since $\mathcal{F}_C$ has nonempty interior—and specifically, $\mathbf{0}\in\operatorname{int}\mathcal{F}_C$ (Assumption 9.1)—then if the original predicted feasible set $\{\mathbf{x}\in\mathbb{R}^n : f_{\mathrm{ICNN}}(\mathbf{x})\le 0\}$ is bounded, then (9.8) has a feasible solution. This is because there must be a ε-neighborhood about the origin that remains contained in $\mathcal{F}_C$; thus, since the predicted feasible region of f_{ICNN} is bounded, it is possible to choose a translation $\mathbf{v}$ and a sufficiently large (yet finite) r to ensure the transformed predicted region is contained in this ε-neighborhood.

Now, let us consider the objective (9.8a) and the constraints (9.8b) separately. We can assume that $r>0$, since $r=0$ would only be feasible if $\mathcal{F}_C$ were all of $\mathbb{R}^n$, which violates Assumption 9.1. For the objective, observe that

$$\begin{aligned} &\mathrm{vol}\left(\{\mathbf{x}\in\mathbb{R}^n : f_{\mathrm{ICNN}}(r\mathbf{x}+\mathbf{v})\le 0\}\right) \\ =\ &\mathrm{vol}\left(\{r^{-1}(\mathbf{y}-\mathbf{v})\in\mathbb{R}^n : f_{\mathrm{ICNN}}(\mathbf{y})\le 0, \mathbf{y}\in\mathbb{R}^n\}\right) \\ =\ &r^{-n}\cdot\mathrm{vol}\left(\{\mathbf{y}\in\mathbb{R}^n : f_{\mathrm{ICNN}}(\mathbf{y})\le 0\}\right), \end{aligned} \tag{9.9}$$

where the final equality follows from the fact that homogeneously scaling a body by s in n dimensions scales the volume by s^n, and translation has no impact on volume. Since the volume term in (9.9) is independent of the decision variables r and $\mathbf{v}$, and maximizing r^{-n} will yield the same optimal solution as minimizing r (since the function $s\mapsto s^{-1/n}$ is strictly decreasing on $s>0$), we can replace (9.8a) with $\min_{r\in\mathbb{R}_+,\mathbf{v}\in\mathbb{R}^n} r$ while keeping the same optimal solution. This exactly matches the objective in (9.6a).

Next, consider the constraints (9.8b). By Proposition 9.3.1, these constraints enforce the reliability—or zero false negative rate—of the transformed classifier $f_{\text{ICNN}}(r\mathbf{x}+\mathbf{v})$. For a given $j \in [2m|C|]$, since $r > 0$, we have

$$
\begin{aligned}
&\left\{\begin{array}{l} \max\limits_{\mathbf{x}\in\mathbb{R}^n} \ \mathbf{a}_j^\top \mathbf{x} \\ \ \text{s.t.} \ f_{\text{ICNN}}(r\mathbf{x}+\mathbf{v}) \le 0 \end{array}\right\} \le b_j \\
\iff &\left\{\begin{array}{l} \max\limits_{\mathbf{y}\in\mathbb{R}^n} \ \mathbf{a}_j^\top r^{-1}(\mathbf{y}-\mathbf{v}) \\ \ \text{s.t.} \ f_{\text{ICNN}}(\mathbf{y}) \le 0 \end{array}\right\} \le b_j \\
\iff &\left\{\begin{array}{l} \max\limits_{\mathbf{y}\in\mathbb{R}^n} \ \mathbf{a}_j^\top \mathbf{y} \\ \ \text{s.t.} \ f_{\text{ICNN}}(\mathbf{y}) \le 0 \end{array}\right\} \le \mathbf{a}_j^\top \mathbf{v} + b_j r
\end{aligned}
$$

which exactly matches (9.6b) and (9.7). □

9.4 Training Reliable ICNN Classifiers with Differentiable Convex Optimization Layers

Theorem 9.3.2 in the previous section provides an approach to scale the parameters of an existing ICNN classifier to guarantee provable reliability, or zero false negative rate. However, this post-hoc scaling process could yield significant conservativeness. This is because scaling down the predicted feasible region by a factor of $r > 1$ decreases its volume by a factor of r^n; under mild assumptions on the probability distribution over net injections $\mathbf{x} \in \mathbb{R}^n$ seen at deployment time, this scaling could beget an exponential increase in the false positive rate compared to the original, unreliable classifier.

To avoid this conservativeness, it is necessary to incorporate this scaling procedure into the training of the ICNN classifier, rather than applying it only after training. A natural approach is as follows: at each epoch of training, first solve the problems (9.6) and (9.7) to determine the optimal scaling parameters r^* and $\mathbf{v}^*$. Then, evaluate the training loss of the transformed ICNN classifier—for a single injection/label pair $(\mathbf{x}, y)$, we denote this loss $L\left(f_{\text{ICNN}}(r^*\mathbf{x}+\mathbf{v}^*), y\right)$, where L is some classification loss—and update the model f_{ICNN} using the gradient $\frac{\partial L}{\partial f_{\text{ICNN}}}$, where ∂f_{ICNN} refers to the gradient with respect to all the parameters of f_{ICNN}. This approach aligns the training loss with the objective of learning the optimal reliable classifier, since the loss that is minimized through gradient descent is that of the reliable, scaled version of the generic classifier f_{ICNN}.

However, this approach is incomplete. In particular, note that the scaling parameters $r^*, \mathbf{v}^*$ resulting from the problem (9.6) themselves depend on the parameters of f_{ICNN} through each z_j^*. Defining $\hat{y} := f_{\text{ICNN}}(r^*\mathbf{x} + \mathbf{v}^*)$, by the chain rule, the gradient of $L(\hat{y}, y)$ with respect to the parameters of f_{ICNN} is

$$\frac{\partial L}{\partial f_{\text{ICNN}}}(\hat{y}, y) = \frac{\partial L}{\partial \hat{y}}\left(\frac{\partial \hat{y}}{\partial f_{\text{ICNN}}} + \frac{\partial \hat{y}}{\partial r^*}\sum_j \frac{\partial r^*}{\partial z_j^*}\frac{\partial z_j^*}{\partial f_{\text{ICNN}}} + \frac{\partial \hat{y}}{\partial \mathbf{v}^*}\sum_j \frac{\partial \mathbf{v}^*}{\partial z_j^*}\frac{\partial z_j^*}{\partial f_{\text{ICNN}}}\right).$$

Thus to compute the gradient of the loss L with respect to the parameters of the ICNN f_{ICNN}, it is necessary to also compute the gradients of the optimal solutions $r^*, \mathbf{v}^*$ of (9.6) with respect to each z_j^*, and the gradient of each optimal value z_j^* of (9.7) with respect to f_{ICNN}'s parameters. To compute these gradients, we can employ differentiable convex optimization layers [290], which automatically compute the gradient of a convex optimization problem with respect to problem parameters by differentiating through the Karush-Kuhn-Tucker (KKT) conditions of the problem, allowing the incorporation of such problems into machine learning training methodologies in a fully differentiable manner. By computing r^* and $\mathbf{v}^*$ using differentiable layers, we ensure that the training process is "aware" of the scaling procedure that is applied to f_{ICNN} to guarantee reliability.

While this fully differentiable approach ensures that the scaling procedure is accounted for when computing the loss gradient, it requires computing both the solution to (9.6) and the solutions to (9.7) for all $j \in [2m|C|]$ using differentiable layers, which typically require additional computational overhead beyond a non-differentiable solution [290]. Because we need to apply this scaling at each epoch of training to enforce reliability, reducing the number of differentiable optimization layers used at each step of training would improve computational efficiency.

Fortunately, as we show in the following theorem, it is possible to obtain a fully differentiable scaling procedure using just a *single* differentiable optimization problem.

Theorem 9.4.1. *Let z_j^* be defined as in (9.7) for each $j \in [2m|C|]$, and let $j^* :=$ $\arg\max_j z_j^*/b_j$. Define r^* to be the optimal value of the following problem:*

$$\begin{aligned} r^* := \max_{\mathbf{x}\in\mathbb{R}^n} \quad & \mathbf{a}_{j^*}^\top \mathbf{x}/b_{j^*} \\ \text{s.t.} \quad & f_{\text{ICNN}}(\mathbf{x}) \leq 0. \end{aligned} \tag{9.10}$$

Then the transformed ICNN classifier $\hat{f}_{\text{ICNN}}$ defined as

$$\hat{f}_{\text{ICNN}}(\mathbf{x}) := f_{\text{ICNN}}(r^*\mathbf{x})$$

has zero false negative rate. Moreover, (9.10) *has a feasible solution as long as the original predicted feasible set* $\{\mathbf{x} \in \mathbb{R}^n : f_{\text{ICNN}}(\mathbf{x}) \leq 0\}$ *is bounded.*

Proof. Consider the optimization problem (9.6), and fix $\mathbf{v} = \mathbf{0}$; this problem remains feasible, by the assumption that the predicted feasible set is bounded, and since $\mathbf{0} \in \operatorname{int} \mathcal{F}_C$ (Assumption 9.1) implies that $\mathbf{b} > \mathbf{0}$. The optimal solution r^* to (9.6) is the smallest value of r that satisfies the constraints (9.6b); this is exactly

$$r^* := \max_j z_j^*/b_j.$$

It is straightforward to see that this r^* is identical to the one obtained by (9.10). Thus, the scaling obtained from (9.10) inherits the zero false negative rate yielded by (9.6). □

In Theorem 9.4.1, the values z_j^* only need to be computed in order to determine the maximizing index j^*; then, the scaling ratio r^* is computed using just the single optimization problem (9.10). As such, all of the z_j^* can be computed in parallel in a non-differentiable fashion, and only (9.10) must be solved using a differentiable layer. Note additionally that the lack of a translation variable $\mathbf{v}$ in (9.10) should not yield any additional conservativeness during training, since the ICNN can learn biases that would imitate the impact of any such possible $\mathbf{v}$.

We outline in Algorithm 13 a training methodology incorporating the fast, differentiable scaling procedure in Theorem 9.4.1. In this process, we begin by "warm-starting" the training for M_w epochs by performing standard gradient descent on the classification loss without scaling for reliability. Then, for each of the remaining M_s epochs, the model is scaled using a differentiable layer implementing (9.10) before evaluating the training loss. Note that after every gradient step, the ICNN's weights $\mathbf{W}_i$ must be clipped to the positive orthant to maintain convexity.

9.5 Experimental Results

In this section, we describe the results of our ICNN training methodology (Algorithm 13) in a case study of $N-2$ contingency screening on the IEEE 39-bus test network [348, 349]. All experiments were performed on a MacBook Pro with 12-core M3 Pro processor, and the code for implementing the experiments is available upon request.

We used the IEEE 39-bus test network implemented in pandapower [350]. We generated 14,000 random demand vectors from a multivariate normal distribution

Algorithm 13: Training procedure for reliable ICNN classifiers

Input: training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, initial ICNN f_{ICNN}, warm-start epochs M_w, scaling epochs M_s, batch size s

/* Warm-start the ICNN training without scaling */

for each epoch in $[M_w]$ **do**

  for each mini-batch $B \subset [N]$ **do**

    Evaluate the loss $\frac{1}{s}\sum_{i\in B} L\left(f_{\text{ICNN}}(\mathbf{x}_i), y_i\right)$

    Compute the gradient $\frac{\partial\,\text{loss}}{\partial f_{\text{ICNN}}}$ and use it to update f_{ICNN}

  end

end

/* Train with scaling to enforce reliability */

for each epoch in $[M_s]$ **do**

  Compute

$$\begin{aligned} z_j^* := \max_{\mathbf{x}\in\mathbb{R}^n} \quad & \mathbf{a}_j^\top \mathbf{x} \\ \text{s.t.} \quad & f_{\text{ICNN}}(\mathbf{x}) \le 0 \end{aligned}$$

  for each $j \in [2m|C|]$

  Set $j^* := \arg\max_j z_j^*/b_j$

  Compute

$$\begin{aligned} r^* := \max_{\mathbf{x}\in\mathbb{R}^n} \quad & \mathbf{a}_{j^*}^\top \mathbf{x}/b_{j^*} \\ \text{s.t.} \quad & f_{\text{ICNN}}(\mathbf{x}) \le 0 \end{aligned}$$

  using a differentiable convex optimization layer

  Evaluate the loss $\frac{1}{s}\sum_{i\in B} L\left(f_{\text{ICNN}}(r^*\mathbf{x}_i), y_i\right)$ of the scaled model on a mini-batch B

  Compute the gradient $\frac{\partial\,\text{loss}}{\partial f_{\text{ICNN}}}$ and use it to update f_{ICNN}

end

centered at the nominal demand with relative standard deviation 15% and random covariance. We assigned each generator a linear cost with random coefficient between 10 and 50, and set line limits uniformly to 1600 MW. We then solved the DC-OPF problem (9.1) for each demand instance to obtain net injections, which were then standardized and split into a 10,000 sample training set, a 2,000 sample validation set, and a 2,000 sample test set.

To construct the true feasible set $\mathcal{F}_C$, we took the set of all $N-2$ contingencies and dropped any islanding contingencies as well as contingencies that were infeasible more than 90% of the time, since these should be handled separately. We eliminated any dimensions for which the generated injection data was constant and eliminated redundant constraints using the method from [332, Theorem 2], using as a bounding box the empirical dimension-wise minimum and maximum net injections, multiplied by 1.2 for buffer and extended to include the origin. This resulted in a constraint matrix $\mathbf{A}$ with 3,613 rows and 26 columns. To account for the standardized training data, we multiplied the rows of $\mathbf{A}$ by $\boldsymbol{\sigma}$ and subtracted $\mathbf{A}\boldsymbol{\mu}$ from $\mathbf{b}$, where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the dimension-wise mean and standard deviation of the unstandardized training data.

We trained both ICNNs and standard, nonconvex neural networks (NNs) for the contingency screening task using PyTorch [351]. All networks had a hidden width of 50, we enforced boundedness of the predicted feasible set by adding a layer ensuring the output would always be positive outside of the aforementioned bounding box of net injections, and we trained models using hidden depths of 1, 2, and 3, as well as weights of 0.5, 1, and 1.5 on the positive class of the binary cross-entropy loss to probe the impact of positive class weight on false negative rate. For each choice of parameters, we trained 3 models with independent seeds, and in our results we report the mean and standard deviation of performance over these seeds. We trained the ICNNs using 500 warm-start epochs and 9,500 scaling epochs, and the nonconvex NNs were trained using 10,000 standard epochs. The cvxpylayers library [290] was used to differentiably solve the optimization problem in line 11 of the training methodology (Algorithm 13). We used the Adam optimizer [306] with learning rate 10^{-2}, decreasing the learning rate by a factor of 10 at epoch 1,500 and again at 8,500. During each training run, we kept track of the false positive rate on the validation set at each epoch and selected as the training output the model with the best such validation set performance.

We show in Figure 9.2 a 2-dimensional slice of the true feasible region $\mathcal{F}_C$ and the predicted feasible region of a 1-layer ICNN trained via our methodology. It is evident that the ICNN respects the inner approximation property as a result of the scaling procedure while learning to focus on the data-intensive region at the bottom of the true feasible region. The ICNN does not need to learn the shape of the entire true feasible region due to data sparsity at the top of this slice, enabling a more efficient representation.

Contingency Screening Results

We show in Figure 9.3 a comparison of ICNNs with several hidden depths trained via our methodology against standard NNs and the exhaustive method of checking all constraints individually for the contingency screening problem. Note that the "Positive Weight" value refers to the weight assigned to elements of the positive

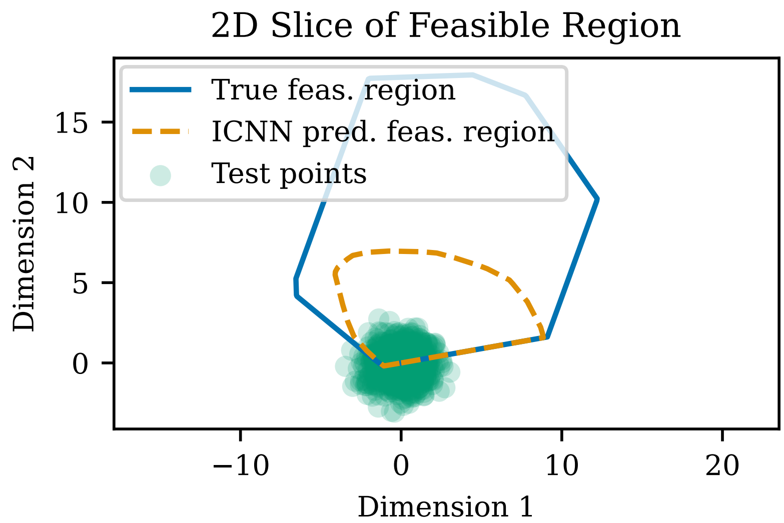


Figure 9.2: 2-dimensional slice of the true feasible region and the predicted feasible region of a trained ICNN with hidden depth 1, with net injections from the test set overlaid.

class in the training loss, where weights less than 1 typically encourage lower false positive rates, and weights greater than 1 typically encourage lower false negative rates.

Notably, the ICNNs trained with our differentiable scaling procedure in Algorithm 13 achieve a speedup of 10-20$\times$ over the exhaustive method, depending on the depth of the ICNN (Figure 9.3, top). Moreover, they uniformly achieve a false negative rate of 0 (Figure 9.3, middle), as guaranteed by our theoretical results, and a false positive rate between 2% and 5% (Figure 9.3, bottom). While the effect is not significant, it appears in the cases of hidden depth 1 and 3 that a lower positive weight may decrease the false positive rate of our approach, though further study is needed to understand whether this behavior holds in general.

In comparison, the nonconvex NNs achieve a better false positive rate, ranging between 0.5% and 1%, but suffer significant false negative rates of 1% to 3%, demonstrating that they cannot reliably be used for contingency screening, as they could misclassify infeasible scenarios as feasible. Our approach thus enables significantly faster screening than the exhaustive method while ensuring the reliability that cannot be guaranteed by standard NNs.

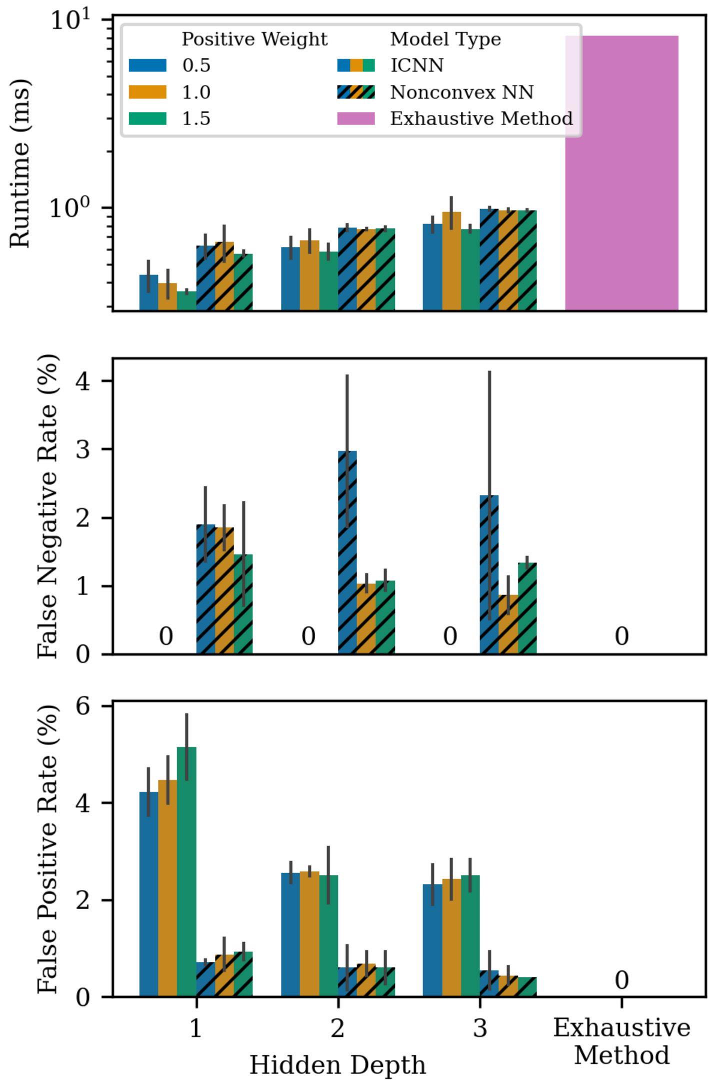


Figure 9.3: Results for our ICNN-based contingency analysis method, compared against a nonconvex neural network (NN) model and exhaustive checking of contingencies. (Top) Runtime to screen the feasibility of the 2,000 test injections. (Middle) False negative rate. (Bottom) False positive rate.

Faster Preventive Dispatch via SC-OPF

In practice, power system operators often want to perform *preventive* dispatch to ensure that the chosen operating point will remain feasible in the case of contingencies. This problem, known as security-constrained (SC)-DC-OPF, adds to (9.1) the additional constraint that $\mathbf{x} := \mathbf{p} - \mathbf{d}$ should be feasible for all contingencies in the reference set C—that is, $\mathbf{p} - \mathbf{d} \in \mathcal{F}_C$:

$$
\begin{aligned}
\min_{\mathbf{p}\in\mathbb{R}^n} \quad & \sum_{i\in[n]} c_i(p_i) && (9.11a)\\
\text{s.t.} \quad & \underline{\mathbf{p}} \le \mathbf{p} \le \overline{\mathbf{p}} && (9.11b)\\
& \mathbf{1}^\top(\mathbf{p} - \mathbf{d}) = 0 && (9.11c)\\
& \underline{\mathbf{f}} \le \mathbf{H}(\mathbf{p} - \mathbf{d}) \le \overline{\mathbf{f}} && (9.11d)\\
& \mathbf{p} - \mathbf{d} \in \mathcal{F}_C. && (9.11e)
\end{aligned}
$$

Because our ICNN approach to contingency screening yields an ICNN $f_{\text{ICNN}}(r^*\cdot)$ whose 0-sublevel set is an inner approximation to $\mathcal{F}_C$, one might naturally consider replacing the security constraint (9.11e) in the full SC-OPF problem with the conservative inner approximation $\hat{f}_{\text{ICNN}}(r^*(\mathbf{p} - \mathbf{d})) \le 0$ in an attempt to accelerate the solution time of this problem, since the original set $\mathcal{F}_C$ is typically high-dimensional. We test the performance of this approach and its impact on system cost and infeasibility using our ICNN models trained on the IEEE 39-bus system, and we display the results in Figure 9.4.

We see that, while the ICNNs with hidden depth 3 do not offer a speedup compared to solving (9.11) exactly, the 2-layer ICNNs halve the runtime, and the shallowest 1-layer ICNNs speed up this problem by nearly a factor of 10 (Figure 9.4, top). Remarkably, they achieve this speedup while increasing the dispatch cost by no more than 0.1% on average over the full SC-OPF problem (Figure 9.4, middle), and increasing the share of infeasible demand instances by only ~1%. It also appears that, for the ICNNs with hidden depth 1, decreasing the positive weight leads to better cost and less infeasibility. This agrees with intuition, since a lower positive weight encourages lower false positive rates, meaning that the ICNN should be a less conservative inner approximation to the set $\mathcal{F}_C$. However, further study will be needed to determine whether this observation generalizes, as the trend falls within the error bars and our deeper models do not seem to exhibit this behavior.

To conclude, note that we could modify our training methodology in Algorithm 13 by replacing the classification loss with a differentiable convex optimization layer

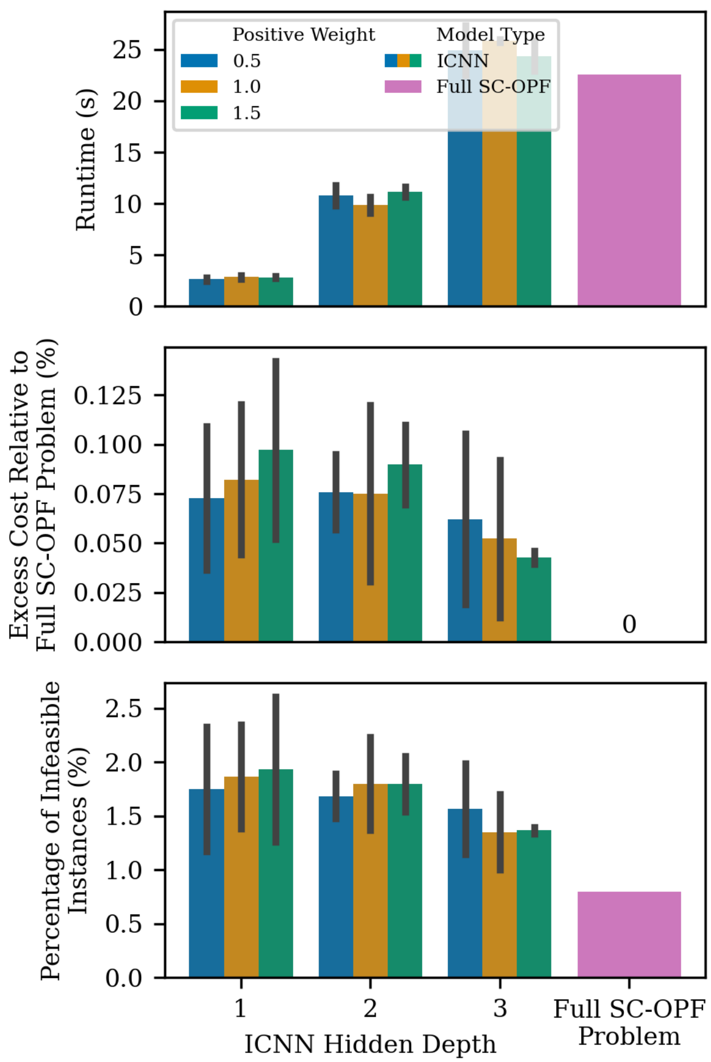


Figure 9.4: Results for the ICNN-based SC-OPF problem compared to the full SC-OPF problem (9.11). (Top) Runtime to solve the SC-OPF problem or ICNN version thereof on 2,000 test injections, disregarding infeasible injections. (Middle) Percent excess cost of the ICNN version of SC-OPF relative to the full SC-OPF problem (9.11). (Bottom) Percentage of infeasible demand instances for the ICNN version of SC-OPF compared against the full SC-OPF problem (9.11).

encoding the SC-OPF problem with ICNN security constraint. This would likely improve the performance of the ICNN for SC-OPF, since training the model end-to-end in such a manner would align training with the eventual downstream task faced by the model. We leave an implementation and evaluation of this change to future work.

9.6 Discussion and Conclusions

In this chapter, we propose a methodology for data-driven training of input-convex neural network classifiers for contingency screening in power systems with zero false negative rate. We show that certifying and enforcing zero false negative rate—i.e., reliability—of an ICNN classifier can be achieved by solving a collection of optimization problems, and by incorporating these problems into a differentiable convex optimization layer during ICNN training, we can restrict training to be over the set of provably reliable models. We evaluate the performance of our approach on contingency screening and preventive dispatch on the IEEE 39-bus test system, showing that it achieves good performance, guaranteed reliability, and a significant computational speedup over conventional methods. We anticipate that the computational benefit of our approach will be even more significant for larger-scale power systems and higher-order contingency screening problems.

A number of interesting avenues remain open for future work, including (a) scaling up this approach to enable application to larger-scale power systems; (b) combining this screening approach with, e.g., methods from group testing to achieve comparable speedups for the full contingency analysis problem; and (c) extending this methodology to other applications that require constructing tractable inner approximations to some complicated set, such as learning data-driven and safe inner approximations to AC-OPF feasible regions or electric vehicle aggregate flexibility sets. In addition, given the similarities between this methodology and that proposed in Chapter 7 for learning calibrated uncertainty sets, it would be interesting to explore whether this broader conceptual framework can be extended to enable training machine learning models while enforcing more general notions of reliability.

BIBLIOGRAPHY

[1] “Global Electricity Review 2025,” Ember, [Online]. Available: https://ember-energy.org/latest-insights/global-electricity-review-2025.

[2] S. Vahedi, J. Zhao, B. Pierre, F. Lei, E. Anagnostou, K. He, C. Jones, and B. Wang, “Wildfire and Power Grid Nexus in a Changing Climate,” *Nature Reviews Electrical Engineering*, vol. 2, no. 4, pp. 225–243, Apr. 2025, ISSN: 2948-1201. DOI: 10.1038/s44287-025-00150-0. [Online]. Available: https://www.nature.com/articles/s44287-025-00150-0.

[3] Y. Wang, C. Chen, J. Wang, and R. Baldick, “Research on Resilience of Power Systems Under Natural Disasters—A Review,” *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1604–1613, Mar. 2016, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2015.2429656. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7105972.

[4] M. Abdel-Aty and S. Ding, “A Matched Case-Control Analysis of Autonomous vs Human-Driven Vehicle Accidents,” *Nature Communications*, vol. 15, no. 1, p. 4931, Jun. 2024, ISSN: 2041-1723. DOI: 10.1038/s41467-024-48526-4. [Online]. Available: https://www.nature.com/articles/s41467-024-48526-4.

[5] D. Silver et al., “Mastering the Game of Go with Deep Neural Networks and Tree Search,” *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016, ISSN: 1476-4687. DOI: 10.1038/nature16961. [Online]. Available: https://www.nature.com/articles/nature16961.

[6] D. Silver et al., “A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go through Self-Play,” *Science*, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018. DOI: 10.1126/science.aar6404. [Online]. Available: https://www.science.org/doi/10.1126/science.aar6404.

[7] Meta Fundamental AI Research Diplomacy Team (FAIR) et al., “Human-Level Play in the Game of Diplomacy by Combining Language Models with Strategic Reasoning,” *Science*, vol. 378, no. 6624, pp. 1067–1074, Dec. 2022. DOI: 10.1126/science.ade9097. [Online]. Available: https://www.science.org/doi/10.1126/science.ade9097.

[8] Y. Gao and GitHub Customer Research. “Research: Quantifying GitHub Copilot’s Impact in the Enterprise with Accenture,” The GitHub Blog, [Online]. Available: https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-in-the-enterprise-with-accenture/.

[9] "Meta's New AI System to Help Tackle Harmful Content," Meta, [Online]. Available: https://about.fb.com/news/2021/12/metas-new-ai-system-tackles-harmful-content/.

[10] "DeepMind AI Reduces Google Data Centre Cooling Bill by 40%," Google DeepMind, [Online]. Available: https://deepmind.google/discover/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-by-40/.

[11] "Energy Vault Announces FID Approval for 57 MW Cross Trails Battery Energy Storage System in Texas and 10-Year Offtake Agreement with Gridmatic," [Online]. Available: https://www.businesswire.com/news/home/20241108238574/en/Energy-Vault-Announces-FID-Approval-for-57-MW-Cross-Trails-Battery-Energy-Storage-System-in-Texas-and-10-Year-Offtake-Agreement-with-Gridmatic.

[12] "Artificial Intelligence – Exploring Its Use in Grid Modernization | California ISO," [Online]. Available: https://www.caiso.com/about/news/energy-matters-blog/artificial-intelligence-exploring-its-use-in-grid-modernization.

[13] "2025 Enterprise Project Candidates," [Online]. Available: https://www.nyiso.com/documents/20142/45575726/03a%20BPWG%202024-07-03%20Enterprise%20Project%20Descriptions.pdf/dc99ac8a-53b8-e9e1-0850-1ecaee3c662f.

[14] "PJM, Google & Tapestry Join Forces To Apply AI To Enhance Regional Planning, Generation Interconnection | PJM Inside Lines," [Online]. Available: https://insidelines.pjm.com/pjm-google-tapestry-join-forces-to-apply-ai-to-enhance-regional-planning-generation-interconnection/.

[15] "Tapestry and Coordinador Electrico Nacional (CEN)," [Online]. Available: https://x.company/case-study/tapestry-cen/.

[16] R. S. S. Kumar, D. O'Brien, K. Albert, S. Viljöen, and J. Snover. "Failure Modes in Machine Learning Systems." arXiv: 1911.11034 [cs], [Online]. Available: http://arxiv.org/abs/1911.11034.

[17] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Advances in Adversarial Attacks and Defenses in Computer Vision: A Survey," *IEEE Access*, vol. 9, pp. 155 161–155 196, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3127960. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9614158.

[18] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. MIT Press, Jun. 2022, 246 pp., ISBN: 978-0-262-54587-7. Google Books: MBZuEAAAQBAJ.

[19] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. S. Yu, "Generalizing to Unseen Domains: A Survey on Domain Generalization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 8052–8072, Aug. 2023, ISSN: 1558-2191. DOI: 10.1109/TKDE.2022.3178128. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9782500.

[20] L. Huang et al., "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–55, Mar. 2025, ISSN: 1046-8188, 1558-2868. DOI: 10.1145/3703155. arXiv: 2311.05232 [cs]. [Online]. Available: http://arxiv.org/abs/2311.05232.

[21] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner, "United States Data Center Energy Usage Report," LBNL–1005775, 1372902, 2024. DOI: 10.2172/1372902. [Online]. Available: http://www.osti.gov/servlets/purl/1372902/.

[22] "Energy and AI – Analysis," IEA, [Online]. Available: https://www.iea.org/reports/energy-and-ai.

[23] P. Li, J. Yang, M. A. Islam, and S. Ren. "Making AI Less "Thirsty": Uncovering and Addressing the Secret Water Footprint of AI Models." arXiv: 2304.03271 [cs], [Online]. Available: http://arxiv.org/abs/2304.03271.

[24] Y. Han, Z. Wu, P. Li, A. Wierman, and S. Ren. "The Unpaid Toll: Quantifying the Public Health Impact of AI." arXiv: 2412.06288 [cs], [Online]. Available: http://arxiv.org/abs/2412.06288.

[25] D. Bienstock and A. Verma, "Strong NP-hardness of AC Power Flows Feasibility," *Operations Research Letters*, vol. 47, no. 6, pp. 494–501, Nov. 2019, ISSN: 0167-6377. DOI: 10.1016/j.orl.2019.08.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167637719302470.

[26] F. Capitanescu, J. L. Martinez Ramos, P. Panciatici, D. Kirschen, A. Marano Marcolini, L. Platbrood, and L. Wehenkel, "State-of-the-Art, Challenges, and Future Trends in Security Constrained Optimal Power Flow," *Electric Power Systems Research*, vol. 81, no. 8, pp. 1731–1741, Aug. 2011, ISSN: 0378-7796. DOI: 10.1016/j.epsr.2011.04.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779611000885.

[27] D. Bienstock, *Electrical Transmission System Cascades and Vulnerability* (MOS-SIAM Series on Optimization). Society for Industrial and Applied Mathematics, Dec. 2015, ISBN: 978-1-61197-415-7. DOI: 10.1137/1.

9781611974164. [Online]. Available: https://epubs.siam.org/doi/book/10.1137/1.9781611974164.

[28] A. Svoboda, C.-L. Tseng, C.-A. Li, and R. Johnson, "Short-Term Resource Scheduling with Ramp Constraints [Power Generation Scheduling]," *IEEE Transactions on Power Systems*, vol. 12, no. 1, pp. 77–83, Feb. 1997, ISSN: 1558-0679. DOI: 10.1109/59.574926. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/574926.

[29] N. Christianson, L. Werner, A. Wierman, and S. Low, "Dispatch-Aware Planning for Feasible Power System Operation," *Electric Power Systems Research*, vol. 212, p. 108 597, Nov. 2022, ISSN: 0378-7796. DOI: 10.1016/j.epsr.2022.108597. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779622006733.

[30] M. Badiei, N. Li, and A. Wierman, "Online Convex Optimization with Ramp Constraints," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Osaka: IEEE, Dec. 2015, pp. 6730–6736, ISBN: 978-1-4799-7886-1. DOI: 10.1109/CDC.2015.7403279. [Online]. Available: http://ieeexplore.ieee.org/document/7403279/.

[31] C. Chen, L. Tong, and Y. Guo, "Pricing Energy Storage in Real-Time Market," in *2021 IEEE Power & Energy Society General Meeting (PESGM)*, ISSN: 1944-9933, Jul. 2021, pp. 1–5. DOI: 10.1109/PESGM46819.2021.9638013.

[32] E. Bitar and Y. Xu, "Deadline Differentiated Pricing of Deferrable Electric Loads," *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 13–25, Jan. 2017, ISSN: 1949-3061. DOI: 10.1109/TSG.2016.2601914. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7548363.

[33] P. Wiesner, I. Behnke, D. Scheinert, K. Gontarska, and L. Thamsen, "Let's Wait Awhile: How Temporal Workload Shifting Can Reduce Carbon Emissions in the Cloud," in *Proceedings of the 22nd International Middleware Conference*, 2021, pp. 260–272. DOI: 10.1145/3464298.3493399.

[34] L. Werner, A. Wierman, and S. H. Low, "Pricing Flexibility of Shiftable Demand in Electricity Markets," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, ser. E-Energy '21, New York, NY, USA: Association for Computing Machinery, Jun. 2021, pp. 1–14, ISBN: 978-1-4503-8333-2. DOI: 10.1145/3447555.3464847. [Online]. Available: https://dl.acm.org/doi/10.1145/3447555.3464847.

[35] N. Troy, E. Denny, and M. O'Malley, "Base-Load Cycling on a System With Significant Wind Penetration," *IEEE Transactions on Power Systems*, vol. 25, no. 2, pp. 1088–1097, May 2010, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2009.2037326.

[36] W.-S. Tan, M. P. Abdullah, and M. Shaaban, “A Day-Ahead Generation Scheduling with Energy Storage Considering Cycling Ramp Costs,” in *2016 IEEE International Conference on Power and Energy (PECon)*, Nov. 2016, pp. 95–100. DOI: 10.1109/PECON.2016.7951540.

[37] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, “Dynamic Right-Sizing for Power-Proportional Data Centers,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1378–1391, Oct. 2013, ISSN: 1558-2566. DOI: 10.1109/TNET.2012.2226216.

[38] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew, “Online Algorithms for Geographical Load Balancing,” in *2012 International Green Computing Conference (IGCC)*, IEEE, Jun. 2012. DOI: 10.1109/igcc.2012.6322266. [Online]. Available: https://doi.org/10.1109/igcc.2012.6322266.

[39] P. D. Hines, I. Dobson, E. Cotilla-Sanchez, and M. Eppstein, “"Dual Graph" and "Random Chemistry" Methods for Cascading Failure Analysis,” in *2013 46th Hawaii International Conference on System Sciences*, Jan. 2013, pp. 2141–2150. DOI: 10.1109/HICSS.2013.1.

[40] L. Gan and S. H. Low. “Convex Relaxations and Linear Approximation for Optimal Power Flow in Multiphase Radial Networks.” arXiv: 1406.3054 [math], [Online]. Available: http://arxiv.org/abs/1406.3054.

[41] J. Warrington, P. J. Goulart, S. Mariéthoz, and M. Morari, “Robust Reserve Operation in Power Systems Using Affine Policies,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, ISSN: 0743-1546, Dec. 2012, pp. 1111–1117. DOI: 10.1109/CDC.2012.6425913.

[42] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng, “Adaptive Robust Optimization for the Security Constrained Unit Commitment Problem,” *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 52–63, Feb. 2013, Conference Name: IEEE Transactions on Power Systems, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2012.2205021.

[43] A. Lorca, X. A. Sun, E. Litvinov, and Tongxin Zheng, “Multistage Adaptive Robust Optimization for the Unit Commitment Problem,” *Operations Research*, vol. 64, no. 1, pp. 32–51, Feb. 2016, ISSN: 0030364X. DOI: 10.1287/opre.2015.1456.

[44] T. Tapia, Á. Lorca, D. Olivares, M. Negrete-Pincetic, and A. J. Lamadrid L, “A Robust Decision-Support Method Based on Optimization and Simulation for Wildfire Resilience in Highly Renewable Power Systems,” *European Journal of Operational Research*, vol. 294, no. 2, pp. 723–733, Oct. 2021, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2021.02.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221721000862.

[45] B. K. Poolla, A. R. Hota, S. Bolognani, D. S. Callaway, and A. Cherukuri, "Wasserstein Distributionally Robust Look-Ahead Economic Dispatch," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 2010–2022, May 2021, Conference Name: IEEE Transactions on Power Systems, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2020.3034488. [Online]. Available: https://ieeexplore.ieee.org/document/9242289.

[46] Y. Dvorkin, "A Chance-Constrained Stochastic Electricity Market," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2993–3003, Jul. 2020, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2019.2961231. [Online]. Available: https://ieeexplore.ieee.org/document/8937824.

[47] R. Mieth, J. Kim, and Y. Dvorkin, "Risk- and Variance-Aware Electricity Pricing," *Electric Power Systems Research*, vol. 189, p. 106 804, Dec. 2020, ISSN: 0378-7796. DOI: 10.1016/j.epsr.2020.106804. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779620306076.

[48] M. Ndrio, A. N. Madavan, and S. Bose, "Pricing Conditional Value at Risk-Sensitive Economic Dispatch," in *2021 IEEE Power & Energy Society General Meeting (PESGM)*, ISSN: 1944-9933, Jul. 2021, pp. 01–05. DOI: 10.1109/PESGM46819.2021.9637845.

[49] A. N. Madavan, N. Dahlin, S. Bose, and L. Tong, "Risk-Based Hosting Capacity Analysis in Distribution Systems," *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 355–365, Jan. 2024, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2023.3238846.

[50] D. Maji, P. Shenoy, and R. K. Sitaraman, "CarbonCast: Multi-Day Forecasting of Grid Carbon Intensity," in *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '22, Boston, Massachusetts: Association for Computing Machinery, 2022, pp. 198–207, ISBN: 9781450398909. DOI: 10.1145/3563357.3564079. [Online]. Available: https://doi.org/10.1145/3563357.3564079.

[51] K. Leerbeck, P. Bacher, R. G. Junker, G. Goranović, O. Corradi, R. Ebrahimy, A. Tveit, and H. Madsen, "Short-Term Forecasting of CO2 Emission Intensity in Power Grids by Machine Learning," *Applied Energy*, vol. 277, p. 115 527, Nov. 2020, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2020.115527. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261920310394.

[52] C. Crozier, K. Baker, Y. Du, J. Mohammadi, and M. Li, "Data-driven Contingency Selection for Fast Security Constrained Optimal Power Flow," in *2022 17th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, ISSN: 2642-6757, Jun. 2022, pp. 1–6. DOI: 10.1109/PMAPS53380.2022.9810574.

[53] A. M. Nakiganda, C. Cheylan, and S. Chatzivasileiadis. "Topology-Aware Neural Networks for Fast Contingency Analysis of Power Systems." arXiv:2310.04213 [cs, eess], [Online]. Available: http://arxiv.org/abs/2310.04213.

[54] P. Donti, A. Agarwal, N. V. Bedmutha, L. Pileggi, and J. Z. Kolter, "Adversarially Robust Learning for Security-Constrained Optimal Power Flow," in *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 28 677–28 689. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/f0f07e680de407b0f12abf15bd520097-Abstract.html.

[55] L. Zhang, Y. Chen, and B. Zhang, "A Convex Neural Network Solver for DCOPF With Generalization Guarantees," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 2, pp. 719–730, Jun. 2022, ISSN: 2325-5870, 2372-2533. DOI: 10.1109/TCNS.2021.3124283. [Online]. Available: https://ieeexplore.ieee.org/document/9599403/.

[56] L. Zhang, D. Tabas, and B. Zhang. "An Efficient Learning-Based Solver for Two-Stage DC Optimal Power Flow with Feasibility Guarantees," [Online]. Available: http://arxiv.org/abs/2304.01409.

[57] M. Zhou, M. Chen, and S. H. Low, "DeepOPF-FT: One Deep Neural Network for Multiple AC-OPF Problems with Flexible Topology," *IEEE Transactions on Power Systems*, vol. 38, no. 1, pp. 964–967, Jan. 2023, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2022.3217407. [Online]. Available: https://ieeexplore.ieee.org/document/9940481/.

[58] Y. Chen, Y. Tan, and B. Zhang, "Exploiting Vulnerabilities of Load Forecasting Through Adversarial Attacks," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, ser. E-Energy '19, New York, NY, USA: Association for Computing Machinery, Jun. 15, 2019, pp. 1–11, ISBN: 978-1-4503-6671-7. DOI: 10.1145/3307772.3328314. [Online]. Available: https://dl.acm.org/doi/10.1145/3307772.3328314.

[59] C. Yeh et al., "SustainGym: Reinforcement Learning Environments for Sustainable Energy Systems," in *Thirty-Seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, New Orleans, LA, USA, Dec. 2023. [Online]. Available: https://openreview.net/forum?id=vZ9tA3o3hr.

[60] G. Buster, P. Pinchuk, J. Barrons, R. McKeever, A. Levine, and A. Lopez, "Supporting Energy Policy Research with Large Language Models: A Case Study in Wind Energy Siting Ordinances," *Energy and AI*, vol. 18, p. 100 431, Dec. 2024, ISSN: 2666-5468. DOI: 10.1016/j.egyai.2024.100431. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666546824000971.

[61] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial Examples Are Not Bugs, They Are Features," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://papers.nips.cc/paper_files/paper/2019/hash/e2c420d928d4bf8ce0ff2ec19b371514-Abstract.html.

[62] N. Loo, R. Hasani, A. Amini, and D. Rus, "Evolution of Neural Tangent Kernels under Benign and Adversarial Training," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22, Red Hook, NY, USA: Curran Associates Inc., Nov. 28, 2022, pp. 11 642–11 657, ISBN: 978-1-71387-108-8.

[63] I. J. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and Harnessing Adversarial Examples." arXiv: 1412.6572 [stat], [Online]. Available: http://arxiv.org/abs/1412.6572.

[64] E. Wong, F. Schmidt, J. H. Metzen, and J. Z. Kolter, "Scaling Provable Adversarial Defenses," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/hash/358f9e7be09177c17d0d17ff73584307-Abstract.html.

[65] K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar. "Regularized Learning for Domain Adaptation under Label Shifts." arXiv: 1903.09734 [cs], [Online]. Available: http://arxiv.org/abs/1903.09734.

[66] R. R. Bunel, I. Turkaslan, P. Torr, P. Kohli, and P. K. Mudigonda, "A Unified View of Piecewise Linear Neural Network Verification," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/hash/be53d253d6bc3258a8160556dda3e9b2-Abstract.html.

[67] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C.-J. Hsieh, and J. Z. Kolter, "Beta-CROWN: Efficient Bound Propagation with Per-neuron Split Constraints for Neural Network Robustness Verification," in *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 29 909–29 921. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/fac7fead96dafceaf80c1daffeae82a4-Abstract.html.

[68] A. Venzke and S. Chatzivasileiadis, "Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 383–397, Jan. 2021, ISSN: 1949-3061. DOI: 10.1109/TSG.2020.3009401. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9141308.

[69] T. Lykouris and S. Vassilvtiskii, "Competitive caching with machine learned advice," in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, PMLR, Jul. 2018, pp. 3296–3305. [Online]. Available: https://proceedings.mlr.press/v80/lykouris18a.html.

[70] M. Purohit, Z. Svitkina, and R. Kumar, "Improving Online Algorithms via ML Predictions," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/73a427badebe0e32caa2e1fc7530b7f3-Abstract.html.

[71] R. Lee, J. Maghakian, M. Hajiesmaili, J. Li, R. Sitaraman, and Z. Liu, "Online Peak-Aware Energy Scheduling with Untrusted Advice," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, Virtual Event Italy: ACM, Jun. 2021, pp. 107–123, ISBN: 978-1-4503-8333-2. DOI: 10.1145/3447555.3464860. [Online]. Available: https://dl.acm.org/doi/10.1145/3447555.3464860.

[72] E. Balkanski, V. Gkatzelis, and G. Shahkarami, "Randomized Strategic Facility Location with Predictions," *Advances in Neural Information Processing Systems*, vol. 37, pp. 35 639–35 664, Dec. 16, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/hash/3ec7806669b4048cdba4d1defc76ace3-Abstract-Conference.html.

[73] M. Dinitz, S. Im, T. Lavastida, B. Moseley, and S. Vassilvitskii, "Faster Matchings via Learned Duals," in *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 10 393–10 406. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/5616060fb8ae85d93f334e7267307664-Abstract.html.

[74] T. Li. "Learning-Augmented Scheduling for Solar-Powered Electric Vehicle Charging." arXiv:2311.05941 [cs, eess], [Online]. Available: http://arxiv.org/abs/2311.05941.

[75] A. Antoniadis, C. Coester, M. Elias, A. Polak, and B. Simon, "Online Metric Algorithms with Untrusted Predictions," in *Proceedings of the 37th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, Nov. 2020, pp. 345–355. [Online]. Available: https://proceedings.mlr.press/v119/antoniadis20a.html.

[76] A. Borodin, N. Linial, and M. E. Saks, "An Optimal On-line Algorithm for Metrical Task System," *Journal of the ACM*, vol. 39, no. 4, pp. 745–763, Oct. 1992, ISSN: 0004-5411. DOI: 10.1145/146585.146588. [Online]. Available: https://doi.org/10.1145/146585.146588.

[77] J. Friedman and N. Linial, "On Convex Body Chasing," *Discrete & Computational Geometry*, vol. 9, no. 3, pp. 293–321, Mar. 1993, ISSN: 0179-5376, 1432-0444. DOI: 10.1007/BF02189324. [Online]. Available: http://link.springer.com/10.1007/BF02189324.

[78] R. El-Yaniv, A. Fiat, R. M. Karp, and G. Turpin, "Optimal Search and One-Way Trading Online Algorithms," *Algorithmica*, vol. 30, no. 1, pp. 101–139, May 2001. DOI: 10.1007/s00453-001-0003-0. [Online]. Available: https://doi.org/10.1007/s00453-001-0003-0.

[79] J. Lorenz, K. Panagiotou, and A. Steger, "Optimal Algorithms for k-Search with Application in Option Pricing," *Algorithmica*, vol. 55, no. 2, pp. 311–328, Aug. 2008. DOI: 10.1007/s00453-008-9217-8. [Online]. Available: https://doi.org/10.1007/s00453-008-9217-8.

[80] P. L. Donti, D. Rolnick, and J. Z. Kolter, "DC3: A Learning Method for Optimization with Hard Constraints," in *International Conference on Learning Representations*, Oct. 2020. [Online]. Available: https://openreview.net/forum?id=V1ZHVxJ6dSS.

[81] Z. Shi, Y. Wang, H. Zhang, J. Yi, and C.-J. Hsieh. "Fast Certified Robust Training with Short Warmup." arXiv: 2103.17268 [cs], [Online]. Available: http://arxiv.org/abs/2103.17268.

[82] Z. Shi, C.-J. Hsieh, and H. Zhang. "Certified Training with Branch-and-Bound: A Case Study on Lyapunov-stable Neural Control." arXiv: 2411.18235 [cs], [Online]. Available: http://arxiv.org/abs/2411.18235.

[83] E. Even-Dar, M. Kearns, and J. Wortman, "Risk-Sensitive Online Learning," in *Algorithmic Learning Theory*, J. L. Balcázar, P. M. Long, and F. Stephan, Eds., Berlin, Heidelberg: Springer, 2006, pp. 199–213. DOI: 10.1007/11894841_18.

[84] L. Leqi, A. Huang, Z. Lipton, and K. Azizzadenesheli, "Supervised Learning with General Risk Functionals," in *Proceedings of the 39th International Conference on Machine Learning*, PMLR, Jun. 28, 2022, pp. 12 570–12 592. [Online]. Available: https://proceedings.mlr.press/v162/leqi22a.html.

[85] Y. Chow and M. Ghavamzadeh, "Algorithms for CVaR Optimization in MDPs," in *Advances in Neural Information Processing Systems*, vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/35f1050a4381d2d216bf56ad46b0277d-Paper.pdf.

[86] K. Wang, N. Kallus, and W. Sun, "Near-Minimax-Optimal Risk-Sensitive Reinforcement Learning with CVaR," in *Proceedings of the 40th International Conference on Machine Learning*, PMLR, Jul. 3, 2023, pp. 35 864–35 907. [Online]. Available: https://proceedings.mlr.press/v202/wang23m.html.

[87] Y. Chow, A. Tamar, S. Mannor, and M. Pavone, "Risk-Sensitive and Robust Decision-Making: A CVaR Optimization Approach," in *Advances in Neural Information Processing Systems*, vol. 28, Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/64223ccf70bbb65a3a4aceac37e21016-Paper.pdf.

[88] A. N. Madavan and S. Bose, "A Stochastic Primal-Dual Method for Optimization with Conditional Value at Risk Constraints," *Journal of Optimization Theory and Applications*, vol. 190, no. 2, pp. 428–460, Aug. 2021, ISSN: 1573-2878. DOI: 10.1007/s10957-021-01888-x.

[89] M. Ahmadi, X. Xiong, and A. D. Ames, "Risk-Averse Control via CVaR Barrier Functions: Application to Bipedal Robot Locomotion," *IEEE Control Systems Letters*, vol. 6, pp. 878–883, 2022, ISSN: 2475-1456. DOI: 10.1109/LCSYS.2021.3086854.

[90] A. N. Angelopoulos and S. Bates, "Conformal Prediction: A Gentle Introduction," *Foundations and Trends® in Machine Learning*, vol. 16, no. 4, pp. 494–591, 2023, ISSN: 1935-8237. DOI: 10.1561/2200000101. [Online]. Available: http://dx.doi.org/10.1561/2200000101.

[91] A. N. Angelopoulos, S. Bates, A. Fisch, L. Lei, and T. Schuster. "Conformal Risk Control." arXiv: 2208.02814 [cs, math, stat], [Online]. Available: http://arxiv.org/abs/2208.02814.

[92] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate Uncertainties for Deep Learning Using Calibrated Regression," in *Proceedings of the 35th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, Jul. 2018, pp. 2796–2804. [Online]. Available: https://proceedings.mlr.press/v80/kuleshov18a.html.

[93] S. Albers and J. Quedenfeld, "Optimal Algorithms for Right-Sizing Data Centers," in *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures*, ser. SPAA '18, New York, NY, USA: Association for Computing Machinery, Jul. 2018, pp. 363–372, ISBN: 978-1-4503-5799-9. DOI: 10.1145/3210377.3210385. [Online]. Available: https://doi.org/10.1145/3210377.3210385.

[94] A. Antoniadis, C. Coester, M. Elias, A. Polak, and B. Simon, "Learning-Augmented Dynamic Power Management with Multiple States via New Ski Rental Bounds," in *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 16 714–16 726. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/8b8388180314a337c9aa3c5aa8e2f37a-Paper.pdf.

[95] Y. Zhang, M. H. Hajiesmaili, S. Cai, M. Chen, and Q. Zhu, "Peak-Aware Online Economic Dispatching for Microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 323–335, Jan. 2018, ISSN: 1949-3061. DOI: 10.1109/

TSG.2016.2551282. [Online]. Available: https://ieeexplore.ieee.org/document/7448471.

[96] R. Lee, B. Sun, M. Hajiesmaili, and J. C. S. Lui, "Online Search with Predictions: Pareto-optimal Algorithm and its Applications in Energy Markets," in *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, ser. E-Energy '24, New York, NY, USA: Association for Computing Machinery, May 2024, pp. 386–407. DOI: 10.1145/3632775.3639590. [Online]. Available: https://dl.acm.org/doi/10.1145/3632775.3639590.

[97] E. Koutsoupias and C. H. Papadimitriou, "On the k-server Conjecture," *Journal of the ACM*, vol. 42, no. 5, pp. 971–983, Sep. 1995, ISSN: 0004-5411. DOI: 10.1145/210118.210128. [Online]. Available: https://doi.org/10.1145/210118.210128.

[98] S.-J. Kim and G. B. Giannakis, "Real-time Electricity Pricing for Demand Response Using Online Convex Optimization," in *Innovative Smart Grid Technologies Conference 2014*, Feb. 2014, pp. 1–5. DOI: 10.1109/ISGT.2014.6816447.

[99] G. Goel and A. Wierman, "An Online Algorithm for Smoothed Regression and LQR Control," in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, ISSN: 2640-3498, PMLR, Apr. 2019, pp. 2504–2513. [Online]. Available: https://proceedings.mlr.press/v89/goel19a.html.

[100] Y. Li, G. Qu, and N. Li, "Online Optimization With Predictions and Switching Costs: Fast Algorithms and the Fundamental Limit," *IEEE Transactions on Automatic Control*, vol. 66, no. 10, pp. 4761–4768, Oct. 2021, ISSN: 1558-2523. DOI: 10.1109/TAC.2020.3040249.

[101] S. Bubeck, B. Klartag, Y. T. Lee, Y. Li, and M. Sellke, "Chasing Nested Convex Bodies Nearly Optimally," in *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, ser. Proceedings, Society for Industrial and Applied Mathematics, Dec. 2019, pp. 1496–1508. DOI: 10.1137/1.9781611975994.91. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611975994.91.

[102] N. Chen, G. Goel, and A. Wierman, "Smoothed Online Convex Optimization in High Dimensions via Online Balanced Descent," in *Proceedings of the 31st Conference On Learning Theory*, ISSN: 2640-3498, PMLR, Jul. 2018, pp. 1574–1594. [Online]. Available: https://proceedings.mlr.press/v75/chen18b.html.

[103] L. Zhang, W. Jiang, S. Lu, and T. Yang, "Revisiting Smoothed Online Learning," in *Proceedings of NeurIPS '21*, vol. 34, 2021, pp. 13 599–13 612. [Online]. Available: https://openreview.net/forum?id=sn0wj3Dci2J.

[104] C. J. Argue, S. Bubeck, M. B. Cohen, A. Gupta, and Y. T. Lee, "A Nearly-Linear Bound for Chasing Nested Convex Bodies," in *Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Society for Industrial and Applied Mathematics, Jan. 2019, pp. 117–122. DOI: 10.1137/1.9781611975482.8. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611975482.8.

[105] C. J. Argue, A. Gupta, Z. Tang, and G. Guruganesh, "Chasing Convex Bodies with Linear Competitive Ratio," *Journal of the ACM*, vol. 68, no. 5, pp. 1–10, 2021. DOI: 10.1137/1.9781611975994.

[106] M. Sellke, "Chasing Convex Bodies Optimally," in *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, USA: Society for Industrial and Applied Mathematics, Jan. 2020, pp. 1509–1518. [Online]. Available: https://dl.acm.org/doi/10.5555/3381089.3381181.

[107] Y. Lin, Personal correspondence, 2022.

[108] C. J. Argue, A. Gupta, and G. Guruganesh, "Dimension-Free Bounds for Chasing Convex Functions," in *Proceedings of Thirty Third Conference on Learning Theory*, ISSN: 2640-3498, PMLR, Jul. 2020, pp. 219–241. [Online]. Available: https://proceedings.mlr.press/v125/argue20a.html.

[109] M. Mahdian, H. Nazerzadeh, and A. Saberi, "Online Optimization with Uncertain Information," *ACM Transactions on Algorithms*, vol. 8, no. 1, pp. 1–29, Jan. 2012, ISSN: 1549-6325, 1549-6333. DOI: 10.1145/2071379.2071381. [Online]. Available: https://dl.acm.org/doi/10.1145/2071379.2071381.

[110] A. Wei and F. Zhang, "Optimal Robustness-Consistency Trade-offs for Learning-Augmented Online Algorithms," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 8042–8053. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/5bd844f11fa520d54fa5edec06ea2507-Paper.pdf.

[111] S. Angelopoulos, C. Dürr, S. Jin, S. Kamali, and M. Renault, "Online Computation with Untrusted Advice," in *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, T. Vidick, Ed., ser. Leibniz International Proceedings in Informatics (LIPIcs), ISSN: 1868-8969, vol. 151, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020, 52:1–52:15, ISBN: 978-3-95977-134-4. DOI: 10.4230/LIPIcs.ITCS.2020.52. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2020/11737.

[112] N. Golowich and A. Moitra, "Can Q-Learning be Improved with Advice?" *arXiv:2110.13052 [cs, math, stat]*, Oct. 2021, arXiv: 2110.13052. [Online]. Available: http://arxiv.org/abs/2110.13052.

[113] D. Rutten, N. Christianson, D. Mukherjee, and A. Wierman, "Smoothed Online Optimization with Unreliable Predictions," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 7, no. 1, 12:1–12:36, Mar. 2023. DOI: 10.1145/3579442. [Online]. Available: https://dl.acm.org/doi/10.1145/3579442.

[114] A. Fiat, Y. Rabani, and Y. Ravid, "Competitive k-server Algorithms," *Journal of Computer and System Sciences*, vol. 48, no. 3, pp. 410–428, Jun. 1994, ISSN: 0022-0000. DOI: 10.1016/S0022-0000(05)80060-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022000005800601.

[115] R. Baeza-Yates, J. Culberson, and G. Rawlins, "Searching in the Plane," *Information and Computation*, vol. 106, no. 2, pp. 234–252, Oct. 1993, ISSN: 0890-5401. DOI: 10.1006/inco.1993.1054. [Online]. Available: https://doi.org/10.1006/inco.1993.1054.

[116] A. Blum and C. Burch, "On-line Learning and the Metrical Task System Problem," in *Proceedings of the Tenth Annual Conference on Computational Learning Theory - COLT '97*, Nashville, Tennessee, United States: ACM Press, 1997, pp. 45–53, ISBN: 978-0-89791-891-6. DOI: 10.1145/267460.267475. [Online]. Available: http://portal.acm.org/citation.cfm?doid=267460.267475.

[117] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997, ISSN: 0022-0000. DOI: 10.1006/jcss.1997.1504. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002200009791504X.

[118] S. Angelopoulos, "Online Search with a Hint," in *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, J. R. Lee, Ed., ser. Leibniz International Proceedings in Informatics (LIPIcs), ISSN: 1868-8969, vol. 185, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021, 51:1–51:16, ISBN: 978-3-95977-177-1. DOI: 10.4230/LIPIcs.ITCS.2021.51. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2021/13590.

[119] A. Antoniadis and K. Schewior, "A Tight Lower Bound for Online Convex Optimization with Switching Costs," in *Approximation and Online Algorithms*, R. Solis-Oba and R. Fleischer, Eds., vol. 10787, Springer International Publishing, 2018, pp. 164–175. DOI: 10.1007/978-3-319-89441-6_13. [Online]. Available: http://link.springer.com/10.1007/978-3-319-89441-6_13.

[120] N. Bansal, A. Gupta, R. Krishnaswamy, K. Pruhs, K. Schewior, and C. Stein, "A 2-Competitive Algorithm For Online Convex Optimization With Switching Costs," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, N. Garg, K. Jansen, A. Rao, and J. D. P. Rolim, Eds., ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 40, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 96–109, ISBN: 978-3-939897-89-7. DOI: 10.4230/LIPIcs.APPROX-RANDOM.2015.96. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2015/5297.

[121] J.-L. Joly, "Caracterisations d'espaces hilbertiens au moyen de la constante rectangle," *Journal of Approximation Theory*, vol. 2, pp. 301–311, 1969.

[122] M. Baronti, E. Casini, and P. L. Papini, "Revisiting the Rectangular Constant in Banach Spaces," *Bulletin of the Australian Mathematical Society*, pp. 1–10, Apr. 2021, ISSN: 0004-9727, 1755-1633. DOI: 10.1017/S0004972721000253. [Online]. Available: https://www.cambridge.org/core/product/identifier/S0004972721000253/type/journal_article.

[123] M. A. Rieffel, "Lipschitz Extension Constants Equal Projection Constants," *Contemporary Mathematics*, vol. 414, pp. 147–162, 2006, arXiv: math/0508097. [Online]. Available: http://arxiv.org/abs/math/0508097.

[124] R. L. Thele, "Some Results on the Radial Projection in Banach Spaces," *Proceedings of the American Mathematical Society*, vol. 42, no. 2, pp. 483–486, 1974, Publisher: American Mathematical Society, ISSN: 0002-9939. DOI: 10.2307/2039529. [Online]. Available: https://www.jstor.org/stable/2039529.

[125] M. Bienkowski, J. Byrka, M. Chrobak, C. Coester, L. Jez, and E. Koutsoupias, "Better Bounds for Online Line Chasing," in *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, P. Rossmanith, P. Heggernes, and J.-P. Katoen, Eds., ser. Leibniz International Proceedings in Informatics (LIPIcs), ISSN: 1868-8969, vol. 138, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 8:1–8:13, ISBN: 978-3-95977-117-7. DOI: 10.4230/LIPIcs.MFCS.2019.8. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2019/10952.

[126] G. Birkhoff, "Orthogonality in Linear Metric Spaces," *Duke Mathematical Journal*, vol. 1, no. 2, pp. 169–172, Jun. 1935, ISSN: 0012-7094, 1547-7398. DOI: 10.1215/S0012-7094-35-00115-6. [Online]. Available: https://projecteuclid.org/journals/duke-mathematical-journal/volume-1/issue-2/Orthogonality-in-linear-metric-spaces/10.1215/S0012-7094-35-00115-6.full.

[127] R. C. James, "Orthogonality and Linear Functionals in Normed Linear Spaces," *Transactions of the American Mathematical Society*, vol. 61, no. 2, pp. 265–292, 1947, ISSN: 0002-9947, 1088-6850. DOI: 10.1090/S0002-9947-1947-0021241-4. [Online]. Available: https://www.ams.org/tran/1947-061-02/S0002-9947-1947-0021241-4/.

[128] J. Desbiens, "Constante rectangle et biais d'un espace de Banach," *Bulletin of the Australian Mathematical Society*, vol. 42, no. 3, pp. 465–482, Dec. 1990, ISSN: 0004-9727, 1755-1633. DOI: 10.1017/S000497270002863X. [Online]. Available: https://www.cambridge.org/core/product/identifier/S000497270002863X/type/journal_article.

[129] B. Bollobás, *Linear Analysis: An Introductory Course* (Cambridge Mathematical Textbooks), 2nd. Cambridge; New York: Cambridge University Press, 1999, ISBN: 978-0-521-65577-4.

[130] N. Buchbinder, S. Chen, J. S. Naor, and O. Shamir, "Unified Algorithms for Online Learning and Competitive Analysis," in *Proceedings of the 25th Annual Conference on Learning Theory*, ISSN: 1938-7228, JMLR Workshop and Conference Proceedings, Jun. 2012, pp. 5.1–5.18. [Online]. Available: https://proceedings.mlr.press/v23/buchbinder12.html.

[131] A. Daniely and Y. Mansour, "Competitive Ratio vs Regret Minimization: Achieving the Best of Both Worlds," in *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, ISSN: 2640-3498, PMLR, Mar. 2019, pp. 333–368. [Online]. Available: https://proceedings.mlr.press/v98/daniely19a.html.

[132] Y. Li, G. Qu, and N. Li, *Online Optimization with Predictions and Switching Costs: Fast Algorithms and the Fundamental Limit*, Jan. 2018. [Online]. Available: https://arxiv.org/abs/1801.07780v3.

[133] N. Christianson, C. Yeh, T. Li, M. Hosseini, M. T. Rad, A. Golmohammadi, and A. Wierman, "Robust Machine-Learned Algorithms for Efficient Grid Operation," *Environmental Data Science*, vol. 4, e24, Apr. 2025, ISSN: 2634-4602. DOI: 10.1017/eds.2024.28. [Online]. Available: https://www.cambridge.org/core/journals/environmental-data-science/article/robust-machinelearned-algorithms-for-efficient-grid-operation/0E29B936A8FEE565B169F70372B7F9DE.

[134] S. Albers and J. Quedenfeld, "Algorithms for Right-Sizing Heterogeneous Data Centers," in *Proceedings of the 33rd ACM Symposium on Parallelism in Algorithms and Architectures*, ser. SPAA '21, New York, NY, USA: Association for Computing Machinery, Jul. 2021, pp. 48–58, ISBN: 978-1-4503-8070-6. DOI: 10.1145/3409964.3461789. [Online]. Available: https://doi.org/10.1145/3409964.3461789.

[135] S. Bubeck, Y. Rabani, and M. Sellke, “Online Multiserver Convex Chasing and Optimization,” in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Society for Industrial and Applied Mathematics, Jan. 2021, pp. 2093–2104. DOI: 10.1137/1.9781611976465.125. [Online]. Available: https://epubs.siam.org/doi/10.1137/1.9781611976465.125.

[136] S. Deng, J. Li, and Y. Rabani, “Approximation Algorithms for Clustering with Dynamic Points,” *Journal of Computer and System Sciences*, vol. 130, pp. 43–70, Dec. 2022, ISSN: 0022-0000. DOI: 10.1016/j.jcss.2022.07.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022000022000514.

[137] S. Dehghani, S. Ehsani, M. Hajiaghayi, V. Liaghat, and S. Seddighin, “Stochastic k-Server: How Should Uber Work?” In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, I. Chatzigiannakis, P. Indyk, F. Kuhn, and A. Muscholl, Eds., ser. Leibniz International Proceedings in Informatics (LIPIcs), ISSN: 1868-8969, vol. 80, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, 126:1–126:14, ISBN: 978-3-95977-041-5. DOI: 10.4230/LIPIcs.ICALP.2017.126. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2017/7480.

[138] S. Bubeck, C. Coester, and Y. Rabani, “The Randomized k-Server Conjecture is False!” In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC 2023)*, ser. STOC 2023, Orlando, FL, USA: Association for Computing Machinery, 2023, pp. 581–594, ISBN: 9781450399135. DOI: 10.1145/3564246.3585132. [Online]. Available: https://doi.org/10.1145/3564246.3585132.

[139] M. Mitzenmacher and S. Vassilvitskii, “Algorithms with Predictions,” in *Beyond the Worst-Case Analysis of Algorithms*, T. Roughgarden, Ed., Cambridge: Cambridge University Press, 2021, pp. 646–662, ISBN: 978-1-108-49431-1. DOI: 10.1017/9781108637435.037. [Online]. Available: https://www.cambridge.org/core/books/beyond-the-worstcase-analysis-of-algorithms/algorithms-with-predictions/D8E70B699F40C0704CB5FEE83878EC94.

[140] N. Christianson, T. Handina, and A. Wierman, “Chasing Convex Bodies and Functions with Black-Box Advice,” in *Proceedings of the Thirty Fifth Conference on Learning Theory*, PMLR, Jun. 2022, pp. 867–908. [Online]. Available: https://proceedings.mlr.press/v178/christianson22a.html.

[141] A. Lindermayr, N. Megow, and B. Simon, “Double Coverage with Machine-Learned Advice,” in *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, M. Braverman, Ed., ser. Leibniz International Proceedings in Informatics (LIPIcs), ISSN: 1868-8969, vol. 215, Dagstuhl,

Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 99:1–99:18, ISBN: 978-3-95977-217-4. DOI: 10.4230/LIPIcs.ITCS.2022.99. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2022/15695.

[142] S. Bubeck, M. B. Cohen, J. R. Lee, and Y. T. Lee, "Metrical Task Systems on Trees via Mirror Descent and Unfair Gluing," *SIAM Journal on Computing*, vol. 50, no. 3, pp. 909–923, Jan. 2021, ISSN: 0097-5397, 1095-7111. DOI: 10.1137/19M1237879. [Online]. Available: https://epubs.siam.org/doi/10.1137/19M1237879.

[143] C. Coester and J. R. Lee, "Pure Entropic Regularization for Metrical Task Systems," in *Proceedings of the Thirty-Second Conference on Learning Theory*, ISSN: 2640-3498, PMLR, Jun. 2019, pp. 835–848. [Online]. Available: https://proceedings.mlr.press/v99/coester19a.html.

[144] Y. Lin, G. Goel, and A. Wierman, "Online Optimization with Predictions and Non-convex Losses," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 4, no. 1, 18:1–18:32, May 2020. DOI: 10.1145/3379484. [Online]. Available: https://doi.org/10.1145/3379484.

[145] M. Manasse, L. McGeoch, and D. Sleator, "Competitive Algorithms for On-line Problems," in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, ser. STOC '88, New York, NY, USA: Association for Computing Machinery, Jan. 1988, pp. 322–333, ISBN: 978-0-89791-264-8. DOI: 10.1145/62212.62243. [Online]. Available: https://doi.org/10.1145/62212.62243.

[146] E. Koutsoupias, "The k-server Problem," *Computer Science Review*, vol. 3, no. 2, pp. 105–118, May 2009, ISSN: 15740137. DOI: 10.1016/j.cosrev.2009.04.002. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1574013709000197.

[147] N. Bansal, N. Buchbinder, A. Madry, and J. S. Naor, "A Polylogarithmic-Competitive Algorithm for the k-Server Problem," *Journal of the ACM*, vol. 62, no. 5, 40:1–40:49, Nov. 2015, ISSN: 0004-5411. DOI: 10.1145/2783434. [Online]. Available: https://doi.org/10.1145/2783434.

[148] S. Bubeck, M. B. Cohen, Y. T. Lee, J. R. Lee, and A. Madry, "k-server via Multiscale Entropic Regularization," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2018, New York, NY, USA: Association for Computing Machinery, Jun. 2018, pp. 3–16, ISBN: 978-1-4503-5559-9. DOI: 10.1145/3188745.3188798. [Online]. Available: https://doi.org/10.1145/3188745.3188798.

[149] S. S. Ramesh, P. G. Sessa, A. Krause, and I. Bogunovic, "Movement Penalized Bayesian Optimization with Application to Wind Energy Systems," in *ICML 2022 Workshop on Adaptive Experimental Design and Active Learn-*

ing in the Real World, 2022. [Online]. Available: https://arxiv.org/abs/2210.08087.

[150] S. S. Zhang, “The Effect of the Charging Protocol on the Cycle Life of a Li-ion Battery,” *Journal of Power Sources*, vol. 161, no. 2, pp. 1385–1391, Oct. 2006, ISSN: 03787753. DOI: 10.1016/j.jpowsour.2006.06.040. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378775306011839.

[151] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*. Cambridge University Press, 2005.

[152] J. King, A. Clifton, and B.-M. Hodge, “Validation of Power Output for the WIND Toolkit,” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep. NREL/TP-5D00-61714, 2014.

[153] P. Virtanen et al., “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[154] F. Santambrogio, *Optimal Transport for Applied Mathematicians* (Progress in Nonlinear Differential Equations and Their Applications). Cham: Springer International Publishing, 2015, vol. 87, ISBN: 978-3-319-20827-5 978-3-319-20828-2. DOI: 10.1007/978-3-319-20828-2. [Online]. Available: http://link.springer.com/10.1007/978-3-319-20828-2.

[155] A. Radovanović, R. Koningstein, I. Schneider, B. Chen, A. Duarte, B. Roy, D. Xiao, M. Haridasan, P. Hung, N. Care, et al., “Carbon-Aware Computing for Datacenters,” *IEEE Transactions on Power Systems*, vol. 38, no. 2, pp. 1270–1280, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9770383.

[156] B. Acun, B. Lee, F. Kazhamiaka, K. Maeng, U. Gupta, M. Chakkaravarthy, D. Brooks, and C.-J. Wu, “Carbon Explorer: A Holistic Framework for Designing Carbon Aware Datacenters,” in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ser. ASPLOS 2023, Vancouver, BC, Canada: Association for Computing Machinery, 2023, pp. 118–132, ISBN: 9781450399166. DOI: 10.1145/3575693.3575754. [Online]. Available: https://doi.org/10.1145/3575693.3575754.

[157] N. Bashir, T. Guo, M. Hajiesmaili, D. Irwin, P. Shenoy, R. Sitaraman, A. Souza, and A. Wierman, “Enabling Sustainable Clouds: The Case for Virtualizing the Energy System,” in *Proceedings of the ACM Symposium on Cloud Computing*, ser. SoCC ’21, Seattle, WA, USA: Association for Computing Machinery, 2021, pp. 350–358, ISBN: 9781450386388. DOI: 10.1145/3472883.3487009. [Online]. Available: https://doi.org/10.1145/3472883.3487009.

[158] W. A. Hanafy, R. Bostandoost, N. Bashir, D. Irwin, M. Hajiesmaili, and P. Shenoy, "The War of the Efficiencies: Understanding the Tension between Carbon and Energy Optimization," in *Proceedings of the 2nd Workshop on Sustainable Computer Systems*, ACM, Jul. 2023. DOI: 10.1145/3604930.3605709. [Online]. Available: https://doi.org/10.1145%2F3604930.3605709.

[159] S. Rajbhandari, O. Ruwase, J. Rasley, S. Smith, and Y. He, "ZeRO-Infinity: Breaking the GPU Memory Wall for Extreme Scale Deep Learning," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–14. [Online]. Available: https://dl.acm.org/doi/10.1145/3458817.3476205.

[160] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti, et al., "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, a Large-Scale Generative Language Model," *arXiv preprint arXiv:2201.11990*, 2022. [Online]. Available: https://arxiv.org/abs/2201.11990.

[161] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models," in *International Conference on Machine Learning*, PMLR, Honolulu, Hawai'i, 2023, pp. 38 087–38 099. [Online]. Available: https://proceedings.mlr.press/v202/xiao23c/xiao23c.pdf.

[162] D. Maji, R. K. Sitaraman, and P. Shenoy, "DACF: Day-Ahead Carbon Intensity Forecasting of Power Grids Using Machine Learning," in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, ser. e-Energy '22, New York, NY, USA: Association for Computing Machinery, 2022, pp. 188–192, ISBN: 9781450393973. DOI: 10.1145/3538637.3538849. [Online]. Available: https://doi.org/10.1145/3538637.3538849.

[163] L. Yang, M. H. Hajiesmaili, R. Sitaraman, A. Wierman, E. Mallada, and W. S. Wong, "Online Linear Optimization with Inventory Management Constraints," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 4, no. 1, May 2020. DOI: 10.1145/3379482. [Online]. Available: https://doi.org/10.1145/3379482.

[164] Z. Zhang, Z. Li, and C. Wu, "Optimal Posted Prices for Online Cloud Resource Allocation," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, pp. 1–26, Jun. 2017. [Online]. Available: https://dl.acm.org/doi/10.1145/3084460.

[165] P. Ambati, N. Bashir, D. Irwin, M. Hajiesmaili, and P. Shenoy, "Hedge Your Bets: Optimizing Long-term Cloud Costs by Mixing VM Purchasing Options," in *2020 IEEE International Conference on Cloud Engineering (IC2E)*, 2020, pp. 105–115. DOI: 10.1109/IC2E48712.2020.00018.

[166] S. Shastri, A. Rizk, and D. Irwin, "Transient Guarantees: Maximizing the Value of Idle Cloud Capacity," in *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE, 2016, pp. 992–1002. [Online]. Available: https://ieeexplore.ieee.org/document/7877163.

[167] B. Sun, R. Lee, M. Hajiesmaili, A. Wierman, and D. Tsang, "Pareto-Optimal Learning-Augmented Algorithms for Online Conversion Problems," in *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 10 339–10 350. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/55a988dfb00a914717b3000a3374694c-Paper.pdf.

[168] Y. Zhou, D. Chakrabarty, and R. Lukose, "Budget Constrained Bidding in Keyword Auctions and Online Knapsack Problems," in *Lecture Notes in Computer Science*, Heidelberg, DE: Springer Berlin Heidelberg, 2008, pp. 566–576. DOI: 10.1007/978-3-540-92185-1_63. [Online]. Available: https://doi.org/10.1007/978-3-540-92185-1_63.

[169] B. Sun, L. Yang, M. Hajiesmaili, A. Wierman, J. C. Lui, D. Towsley, and D. H. Tsang, "The Online Knapsack Problem with Departures," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 6, no. 3, pp. 1–32, 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3570618.

[170] L. Yang, A. Zeynali, M. H. Hajiesmaili, R. K. Sitaraman, and D. Towsley, "Competitive Algorithms for Online Multidimensional Knapsack Problems," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 5, no. 3, Dec. 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3491042.

[171] *Electricity Map*, https://www.electricitymap.org/map, Accessed June 2024.

[172] A. Hoorfar and M. Hassani, "Inequalities on the Lambert W Function and Hyperpower Function," *Journal of Inequalities in Pure and Applied Mathematics*, vol. 9, no. 51, 2 Jan. 2008. [Online]. Available: https://eudml.org/doc/130024.

[173] S. M. Stewart, "On Certain Inequalities Involving the Lambert W function," *Journal of Inequalities in Pure and Applied Mathematics*, vol. 10, no. 96, 4 Nov. 2009. [Online]. Available: https://eudml.org/doc/224588.

[174] E. Mohr, I. Ahmad, and G. Schmidt, "Online Algorithms for Conversion Problems: A Survey," *Surveys in Operations Research and Management Science*, vol. 19, no. 2, pp. 87–104, Jul. 2014. DOI: 10.1016/j.sorms.2014.08.001. [Online]. Available: https://doi.org/10.1016/j.sorms.2014.08.001.

[175] P. Damaschke, P. H. Ha, and P. Tsigas, “Online Search with Time-Varying Price Bounds,” *Algorithmica*, vol. 55, no. 4, pp. 619–642, Dec. 2007. DOI: 10.1007/s00453-007-9156-9. [Online]. Available: https://doi.org/10.1007/s00453-007-9156-9.

[176] B. Sun, A. Zeynali, T. Li, M. Hajiesmaili, A. Wierman, and D. H. Tsang, “Competitive Algorithms for the Online Multiple Knapsack Problem with Application to Electric Vehicle Charging,” *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 4, no. 3, pp. 1–32, Nov. 2020, ISSN: 2476-1249. DOI: 10.1145/3428336.

[177] N. Christianson, J. Shen, and A. Wierman, “Optimal Robustness-Consistency Tradeoffs for Learning-Augmented Metrical Task Systems,” in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, PMLR, Apr. 2023, pp. 9377–9399. [Online]. Available: https://proceedings.mlr.press/v206/christianson23a.html.

[178] P. Li, J. Yang, and S. Ren, “Expert-Calibrated Learning for Online Optimization with Switching Costs,” *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 6, no. 2, pp. 1–35, May 2022, ISSN: 2476-1249. DOI: 10.1145/3530894. [Online]. Available: https://dl.acm.org/doi/10.1145/3530894.

[179] V. Gupta, P. Shenoy, and R. K. Sitaraman, “Combining Renewable Solar and Open Air Cooling for Greening Internet-Scale Distributed Networks,” in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, 2019, pp. 303–314. [Online]. Available: https://dl.acm.org/doi/10.1145/3307772.3328307.

[180] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, “Greening Geographical Load Balancing,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 39, no. 1, pp. 193–204, 2011. [Online]. Available: https://dl.acm.org/doi/10.1145/2007116.2007139.

[181] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, “Renewable and Cooling Aware Workload Management for Sustainable Data Centers,” in *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, 2012, pp. 175–186. [Online]. Available: https://dl.acm.org/doi/10.1145/2318857.2254779.

[182] A. Souza, N. Bashir, J. Murillo, W. Hanafy, Q. Liang, D. Irwin, and P. Shenoy, “Ecovisor: A Virtual Energy System for Carbon-Efficient Applications,” in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ser. ASPLOS 2023, Vancouver, BC, Canada: Association for Computing Machinery, 2023, pp. 252–265, ISBN: 9781450399166. DOI: 10.1145/3575693.3575709. [Online]. Available: https://doi.org/10.1145/3575693.3575709.

[183] National Center for Biotechnology Information, *Basic Local Alignment Search Tool (BLAST)*, https://blast.ncbi.nlm.nih.gov, Accessed March 2022.

[184] M. Y. Kitaev and R. F. Serfozo, "M/M/1 Queues with Switching Costs and Hysteretic Optimal Control," *Operations Research*, vol. 47, no. 2, pp. 310–312, 1999. DOI: 10.1287/opre.47.2.310.

[185] A. Gandhi, M. Harchol-Balter, and I. Adan, "Server Farms with Setup Costs," *Performance Evaluation*, vol. 67, no. 11, pp. 1123–1138, 2010, ISSN: 0166-5316. DOI: https://doi.org/10.1016/j.peva.2010.07.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166531610000957.

[186] I. Mitrani, "Managing Performance and Power Consumption in a Server Farm," *Annals of Operations Research*, vol. 202, no. 1, pp. 121–134, Jul. 2011. DOI: 10.1007/s10479-011-0932-1. [Online]. Available: https://doi.org/10.1007/s10479-011-0932-1.

[187] N. M. Asghari, M. Mandjes, and A. Walid, "Energy-efficient Scheduling in Multi-core Servers," *Computer Networks*, vol. 59, pp. 33–43, 2014, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.bjp.2013.12.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138912861300412X.

[188] Z. Yang, M.-H. Chen, Z. Niu, and D. Huang, "An Optimal Hysteretic Control Policy for Energy Saving in Cloud Computing," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, 2011, pp. 1–5. DOI: 10.1109/GLOCOM.2011.6133628.

[189] N. Bansal and C. Coester, "Online Metric Allocation and Time-Varying Regularization," in *30th Annual European Symposium on Algorithms (ESA 2022)*, S. Chechik, G. Navarro, E. Rotenberg, and G. Herman, Eds., ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 244, Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 13:1–13:13, ISBN: 978-3-95977-247-1. DOI: 10.4230/LIPIcs.ESA.2022.13. [Online]. Available: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2022.13.

[190] R. Jenatton, J. Huang, and C. Archambeau, "Adaptive Algorithms for Online Convex Optimization with Long-term Constraints," in *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., ser. Proceedings of Machine Learning Research, vol. 48, New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 402–411. [Online]. Available: https://proceedings.mlr.press/v48/jenatton16.html.

[191] A. Badanidiyuru, R. Kleinberg, and A. Slivkins, "Bandits with Knapsacks," *Journal of the ACM*, vol. 65, no. 3, Mar. 2018, ISSN: 0004-5411. DOI: 10.1145/3164539. [Online]. Available: https://doi.org/10.1145/3164539.

[192] A. Lechowicz, N. Christianson, J. Zuo, N. Bashir, M. Hajiesmaili, A. Wierman, and P. Shenoy, "The Online Pause and Resume Problem: Optimal Algorithms and An Application to Carbon-Aware Load Shifting," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 7, no. 3, 45:1–45:32, Dec. 2023. DOI: 10.1145/3626776. [Online]. Available: https://doi.org/10.1145/3626776.

[193] A. Lechowicz, N. Christianson, B. Sun, N. Bashir, M. Hajiesmaili, A. Wierman, and P. Shenoy, *Online Conversion with Switching Costs: Robust and Learning-Augmented Algorithms*, Jan. 2024. DOI: 10.48550/arXiv.2310.20598. [Online]. Available: http://arxiv.org/abs/2310.20598.

[194] R. Bostandoost, A. Lechowicz, W. A. Hanafy, N. Bashir, P. Shenoy, and M. Hajiesmaili, "LACS: Learning-Augmented Algorithms for Carbon-Aware Resource Scaling with Uncertain Demand," in *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, Singapore, 2024, pp. 27–45, ISBN: 9798400704802. DOI: 10.1145/3632775.3661942. [Online]. Available: https://doi.org/10.1145/3632775.3661942.

[195] A. Lechowicz, N. Christianson, B. Sun, N. Bashir, M. Hajiesmaili, A. Wierman, and P. Shenoy, "Chasing Convex Functions with Long-term Constraints," in *Proceedings of the 41st International Conference on Machine Learning*, PMLR, 2024. [Online]. Available: https://proceedings.mlr.press/v235/lechowicz24a.html.

[196] Y. Bartal, A. Blum, C. Burch, and A. Tomkins, "A polylog(n)-competitive Algorithm for Metrical Task Systems," in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, El Paso, TX, USA: Association for Computing Machinery, 1997, pp. 711–719. [Online]. Available: https://dl.acm.org/doi/10.1145/258533.258667.

[197] N. Bansal, N. Buchbinder, and J. Naor, "Metrical Task Systems and the k-server Problem on HSTs," in *International Colloquium on Automata, Languages, and Programming*, Springer: Springer, 2010, pp. 287–298. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-14165-2_25.

[198] J. Abernethy, P. L. Bartlett, N. Buchbinder, and I. Stanton, "A Regularization Approach to Metrical Task Systems," in *Algorithmic Learning Theory*. Springer Berlin Heidelberg, 2010, pp. 270–284, ISBN: 9783642161087. DOI: 10.1007/978-3-642-16108-7_23. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-16108-7_23.

[199] B. Jin and W. Ma, "Online Bipartite Matching with Advice: Tight Robustness-Consistency Tradeoffs for the Two-Stage Model," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14 555–14 567, 2022. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/5d68a3f05ee2aae6a0fb2d94959082a0-Abstract-Conference.html.

[200] A. A. Chien, "Driving the Cloud to True Zero Carbon," *Communications of the ACM*, vol. 64, no. 2, p. 5, Jan. 2021, ISSN: 0001-0782. DOI: 10.1145/3445037. [Online]. Available: https://doi.org/10.1145/3445037.

[201] T. Sukprasert, A. Souza, N. Bashir, D. Irwin, and P. Shenoy, "Spatiotemporal Carbon-Aware Scheduling in the Cloud: Limits and Benefits," in *Companion Proceedings of the 14th ACM International Conference on Future Energy Systems*, ser. e-Energy '23 Companion, Orlando, FL, USA: Association for Computing Machinery, 2023, ISBN: 9798400702273. DOI: 10.1145/3599733.3606301. [Online]. Available: https://doi.org/10.1145/3599733.3606301.

[202] A. A. Chien, L. Lin, H. Nguyen, V. Rao, T. Sharma, and R. Wijayawardana, "Reducing the Carbon Impact of Generative AI Inference (Today and in 2035)," in *Proceedings of the 2nd Workshop on Sustainable Computer Systems*, ser. HotCarbon '23, Boston, MA, USA: Association for Computing Machinery, 2023, ISBN: 9798400702426. DOI: 10.1145/3604930.3605705. [Online]. Available: https://doi.org/10.1145/3604930.3605705.

[203] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Geographical Load Balancing with Renewables," *SIGMETRICS Performance Evaluation Review*, vol. 39, no. 3, pp. 62–66, Dec. 2011, ISSN: 0163-5999. DOI: 10.1145/2160803.2160862.

[204] Z. Liu, A. Wierman, Y. Chen, B. Razon, and N. Chen, "Data Center Demand Response: Avoiding the Coincident Peak via Workload Shifting and Local Generation," *Performance Evaluation*, vol. 70, no. 10, pp. 770–791, Oct. 2013, ISSN: 0166-5316. DOI: 10.1016/j.peva.2013.08.014. [Online]. Available: http://dx.doi.org/10.1016/j.peva.2013.08.014.

[205] L. Gan, A. Wierman, U. Topcu, N. Chen, and S. H. Low, "Real-time Deferrable Load Control: Handling the Uncertainties of Renewable Generation," *SIGMETRICS Performance Evaluation Review*, vol. 41, no. 3, pp. 77–79, Jan. 2014, ISSN: 0163-5999. DOI: 10.1145/2567529.2567553. [Online]. Available: https://doi.org/10.1145/2567529.2567553.

[206] A. Wierman, Z. Liu, I. Liu, and H. Mohsenian-Rad, "Opportunities and Challenges for Data Center Demand Response," in *International Green Computing Conference*, IEEE, Nov. 2014. DOI: 10.1109/igcc.2014.7039172. [Online]. Available: http://dx.doi.org/10.1109/IGCC.2014.7039172.

[207] J. Shuja, A. Gani, S. Shamshirband, R. W. Ahmad, and K. Bilal, "Sustainable Cloud Data Centers: A Survey of Enabling Techniques and Technologies," *Renewable and Sustainable Energy Reviews*, vol. 62, pp. 195–214, Sep. 2016, ISSN: 1364-0321. DOI: 10.1016/j.rser.2016.04.034. [Online]. Available: http://dx.doi.org/10.1016/j.rser.2016.04.034.

[208] A. Nadjaran Toosi, C. Qu, M. D. de Assunção, and R. Buyya, "Renewable-Aware Geographical Load Balancing of Web Applications for Sustainable Data Centers," *Journal of Network and Computer Applications*, vol. 83, pp. 155–168, 2017, ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2017.01.036. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804517300590.

[209] H. Dou, Y. Qi, W. Wei, and H. Song, "Carbon-Aware Electricity Cost Minimization for Sustainable Data Centers," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 211–223, 2017. DOI: 10.1109/TSUSC.2017.2711925.

[210] J. Zheng, A. A. Chien, and S. Suh, "Mitigating Curtailment and Carbon Emissions through Load Migration between Data Centers," *Joule*, vol. 4, no. 10, pp. 2208–2222, Oct. 2020, ISSN: 2542-4351. DOI: 10.1016/j.joule.2020.08.001. [Online]. Available: http://dx.doi.org/10.1016/j.joule.2020.08.001.

[211] W. Ma, D. Simchi-Levi, and J. Zhao, "A Competitive Analysis of Online Knapsack Problems with Unit Density," *SSRN Electronic Journal*, 2019, ISSN: 1556-5068. DOI: 10.2139/ssrn.3423199. [Online]. Available: http://dx.doi.org/10.2139/ssrn.3423199.

[212] Moxion Power. "Technology." Accessed 09-02-2024, [Online]. Available: https://www.moxionpower.com/technology/.

[213] J. Fakcharoenphol, S. Rao, and K. Talwar, "A Tight Bound on Approximating Arbitrary Metrics by Tree Metrics," *Journal of Computer and System Sciences*, vol. 69, no. 3, pp. 485–497, Oct. 2007. [Online]. Available: https://www.microsoft.com/en-us/research/publication/a-tight-bound-on-approximating-arbitrary-metrics-by-tree-metrics/.

[214] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[215] S. Diamond and S. Boyd, "CVXPY: A Python-Embedded Modeling Language for Convex Optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016, ISSN: 1533-7928. [Online]. Available: http://jmlr.org/papers/v17/15-408.html.

[216] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the Lambert W function," *Advances in Computational Mathematics*,

vol. 5, no. 1, pp. 329–359, Dec. 1996, ISSN: 1019-7168, 1572-9044. DOI: 10.1007/BF02124750.

[217] D. S. Mitrinovic, J. E. Pečarić, and A. M. Fink, *Inequalities Involving Functions and Their Integrals and Derivatives*. Springer Science & Business Media, 1991, vol. 53.

[218] Amazon Web Services, *Global Infrastructure Regions*, https://aws.amazon.com/about-aws/global-infrastructure/regions_az/, Accessed June 2024.

[219] *WattTime*, https://www.watttime.org, Accessed June 2024.

[220] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis," in *Proceedings of the Third ACM Symposium on Cloud Computing*, ser. SoCC '12, San Jose, California: Association for Computing Machinery, 2012, ISBN: 9781450317610. DOI: 10.1145/2391229.2391236. [Online]. Available: https://doi.org/10.1145/2391229.2391236.

[221] T. Sukprasert, A. Souza, N. Bashir, D. Irwin, and P. Shenoy, "On the Limitations of Carbon-Aware Temporal and Spatial Workload Shifting in the Cloud," in *Proceedings of the Nineteenth European Conference on Computer Systems*, ser. EuroSys '24, Athens, Greece: Association for Computing Machinery, 2024, pp. 924–941, ISBN: 9798400704376. DOI: 10.1145/3627703.3650079. [Online]. Available: https://doi.org/10.1145/3627703.3650079.

[222] R. Bostandoost, W. A. Hanafy, A. Lechowicz, N. Bashir, P. Shenoy, and M. Hajiesmaili, "Data-driven Algorithm Selection for Carbon-Aware Scheduling," in *Proceedings of the 3rd Workshop on Sustainable Computer Systems*, ser. HotCarbon '24, Santa Cruz, CA, USA, Jul. 2024. [Online]. Available: https://hotcarbon.org/assets/2024/pdf/hotcarbon24-final23.pdf.

[223] European Parliament and Council of the European Union. "Regulation (EU) 2016/679 of the European Parliament and of the Council," of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), [Online]. Available: https://data.europa.eu/eli/reg/2016/679/oj.

[224] NREL (National Renewable Energy Laboratory). "2024 Annual Technology Baseline," [Online]. Available: https://atb.nrel.gov/.

[225] Waymo LLC, *Waymo - Self-Driving Cars - Autonomous Vehicles - Ride-Hail*, Accessed 09-02-2024. [Online]. Available: https://waymo.com.

[226] P. R. Brown and F. M. O'Sullivan, "Spatial and Temporal Variation in the Value of Solar Power Across United States Electricity Markets," *Renewable and Sustainable Energy Reviews*, vol. 121, p. 109 594, 2020, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2019.109594. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032119308020.

[227] Los Angeles Department of Water and Power, *Residential Electric Rates - Los Angeles*, Accessed 09-02-2024. [Online]. Available: https://www.ladwp.com/account/understanding-your-rates/residential-electric-rates.

[228] V. Manshadi, S. Rodilitz, D. Saban, and A. Suresh, "Redesigning VolunteerMatch's Ranking Algorithm: Toward More Equitable Access to Volunteers," *SSRN Electronic Journal*, 2023, ISSN: 1556-5068. DOI: 10.2139/ssrn.4497747. [Online]. Available: http://dx.doi.org/10.2139/ssrn.4497747.

[229] C. R. Harris et al., "Array Programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[230] P. Virtanen et al., "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[231] A. R. Karlin, M. S. Manasse, L. A. McGeoch, and S. Owicki, "Competitive Randomized Algorithms for Nonuniform Problems," *Algorithmica*, vol. 11, no. 6, pp. 542–571, Jun. 1994, ISSN: 0178-4617, 1432-0541. DOI: 10.1007/BF01189993.

[232] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments*. Yale University Press, 1959, ISBN: 978-0-300-01372-6. JSTOR: j.ctt1bh4c8h.

[233] J. W. Pratt, "Risk Aversion in the Small and in the Large," *Econometrica*, vol. 32, no. 1/2, pp. 122–136, 1964, ISSN: 0012-9682. DOI: 10.2307/1913738. JSTOR: 1913738.

[234] J. Jia and J. S. Dyer, "A Standard Measure of Risk and Risk-Value Models," *Management Science*, vol. 42, no. 12, pp. 1691–1705, 1996, ISSN: 0025-1909. JSTOR: 2634547.

[235] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent Measures of Risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, Jul. 1999, ISSN: 0960-1627, 1467-9965. DOI: 10.1111/1467-9965.00068.

[236] G. Szegö, "Measures of Risk," *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1253–1272, Jul. 2002, ISSN: 0378-4266. DOI: 10.1016/S0378-4266(02)00262-5.

[237] R. T. Rockafellar and S. Uryasev, “Optimization of Conditional Value-at-Risk,” *The Journal of Risk*, vol. 2, no. 3, pp. 21–41, 2000, ISSN: 14651211. DOI: 10.21314/JOR.2000.038.

[238] R. T. Rockafellar and S. Uryasev, “Conditional Value-at-Risk for General Loss Distributions,” *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1443–1471, Jul. 2002, ISSN: 0378-4266. DOI: 10.1016/S0378-4266(02)00271-6.

[239] C. Acerbi and D. Tasche, “On the Coherence of Expected Shortfall,” *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1487–1503, Jul. 2002, ISSN: 0378-4266. DOI: 10.1016/S0378-4266(02)00283-2.

[240] P. Krokhmal, T. Uryasev, and J. Palmquist, “Portfolio Optimization with Conditional Value-at-Risk Objective and Constraints,” *The Journal of Risk*, vol. 4, no. 2, pp. 43–68, Mar. 2001, ISSN: 14651211. DOI: 10.21314/JOR.2002.057.

[241] Y. Chi and K. S. Tan, “Optimal Reinsurance under VaR and CVaR Risk Measures: A Simplified Approach,” *ASTIN Bulletin: The Journal of the International Actuarial Association*, vol. 41, no. 2, pp. 487–509, Nov. 2011, ISSN: 0515-0361, 1783-1350. DOI: 10.2143/AST.41.2.2136986.

[242] A. Hakobyan, G. C. Kim, and I. Yang, “Risk-Aware Motion Planning and Control Using CVaR-Constrained Optimization,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3924–3931, Oct. 2019, ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2019.2929980.

[243] R. Keramati, C. Dann, A. Tamkin, and E. Brunskill, “Being Optimistic to Be Conservative: Quickly Learning a CVaR Policy,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 4436–4443, Apr. 2020, ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v34i04.5870.

[244] A. Tamkin, C. Dann, R. Keramati, and E. Brunskill, “Distributionally-Aware Exploration for CVaR Bandits,” in *NeurIPS 2019 Workshop on Safety and Robustness on Decision Making*, 2019. [Online]. Available: https://rkeramati.github.io/assets/pdf/CVaR_Bandit.pdf.

[245] D. Baudry, R. Gautron, E. Kaufmann, and O. Maillard, “Optimal Thompson Sampling Strategies for Support-Aware CVaR Bandits,” in *Proceedings of the 38th International Conference on Machine Learning*, PMLR, Jul. 2021, pp. 716–726. [Online]. Available: https://proceedings.mlr.press/v139/baudry21a/baudry21a.pdf.

[246] G. Uziel and R. El-Yaniv, “Growth-Optimal Portfolio Selection under CVaR Constraints,” in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, PMLR, Mar. 2018, pp. 48–57. [Online]. Available: https://proceedings.mlr.press/v84/uziel18a.html.

[247] T. Soma and Y. Yoshida, "Online Risk-Averse Submodular Maximization," *Annals of Operations Research*, vol. 320, no. 1, pp. 393–414, Jan. 2023, ISSN: 1572-9338. DOI: 10.1007/s10479-022-04835-9.

[248] A. R. Karlin, C. Kenyon, and D. Randall, "Dynamic TCP Acknowledgement and Other Stories about e/(e-1)," in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, ser. STOC '01, New York, NY, USA: Association for Computing Machinery, Jul. 2001, pp. 502–509, ISBN: 978-1-58113-349-3. DOI: 10.1145/380752.380845.

[249] H. Zhang and V. Conitzer, "Combinatorial Ski Rental and Online Bipartite Matching," in *Proceedings of the 21st ACM Conference on Economics and Computation*, Virtual Event Hungary: ACM, Jul. 2020, pp. 879–910, ISBN: 978-1-4503-7975-5. DOI: 10.1145/3391403.3399470.

[250] S. Geulen, B. Vocking, and M. Winkler, "Regret Minimization for Online Buffering Problems Using the Weighted Majority Algorithm," in *23rd Annual Conference on Learning Theory*, 2010. [Online]. Available: https://www.learningtheory.org/colt2010/papers/105geulen.pdf.

[251] Q. Lin, H. Yi, J. Pang, M. Chen, A. Wierman, M. Honig, and Y. Xiao, "Competitive Online Optimization under Inventory Constraints," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 1, 10:1–10:28, Mar. 2019. DOI: 10.1145/3322205.3311081.

[252] D. Komm, R. Královič, R. Královič, and T. Mömke, "Randomized Online Computation with High Probability Guarantees," *Algorithmica*, vol. 84, no. 5, pp. 1357–1384, May 2022, ISSN: 1432-0541. DOI: 10.1007/s00453-022-00925-z.

[253] M. Dinitz, S. Im, T. Lavastida, B. Moseley, and S. Vassilvitskii, "Controlling Tail Risk in Online Ski-Rental," in *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2024, pp. 4247–4263. DOI: 10.1137/1.9781611977912.147.

[254] S. Sarykalin, G. Serraino, and S. Uryasev, "Value-at-Risk vs. Conditional Value-at-Risk in Risk Management and Optimization," in *State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, Z.-L. Chen, S. Raghavan, P. Gray, and H. J. Greenberg, Eds., INFORMS, Sep. 2008, pp. 270–294, ISBN: 978-1-877640-23-0. DOI: 10.1287/educ.1080.0052.

[255] G. Schäfer and N. Sivadasan, "Topology Matters: Smoothed Competitiveness of Metrical Task Systems," in *STACS 2004*, V. Diekert and M. Habib, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2004, pp. 489–500, ISBN: 978-3-540-24749-4. DOI: 10.1007/978-3-540-24749-4_43.

[256] N. Haghtalab, T. Roughgarden, and A. Shetty, "Smoothed Analysis with Adaptive Adversaries," in *2021 IEEE 62nd Annual Symposium on Founda-*

tions of Computer Science (FOCS), Feb. 2022, pp. 942–953. DOI: 10.1109/FOCS52979.2021.00095.

[257] H.-J. Böckenhauer, D. Komm, R. Královič, R. Královič, and T. Mömke, "Online Algorithms with Advice: The Tape Model," *Information and Computation*, vol. 254, pp. 59–83, Jun. 2017, ISSN: 0890-5401. DOI: 10.1016/j.ic.2017.03.001.

[258] Y. Emek, P. Fraigniaud, A. Korman, and A. Rosén, "Online Computation with Advice," *Theoretical Computer Science*, Selected Papers from 36th International Colloquium on Automata, Languages and Programming (ICALP 2009), vol. 412, no. 24, pp. 2642–2656, May 2011, ISSN: 0304-3975. DOI: 10.1016/j.tcs.2010.08.007.

[259] H. Witting, *Mathematische Statistik I*. Wiesbaden: Vieweg+Teubner Verlag, 1985, ISBN: 978-3-322-90151-4 978-3-322-90150-7. DOI: 10.1007/978-3-322-90150-7.

[260] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*. Society for Industrial and Applied Mathematics, Jan. 2009, ISBN: 978-0-89871-687-0 978-0-89871-875-1. DOI: 10.1137/1.9780898718751.

[261] J. C. Duchi and H. Namkoong, "Learning Models with Uniform Performance via Distributionally Robust Optimization," *The Annals of Statistics*, vol. 49, no. 3, Jun. 2021, ISSN: 0090-5364. DOI: 10.1214/20-AOS2004.

[262] A. Mafusalov, A. Shapiro, and S. Uryasev, "Estimation and Asymptotics for Buffered Probability of Exceedance," *European Journal of Operational Research*, vol. 270, no. 3, pp. 826–836, Nov. 2018, ISSN: 03772217. DOI: 10.1016/j.ejor.2018.01.021.

[263] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator, "Competitive Snoopy Caching," *Algorithmica*, vol. 3, no. 1, pp. 79–119, Nov. 1988, ISSN: 1432-0541. DOI: 10.1007/BF01762111.

[264] R. J. Aumann, "Mixed and Behavior Strategies in Infinite Extensive Games," in *Advances in Game Theory (AM-52)*, M. Dresher, L. S. Shapley, and A. W. Tucker, Eds., Princeton, NJ: Princeton University Press, 1964, pp. 627–650. JSTOR: j.ctt1b9x22w.31.

[265] R. Bellman and K. Cooke, *Differential-Difference Equations*. Santa Monica, CA: RAND Corporation, 1963. [Online]. Available: https://www.rand.org/content/dam/rand/pubs/reports/2006/R374.pdf.

[266] D. Hathaway, "Using Continuity Induction," *The College Mathematics Journal*, vol. 42, no. 3, pp. 229–231, May 2011, ISSN: 0746-8342, 1931-1346. DOI: 10.4169/college.math.j.42.3.229.

[267] C. Mathieu, *Online algorithms: Ski rental*, https://cs.brown.edu/~claire/Talks/skirental.pdf.

[268] D. S. Mitrinović, J. E. Pečarić, and A. M. Fink, *Classical and New Inequalities in Analysis*. Dordrecht: Springer Netherlands, 1993, ISBN: 978-90-481-4225-5 978-94-017-1043-5. DOI: 10.1007/978-94-017-1043-5.

[269] A. R. Chenreddy, N. Bandi, and E. Delage, "Data-Driven Conditional Robust Optimization," in *Advances in Neural Information Processing Systems*, vol. 35, Dec. 2022, pp. 9525–9537. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/3df874367ce2c43891aab1ab23ae6959-Abstract-Conference.html.

[270] Y. P. Patel, S. Rayan, and A. Tewari, "Conformal Contextual Robust Optimization," in *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, PMLR, Apr. 2024, pp. 2485–2493. [Online]. Available: https://proceedings.mlr.press/v238/p-patel24a.html.

[271] I. Wang, C. Becker, B. Van Parys, and B. Stellato, *Learning Decision-Focused Uncertainty Sets in Robust Optimization*, Jul. 2024. DOI: 10.48550/arXiv.2305.19225. [Online]. Available: http://arxiv.org/abs/2305.19225.

[272] A. Chenreddy and E. Delage, "End-to-end Conditional Robust Optimization," in *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, Jul. 2024, pp. 736–748. [Online]. Available: https://openreview.net/forum?id=Oe9ngGi8Gh.

[273] X. Yan, C. Gu, X. Zhang, and F. Li, "Robust Optimization-Based Energy Storage Operation for System Congestion Management," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2694–2702, Jun. 2020, ISSN: 1932-8184, 1937-9234, 2373-7816. DOI: 10.1109/JSYST.2019.2932897. [Online]. Available: https://ieeexplore.ieee.org/document/8805255/.

[274] S. S. Parvar and H. Nazaripouya, "Optimal Operation of Battery Energy Storage Under Uncertainty Using Data-Driven Distributionally Robust Optimization," *Electric Power Systems Research*, vol. 211, p. 108 180, Oct. 2022, ISSN: 0378-7796. DOI: 10.1016/j.epsr.2022.108180. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779622003911.

[275] C. Gregory, K. Darby-Dowman, and G. Mitra, "Robust Optimization and Portfolio Selection: The Cost of Robustness," *European Journal of Operational Research*, vol. 212, no. 2, pp. 417–428, Jul. 2011, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2011.02.015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221711001469.

[276] A. G. Quaranta and A. Zaffaroni, "Robust Optimization of Conditional Value at Risk and Portfolio Selection," *Journal of Banking & Finance*, vol. 32, no. 10, pp. 2046–2056, Oct. 2008, ISSN: 0378-4266. DOI: 10.1016/j.jbankfin.2007.12.025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378426607004281.

[277] G. Shafer and V. Vovk, "A Tutorial on Conformal Prediction," *Journal of Machine Learning Research*, vol. 9, no. 12, pp. 371–421, 2008. [Online]. Available: http://jmlr.org/papers/v9/shafer08a.html.

[278] Y. Chen, Y. Shi, and B. Zhang, "Optimal Control Via Neural Networks: A Convex Approach," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=H1MW72AcK7.

[279] D. Stutz, Krishnamurthy, Dvijotham, A. T. Cemgil, and A. Doucet, *Learning Optimal Conformal Classifiers*, May 2022. DOI: 10.48550/arXiv.2110.09192. [Online]. Available: http://arxiv.org/abs/2110.09192.

[280] D. Bertsimas and A. Thiele, "Robust and Data-Driven Optimization: Modern Decision Making Under Uncertainty," in *Models, Methods, and Applications for Innovative Decision Making*, ser. INFORMS TutORials in Operations Research, Section: 4, INFORMS, Sep. 2006, pp. 95–122, ISBN: 978-1-877640-20-9. DOI: 10.1287/educ.1063.0022. [Online]. Available: https://pubsonline.informs.org/doi/abs/10.1287/educ.1063.0022.

[281] D. Bertsimas, V. Gupta, and N. Kallus, "Data-driven Robust Optimization," *Mathematical Programming*, vol. 167, no. 2, pp. 235–292, Feb. 2018, ISSN: 1436-4646. DOI: 10.1007/s10107-017-1125-8. [Online]. Available: https://doi.org/10.1007/s10107-017-1125-8.

[282] P. Alexeenko and E. Bitar, "Nonparametric Estimation of Uncertainty Sets for Robust Optimization," in *2020 59th IEEE Conference on Decision and Control (CDC)*, Jeju, Korea (South): IEEE, Dec. 2020, pp. 1196–1203, ISBN: 978-1-72817-447-1. DOI: 10.1109/CDC42340.2020.9303863. [Online]. Available: https://ieeexplore.ieee.org/document/9303863/.

[283] M. Goerigk and J. Kurtz, "Data-driven Robust Optimization Using Deep Neural Networks," *Computers & Operations Research*, vol. 151, p. 106 087, Mar. 2023, ISSN: 0305-0548. DOI: 10.1016/j.cor.2022.106087. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054822003173.

[284] A. N. Elmachtoub, H. Lam, H. Zhang, and Y. Zhao, *Estimate-Then-Optimize versus Integrated-Estimation-Optimization versus Sample Average Approximation: A Stochastic Dominance Perspective*, Aug. 2023. DOI: 10.48550/arXiv.2304.06833. [Online]. Available: http://arxiv.org/abs/2304.06833.

[285] C. Johnstone and B. Cox, "Conformal Uncertainty Sets for Robust Optimization," in *Proceedings of the Tenth Symposium on Conformal and Probabilistic Prediction and Applications*, PMLR, Sep. 2021, pp. 72–90. [Online]. Available: https://proceedings.mlr.press/v152/johnstone21a.html.

[286] C. Sun, L. Liu, and X. Li, "Predict-then-Calibrate: A New Perspective of Robust Contextual LP," Nov. 2023. [Online]. Available: https://openreview.net/forum?id=TnTDiCppx5&referrer=%5Bthe%20profile%20of%20Chunlin%20Sun%5D(%2Fprofile%3Fid%3D~Chunlin_Sun1).

[287] B. Amos, L. Xu, and J. Z. Kolter, "Input Convex Neural Networks," in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, Jul. 2017, pp. 146–155. [Online]. Available: https://proceedings.mlr.press/v70/amos17b.html.

[288] Y. Romano, E. Patterson, and E. Candes, "Conformalized Quantile Regression," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Available: http://papers.neurips.cc/paper/8613-conformalized-quantile-regression.pdf.

[289] B. Amos and J. Z. Kolter, "OptNet: Differentiable Optimization as a Layer in Neural Networks," in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, Jul. 2017, pp. 136–145. [Online]. Available: https://proceedings.mlr.press/v70/amos17a.html.

[290] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable Convex Optimization Layers," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/9ce3c52fc54362e22053399d3181c638-Abstract.html.

[291] P. L. Donti, B. Amos, and J. Z. Kolter, "Task-based End-to-end Model Learning in Stochastic Optimization," in *Advances in Neural Information Processing Systems*, vol. 30, Long Beach, CA, USA: Curran Associates, Inc., Dec. 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3fc2c60b5782f641f76bcefc39fb2392-Paper.pdf.

[292] A. Lin and D. E. Ba, "How to Train Your FALCON: Learning Log-Concave Densities with Energy-Based Neural Networks," Jul. 2023. [Online]. Available: https://openreview.net/forum?id=UP1r5-t-ov.

[293] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Networks," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning

Research, vol. 37, PMLR, May 2015, pp. 1613–1622. [Online]. Available: https://proceedings.mlr.press/v37/blundell15.pdf.

[294] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *arXiv:1506.02142 [cs, stat]*, Oct. 2016. [Online]. Available: http://arxiv.org/abs/1506.02142.

[295] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, Nov. 2005, ISBN: 978-0-262-25683-4. DOI: 10.7551/mitpress/3206.001.0001. [Online]. Available: https://doi.org/10.7551/mitpress/3206.001.0001.

[296] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep Kernel Learning," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, A. Gretton and C. C. Robert, Eds., Cadiz, Spain: PMLR, May 2016, pp. 370–378. [Online]. Available: http://proceedings.mlr.press/v51/wilson16.html.

[297] J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan, "Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness," vol. 33, Curran Associates, Inc., 2020, pp. 7498–7512. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/543e83748234f7cbab21aa0ade66565f-Abstract.html.

[298] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

[299] Z. Nado et al., "Uncertainty Baselines: Benchmarks for Uncertainty & Robustness in Deep Learning," Jan. 2022. DOI: 10.48550/arXiv.2106.04015. [Online]. Available: http://arxiv.org/abs/2106.04015.

[300] B.-S. Einbinder, Y. Romano, M. Sesia, and Y. Zhou, "Training Uncertainty-Aware Classifiers with Conformalized Deep Learning," in *Advances in Neural Information Processing Systems*, New Orleans, LA, USA, Nov. 2022. DOI: 10.48550/arXiv.2205.05878. [Online]. Available: https://papers.neurips.cc/paper_files/paper/2022/file/8c96b559340daa7bb29f56ccfbbc9c2f-Paper-Conference.pdf.

[301] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton University Press, 2009, ISBN: 978-0-691-14368-2. [Online]. Available: https://www.jstor.org/stable/j.ctt7sk8p.

[302] A. Nemirovski and A. Shapiro, "Convex Approximations of Chance Constrained Programs," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, Jan. 2007, ISSN: 1052-6234, 1095-7189. DOI: 10.1137/050622328.

[Online]. Available: http://epubs.siam.org/doi/10.1137/050622328.

[303] H. Rahimian and S. Mehrotra, "Distributionally Robust Optimization: A Review," *arXiv:1908.05659 [cs, math, stat]*, Aug. 2019, arXiv: 1908.05659. [Online]. Available: http://arxiv.org/abs/1908.05659.

[304] Q. P. Zheng, J. Wang, and A. L. Liu, "Stochastic Optimization for Unit Commitment—A Review," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1913–1924, Jul. 2015, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2014.2355204. [Online]. Available: https://ieeexplore.ieee.org/document/6912028.

[305] W. Zhong, K. Xie, Y. Liu, S. Xie, and L. Xie, "Chance Constrained Scheduling and Pricing for Multi-Service Battery Energy Storage," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 5030–5042, Nov. 2021, ISSN: 1949-3061. DOI: 10.1109/TSG.2021.3109140. [Online]. Available: https://ieeexplore.ieee.org/document/9525453.

[306] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, San Diego, CA, USA, May 2015. [Online]. Available: http://arxiv.org/abs/1412.6980.

[307] W. W. Hogan, "Electricity Market Design: Multi-Interval Pricing Models," Accessed 2023-03-23. [Online]. Available: https://scholar.harvard.edu/files/whogan/files/hogan_hepg_multi_period_062220.pdf.

[308] E. Ela and M. O'Malley, "Scheduling and Pricing for Expected Ramp Capability in Real-Time Power Markets," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 1681–1691, May 2016. DOI: 10.1109/TPWRS.2015.2461535.

[309] E. Ela, M. Milligan, and B. Kirby, "Operating Reserves and Variable Generation," National Renewable Energy Laboratory, TP-5500-51978, Aug. 2011. [Online]. Available: https://www.nrel.gov/docs/fy11osti/51978.pdf.

[310] P. Cramton, A. Ockenfels, and S. Stoft, "Capacity Market Fundamentals," *Economics of Energy & Environmental Policy*, vol. 2, no. 2, Apr. 2013, ISSN: 21605882. DOI: 10.5547/2160-5890.2.2.2. [Online]. Available: http://www.iaee.org/en/publications/eeeparticle.aspx?id=46.

[311] D. Munoz-Alvarez, E. Bitar, L. Tong, and J. Wang, "Piecewise Affine Dispatch Policies for Economic Dispatch Under Uncertainty," in *2014 IEEE PES General Meeting | Conference & Exposition*, National Harbor, MD, USA: IEEE, Jul. 2014, pp. 1–5, ISBN: 978-1-4799-6415-4. DOI: 10.1109/PESGM.2014.6939369. [Online]. Available: http://ieeexplore.ieee.org/document/6939369/.

[312] M. S. Modarresi, L. Xie, M. C. Campi, S. Garatti, A. Carè, A. A. Thatte, and P. R. Kumar, "Scenario-Based Economic Dispatch With Tunable Risk Levels in High-Renewable Power Systems," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 5103–5114, Nov. 2019. DOI: 10.1109/TPWRS.2018.2874464.

[313] Y. Guo, C. Chen, and L. Tong, "Pricing Multi-Interval Dispatch under Uncertainty Part I: Dispatch-Following Incentives," *arXiv:1911.05784 [cs, eess]*, arXiv: 1911.05784. [Online]. Available: http://arxiv.org/abs/1911.05784.

[314] C. Chen, Y. Guo, and L. Tong, "Pricing Multi-Interval Dispatch under Uncertainty Part II: Generalization and Performance," *arXiv:1912.13469 [cs, eess]*, arXiv: 1912.13469. [Online]. Available: http://arxiv.org/abs/1912.13469.

[315] B. Hua, D. A. Schiro, T. Zheng, R. Baldick, and E. Litvinov, "Pricing in Multi-Interval Real-Time Markets," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2696–2705, Jul. 2019, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2019.2891541. [Online]. Available: https://ieeexplore.ieee.org/document/8606259/.

[316] J. Zhao, T. Zheng, and E. Litvinov, "A Multi-Period Market Design for Markets With Intertemporal Constraints," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 3015–3025, Jul. 2020, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2019.2963022. [Online]. Available: https://ieeexplore.ieee.org/document/8945389/.

[317] X. Kuang, Y. Dvorkin, A. J. Lamadrid, M. A. Ortega-Vazquez, and L. F. Zuluaga, "Pricing Chance Constraints in Electricity Markets," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4634–4636, Jul. 2018, ISSN: 1558-0679. DOI: 10.1109/TEC.2018.2821922.

[318] R. Mieth and Y. Dvorkin, "Distribution Electricity Pricing Under Uncertainty," *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 2325–2338, May 2020, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2019.2954971.

[319] R. P. O'Neill, P. M. Sotkiewicz, B. F. Hobbs, M. H. Rothkopf, and W. R. Stewart, "Efficient Market-Clearing Prices in Markets with Nonconvexities," *European Journal of Operational Research*, vol. 164, no. 1, pp. 269–285, Jul. 2005, ISSN: 03772217. DOI: 10.1016/j.ejor.2003.12.011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0377221703009196.

[320] L. Xie and M. D. Ilic, "Model Predictive Economic/Environmental Dispatch of Power Systems with Intermittent Resources," in *2009 IEEE Power & Energy Society General Meeting*, Jul. 2009, pp. 1–6. DOI: 10.1109/PES.2009.5275940.

[321] J. Luedtke and S. Ahmed, “A Sample Approximation Approach for Optimization with Probabilistic Constraints,” *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 674–699, Jan. 2008, ISSN: 1052-6234, 1095-7189. DOI: 10.1137/070702928. [Online]. Available: http://epubs.siam.org/doi/10.1137/070702928.

[322] M. Lubin, Y. Dvorkin, and S. Backhaus, “A Robust Approach to Chance Constrained Optimal Power Flow With Renewable Generation,” *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3840–3849, Sep. 2016, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2015.2499753.

[323] *California ISO - Today's Outlook*. [Online]. Available: http://www.caiso.com/TodaysOutlook/Pages/default.aspx.

[324] M. K. Enns, J. J. Quada, and B. Sackett, “Fast Linear Contingency Analysis,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, no. 4, pp. 783–791, Apr. 1982, ISSN: 0018-9510. DOI: 10.1109/TPAS.1982.317142.

[325] J. W. Busby, K. Baker, M. D. Bazilian, A. Q. Gilbert, E. Grubert, V. Rai, J. D. Rhodes, S. Shidore, C. A. Smith, and M. E. Webber, “Cascading risks: Understanding the 2021 Winter Blackout in Texas,” *Energy Research & Social Science*, vol. 77, p. 102 106, Jul. 2021, ISSN: 2214-6296. DOI: 10.1016/j.erss.2021.102106. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214629621001997.

[326] *NERC Reliability Standard PRC-003-0a*, 2010. [Online]. Available: https://www.nerc.com/pa/Stand/Reliability%20Standards/TPL-003-0a.pdf.

[327] U.S.-Canada Power System Outage Task Force, *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*, 2004. [Online]. Available: https://www.nerc.com/pa/rrm/ea/2003%5C%20Blackout%5C%20Final%5C%20Report/ch1-3.pdf.

[328] Jiachun Guo, Yong Fu, Zuyi Li, and M. Shahidehpour, “Direct Calculation of Line Outage Distribution Factors,” *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1633–1634, Aug. 2009, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2009.2023273. [Online]. Available: http://ieeexplore.ieee.org/document/5130240/.

[329] P. A. Kaplunovich and K. S. Turitsyn, “Fast Selection of N-2 Contingencies for Online Security Assessment,” in *2013 IEEE Power & Energy Society General Meeting*, 2013, pp. 1–5. DOI: 10.1109/PESMG.2013.6672792. [Online]. Available: https://ieeexplore.ieee.org/document/6672792.

[330] P. Kaplunovich and K. Turitsyn, “Fast and Reliable Screening of N-2 Contingencies,” *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4243–4252, Nov. 2016, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2015.2510586.

[331] C. M. Davis and T. J. Overbye, “Multiple Element Contingency Screening,” *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1294–1301, Aug. 2011, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2010.2087366.

[332] Y. Yang, X. Guan, and Q. Zhai, “Fast Grid Security Assessment With N - k Contingencies,” *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 2193–2203, May 2017, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2016.2608378.

[333] K. Dvijotham and K. Turitsyn, *Construction of Power Flow Feasibility Sets*, arXiv:1506.07191 [cs], Jul. 2015. [Online]. Available: http://arxiv.org/abs/1506.07191.

[334] H. D. Nguyen, K. Dvijotham, and K. Turitsyn, “Constructing Convex Inner Approximations of Steady-State Security Regions,” *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 257–267, Jan. 2019, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2018.2868752. [Online]. Available: https://ieeexplore.ieee.org/document/8454506/.

[335] D. Lee, K. Turitsyn, D. K. Molzahn, and L. A. Roald, “Robust AC Optimal Power Flow With Robust Convex Restriction,” *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 4953–4966, Nov. 2021, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2021.3075925. [Online]. Available: https://ieeexplore.ieee.org/document/9416854/.

[336] F. A. Taha, T. Vincent, and E. Bitar, “An Efficient Method for Quantifying the Aggregate Flexibility of Plug-In Electric Vehicle Populations,” *IEEE Transactions on Smart Grid*, pp. 1–1, 2024, ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2024.3384871. [Online]. Available: https://ieeexplore.ieee.org/document/10490133/.

[337] C. Dawson and C. Fan, *Adversarial Optimization Leads to Over-Optimistic Security-Constrained Dispatch, but Sampling Can Help*, Oct. 2023. [Online]. Available: http://arxiv.org/abs/2310.06956.

[338] Y. Chen, Y. Shi, and B. Zhang, “Data-Driven Optimal Voltage Regulation Using Input Convex Neural Networks,” *Electric Power Systems Research*, vol. 189, p. 106 741, Dec. 2020, ISSN: 0378-7796. DOI: 10.1016/j.epsr.2020.106741. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779620305447.

[339] A. Rosemberg, M. Tanneau, B. Fanzeres, J. Garcia, and P. Van Hentenryck, “Learning Optimal Power Flow Value Functions with Input-Convex Neural Networks,” *Electric Power Systems Research*, vol. 235, p. 110 643, Oct. 2024, ISSN: 0378-7796. DOI: 10.1016/j.epsr.2024.110643. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779624005297.

[340] C. Elkan, "The Foundations of Cost-Sensitive Learning," in *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2*, ser. IJCAI'01, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Aug. 2001, pp. 973–978, ISBN: 978-1-55860-812-2.

[341] S. Viaene and G. Dedene, "Cost-Sensitive Learning and Decision Making Revisited," *European Journal of Operational Research*, Metaheuristics and Worst-Case Guarantee Algorithms: Relations, Provable Properties and Applications, vol. 166, no. 1, pp. 212–220, Oct. 2005, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2004.03.031. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221704002978.

[342] K. Dvijotham, R. Stanforth, S. Gowal, T. Mann, and P. Kohli, *A Dual Approach to Scalable Verification of Deep Networks*, Aug. 2018. DOI: 10.48550/arXiv.1803.06567. [Online]. Available: http://arxiv.org/abs/1803.06567.

[343] B. G. Anderson, Z. Ma, J. Li, and S. Sojoudi, "Tightened Convex Relaxations for Neural Network Robustness Certification," in *2020 59th IEEE Conference on Decision and Control (CDC)*, Jeju, Korea (South): IEEE, Dec. 2020, pp. 2190–2197, ISBN: 978-1-72817-447-1. DOI: 10.1109/CDC42340.2020.9303750. [Online]. Available: https://ieeexplore.ieee.org/document/9303750/.

[344] Z. ul Abdeen, H. Yin, V. Kekatos, and M. Jin, "Learning Neural Networks under Input-Output Specifications," in *2022 American Control Conference (ACC)*, Jun. 2022, pp. 1515–1520. DOI: 10.23919/ACC53348.2022.9867571.

[345] B. Stott, J. Jardim, and O. Alsac, "DC Power Flow Revisited," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290–1300, Aug. 2009, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2009.2021235. [Online]. Available: https://ieeexplore.ieee.org/document/4956966/?arnumber=4956966.

[346] B. C. Eaves and R. M. Freund, "Optimal Scaling of Balls and Polyhedra," *Mathematical Programming*, vol. 23, no. 1, pp. 138–147, Dec. 1982, ISSN: 0025-5610, 1436-4646. DOI: 10.1007/BF01583784. [Online]. Available: http://link.springer.com/10.1007/BF01583784.

[347] C. Yeh*, N. Christianson*, A. Wu, A. Wierman, and Y. Yue. "End-to-End Conformal Calibration for Optimization Under Uncertainty." arXiv: 2409.20534 [cs, math], [Online]. Available: http://arxiv.org/abs/2409.20534.

[348] T. Athay, R. Podmore, and S. Virmani, "A Practical Method for the Direct Analysis of Transient Stability," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-98, no. 2, pp. 573–584, Mar. 1979, ISSN: 0018-

9510. DOI: 10.1109/TPAS.1979.319407. [Online]. Available: https://ieeexplore.ieee.org/document/4113518.

[349] M. A. Pai, *Energy Function Analysis for Power System Stability*. Boston, MA: Springer US, 1989, ISBN: 978-1-4612-8903-6 978-1-4613-1635-0. DOI: 10.1007/978-1-4613-1635-0. [Online]. Available: http://link.springer.com/10.1007/978-1-4613-1635-0.

[350] L. Thurner, A. Scheidler, F. Schäfer, J.-H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "Pandapower—An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2018.2829021. [Online]. Available: https://ieeexplore.ieee.org/document/8344496.

[351] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 721, Red Hook, NY, USA: Curran Associates Inc., Dec. 2019, pp. 8026–8037.