Optimizing Cloud AI Platforms: Resource Allocation and Market Design

Thesis by
Yu Su

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

Caltech

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2021
Defended May 25th, 2021

Yu Su
ORCID: 0000-0002-7159-4542

ACKNOWLEDGEMENTS

I am fortunate to have had the opportunity to pursue my Ph.D. at Caltech. During my time in Pasadena, I was so happy to interact and engage with many lovely scholars. My journey would not have been so enjoyable without them.

First and foremost, I would like to thank my advisor - Adam Wierman. I still remember our very first meeting: I felt instantaneously excited to talk with him about various topics - research projects and even beyond. This short conversation was representative of my entire Ph.D. study under Adam's guidance. Not only have I learned a lot about tackling various challenging research problems, but I have also learned so much beyond research skills. I truly admire his integrity and appreciate the inspiration that I have received from him. I hope that I can pass on the inspiration to others who aim to aspire as well during my future career. I could not ask for a better advisor. Adam, thank you for the freedom, trust, and all the interesting conversations.

Next, I would like to thank my committee members: Yisong Yue, Steven Low, and Pietro Perona, for the time they dedicated to provide valuable feedback on this thesis.

I want to express my sincere gratitude to my wonderful collaborators and friends: Anatoly Khina, Xiaoqi Ren, Shai Vardi, Navid Azizan, and Juba Ziani. Besides, I would like to thank my fellow friends within and beyond Caltech. I will not dive into the friends list so that no one is left out. However, if you happen to be aware of a better solution, please do not let me know because I might not expect to write this acknowledgement again any time soon.

My special thanks go to Toli, Xiaoqi, Shai, and Brennan. Toli, thank you for your guidance and support during my first two years when I arrived at Caltech. Xiaoqi, thank you for everything! Shai, thank you for keeping telling all the jokes even though I do not laugh every time. Moreover, I truly admire your integrity and honesty. I wish I had more opportunities to work with you than I did. Brennan, thank you for all the time that we spent together on those long problems sets as well as those adventures in LA.

Finally, I would like to thank my parents and Jinglin. This thesis would not have been possible without their love and support.

Thank you for taking your time to read this entire acknowledgement. I am a bit surprised that you actually reached this far. As a token of my gratitude, please let

me buy you a cup of coffee. To redeem the offer, please send me an email with the coupon code "TEA" before we meet next time.

ABSTRACT

The numerous applications of data-driven algorithms and tools across diverse industries have led to tremendous successes in recent years. As the volume of massive data that is created, collected, and consumed continues to grow, there are many new imposed challenges faced by today's cloud AI platforms that support the deployment of machine learning algorithms on a large scale. In this thesis, we tackle the emerging challenges within cloud AI systems and beyond by adopting approaches from the fields of resource allocation and market design.

First, we propose a new scheduler, Generalized Earliest Time First (GETF), and provide the provable, worst-case approximation guarantees for the goals of minimizing both makespan and total weighted completion time of tasks with precedence constraints on related machines with machine-dependent communication times. These two results address long-standing open problems. Further, we adopt the classic speed scaling function to model power consumption and use mean response time to measure the performance. We propose the concept of pseudo-size to quantify importance of tasks and design a family of two-stage scheduling frameworks based on the approximation of pseudo-size. Assuming a good approximation of pseudo-size, we are able to provide the first provable bound of a linear combination of performance and energy goals under this setting.

Second, we study the design of mechanisms for data acquisition in settings with information leakage and verifiable data. We provide the first characterization of an optimal mechanism for data acquisition if agents are concerned about privacy and their data is correlated with each other. Additionally, the mechanism allows, for the first time, a trade-off between the bias and variance of the estimator. Transitioning from the data market into the energy market, we propose a new pricing scheme, which is applicable to general non-convex costs, and allows using general parametric pricing functions. Optimizing for the quantities and the price parameters simultaneously, and the ability to use general parametric pricing functions allows our scheme to find prices that are typically economically more efficient and less discriminatory than those of the existing schemes while still supporting a competitive equilibrium. In addition, we supplement the proposed method with a computationally efficient polynomial-time approximation algorithm, which can be used to approximate the optimal quantities and prices for general non-convex cost functions.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Y. Su, X. Ren, S. Vardi, A. Wierman, and Y. He, “Communication-aware scheduling of precedence-constrained tasks,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 47, no. 2, pp. 21–23, 2019. DOI: 10.1145/3374888.3374897,
Adapted into Chapter 2 of this thesis. Y. Su contributed to the conception of the project, proposing the model and analyzing its performance, and writing the manuscript.

[2] Y. Su, X. Ren, S. Vardi, and A. Wierman, “Communication-aware scheduling of precedence-constrained tasks on related machines,” *Under Review*, 2020,
Adapted into Chapter 2 of this thesis. Y. Su contributed to the conception of the project, proposing the model and analyzing its performance, and writing the manuscript.

[3] Y. Su, J. Yu, V. Anand, and A. Wierman, “Learning-augmented energy-aware scheduling of precedence-constrained tasks,” *Under Review*, 2021,
Adapted into Chapter 3 of this thesis. Y. Su contributed to the conception of the project, proposing the model and analyzing its performance, and writing the manuscript.

[4] G. Liao*, Y. Su*, J. Ziani, A. Wierman, and J. Huang, “The privacy paradox and optimal bias-variance trade-offs in data acquisition,” in *Proceedings of the 2021 ACM Conference on Economics and Computation*, 2021,
Adapted into Chapter 4 of this thesis. Y. Su contributed to the conception of the project, proposing the model and analyzing its performance, and writing the manuscript.

[5] N. Azizan, Y. Su, K. Dvijotham, and A. Wierman, “Optimal pricing in markets with nonconvex costs,” *Operations Research*, vol. 68, no. 2, pp. 480–496, 2020. DOI: 10.1287/opre.2019.1900,
Adapted into Chapter 5 of this thesis. Y. Su contributed to the conception of the project, proposing the model and analyzing its performance, and writing the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

In the last decade, rapid technology trends have driven researchers and engineers to reevaluate the large scale computing system and infrastructure in use today, i.e., cloud platforms and colocation data centers behind the cloud. One important trend is the ongoing digital transformation in almost every industry, spanning from autonomous self-driving, customized recommendations to individual consumers, monitoring individual health conditions via smart personal wearable, to improving crops yields and sustainability in farming. These amazing applications across diverse industries dramatically improve efficiency and help practitioners quickly adapt to new business models.

A key component of digital transformation is the successful deployment of data-driven research and applications, in particular *Machine Learning (ML)* and broadly *Artificial Intelligence (AI)* tools, in both academia and industry. While almost every industry is embracing the AI cloud era, enormous data has been created and collected in the pipeline. The tremendous amount of data has introduced new characteristics to the system internally, and thus has imposed many new challenges to the design of the large scale cloud AI system. Beyond, to assimilate to the internal growth of the cloud AI systems, the external stakeholders that interact with the cloud AI system are undergoing a swift transformation to keep up with these new changes as well. Regardless of the internal or external challenges, these new issues have motivated researchers and engineers to redesign the cloud AI system and infrastructure to adapt to the new trends.

In the following, we first introduce the new rising trends of the evolving technology and highlight the challenges faced by today's cloud AI systems as well as the external stakeholders. Finally, we give an overview of the works presented in this thesis in order to tackle some of these new rising challenges on the cloud and even beyond.

1.1 Trends and Challenges

More and more data has been created, captured, copied, and consumed every day. According to an International Data Corporation (IDC) report, the amount of world's collective data has been growing almost 50x from 1.2 zettabytes [1] (ZB) in 2010 to 59 ZB in 2020, at a ten-year compound annual growth rate (CAGR) of 48%. The staggering number is predicted to continue to grow to reach 175 ZB by 2025 [1]. The massive data serves as the catalyst to the successful development of various data-driven methods and tools, such as *Deep Learning* (DL), *Computer Vision* (CV), *Natural Language Processing* (NLP), etc. These powerful tools have the capability to learn from data without being explicitly programmed, and they are generally referred to as AI technology. However, as the complexity of AI tools increases, the development of AI technology and harnessing the power of these tools have become more and more challenging for small and individual players. Thus the cloud AI platforms, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud, act as an important medium to promote the use of AI technology. To democratize AI, i.e., make AI accessible and affordable to the general enterprises, small business as well as individuals, there are many challenges to the design of the general system as the massive amount of data continues to evolve and grow.

Though these challenges come in many different forms, they can be roughly divided into two categories from the standpoint of the cloud AI system: internal and external challenges. The internal challenges fall into the domain of *resource allocation* while we approach the external challenges from the perspective of *market design*. In the following, we introduce the new trends and challenges from these two angles, respectively.

Resource Allocation. Within the cloud AI system, diverse AI tools have been applied to practical settings, and they are often computationally expensive. For instance, the widespread use and astonishing achievements of deep learning in image recognition, machine translation, etc. (refer to [2] for a recent survey of deep learning applications), is significantly reliant on the advance of computing power of hardware in the past years. A recent study indicates that progress along the current lines is rapidly becoming economically, technically, and environmentally unsustainable [3]. Thus, the overall performance highly relies on how we can efficiently allocate the computing resources to accelerate the computation. Unlike traditional independent jobs, machine learning jobs come with their own unique characteristics, such as precedence constraints, non-uniform communication delays, intensive power consumption, etc. The development of new algorithms and tools are indispensable for the system to suit and incorporate these new characteristics.

[1] 1 ZB is equal to 10^{21} bytes, that is a trillion gigabytes (GB).

Market Design. As the integration of cloud AI platforms into infrastructure advances, its close interactions with the external stakeholders have brought new challenges beyond the cloud AI system. We approach these external challenges from the perspective of market design, and aim to understand the interactions of market participants. First, in order for the cloud AI platforms to prosper, a key driving force is to gather and collect data of high quality to feed and empower the algorithms to learn and evolve. The data market, as a channel for platforms and data providers to interact with each other, plays a significant role to facilitate the collection of high quality data sets. However, growing privacy concerns have imposed many new challenges. Thus, the study of impacts of privacy concerns on the mechanism design for data acquisition becomes a key enabler to the entire pipeline. Second, while the integration of the cloud AI systems into the broad system and infrastructure has been a roadblock, it also presents some new opportunities. The flexible loads in data centers not only provide freedom to optimize over energy efficiency, but also add flexibility to the entire electricity grid. Further, on an individual level, electricity bills for a data center has been on the order of millions of dollars. Thus understanding the market design of the energy market has become an important piece for the entire system and infrastructure as well as individual players.

Next, we take a further dive and frame the concrete goals to tackle the new challenges from the perspectives of resource allocation and market design.

1.1.1 From the Perspective of Resource Allocation

One challenge towards democratizing AI is to improve computational efficiency of the large scale general purpose machine learning platforms, such as Google's TensorFlow [4], Facebook's PyTorch [5], and Microsoft's Azure Machine Learning (AzureML) [6]. These emerging machine learning jobs are often expressed as computational graphs with precedence constraints and they come with gigantic sizes, which requires a large cluster of geo-distributed heterogeneous machines to collaborate and compute in a distributed way. Thus, how to schedule these machine learning jobs in an efficient manner is critical to the performance of ML platforms. Without a good scheduler, it is impossible to take advantage of large clusters of computational resources to adapt to the exponential growth of big data.

The heterogeneity of the cloud AI system makes scheduling ML jobs extremely hard. The AI cloud system is heterogeneous in two key ways. First, a modern computing system always consists of a mixture of heterogeneous machines, such as CPUs, GPUs, TPUs, etc, to build up its capability in dealing with various tasks under different scenarios. These computing machines are characterized by different processing speeds to represent differences in types or machine models. Second, for various deep learning tasks, the communication overhead required for data transfer from one machine to another is significant, and highly limits the scalability of the system [7, 8]. Meanwhile, the communication speeds are nonuniform in the sense that they are machine-dependent to capture the differences in geographic locations and network bandwidth. These newly introduced characteristics of the cloud AI system call for new algorithms for such a resource allocation problem. However, the prior works have been focused on extensions to settings when communication delays are ignored [9, 10] or assumed to be uniform (fixed) [11, 12], and fail to incorporate nonuniform machine dependent communication delays. This naturally leads to the question: *can we design an efficient scheduler for precedence-constrained ML jobs with nonuniform communication delays?*

Another rising challenge is that these machine learning jobs are often energy-intensive. For example, the emissions of training an AI model can be as high as five times the lifetime emission of a car [13]. In fact, the computations required for deep learning have been doubling every 3.4 months, resulting a 300,000x increase from 2012 to 2018 [14, 15]. Thus it is urgent to study how we can design good schedulers for these machine learning jobs with both performance and energy metrics in mind. Moreover, due to the aggressive development of power management technology, power usage effectiveness (PUE), a metric representing the ratio of total amount of energy used by a computer data center facility to the energy delivered to computing equipment, has been dramatically decreasing since the early 2000s. This means that efforts should be focusing on directly optimizing over IT energy consumption to further improve the overall power consumption for data centers. Even a small improvement on the level of individual computing machines aggregates into a big step towards the idea of sustainable computing. However, even scheduling ML jobs without energy considerations has been extremely hard. *Can we design an efficient scheduler for precedence-constrained ML jobs with both performance and energy goals in mind?*

1.1.2 From the Perspective of Market Design

As the integration of cloud AI systems continues, we have to investigate and understand its close interactions with the external factors in order to support the long-term success of the overall system. In particular, we approach these issues surrounding cloud AI systems, i.e., data centers in the physical world, from the angle of market design, with an emphasis on two distinct parts: *data markets* and *energy markets*.

A key element to the prosperity of these online platforms is the success of large scale data-driven technology and tools. A prerequisite for this success to continue is to guarantee that online platforms have the capability to gather and acquire data as desired while satisfying data providers – individuals. Online platforms gather and collect the massive amount of data on billions of individuals to improve advertisements and their systems. While more and more data is being created every day, individuals have more and more privacy concerns about the use of their personal data. These privacy concerns brings another challenge to the design of data markets. To facilitate the interactions between online platforms and individuals, it is crucial to study the data market when privacy concerns are considered. This leads to a natural question: *how do information leakage and privacy concerns impact the design of data markets?*

The data centers that accommodate the cloud AI system consume a massive amount of energy. This not only brings a challenging resource allocation problem, but also plays an important role and presents new opportunities in the process as these huge data centers start to integrate into the electricity grid [16]. The connections between the data centers and the energy market are two-fold: first, energy cost has been a primary portion as part of operating costs for a modern data center, thus understanding the energy market has become a crucial decision question during the construction of a physical data center; second, the flexibility of workloads in data centers also present as a potential opportunity in the dispatch process of the energy market. The energy market has received considerable attention, but proposals often rely on the assumption of convexity or special forms of cost functions [17–19]. Though convexity often dramatically simplifies the problem, real costs in the energy market are generally non-convex in practice. Thus a natural question arises: *can we design an efficient pricing scheme in the presence of non-convexity?*

1.2 An Overview of the Thesis

This thesis consists of four chapters divided into two parts. In Part I, we aim to tackle the new challenges by framing them as resource allocation problems. In Chapter 2, we first investigate how to schedule machine learning jobs composed of precedence-constrained tasks given a heterogeneous cluster of machines with nonuniform communication delays. Then in Chapter 3, we further study the scheduling problem while having both performance and energy goals in mind. Transitioning into Part II, we address the challenges from the angle of market design. In Chapter 4, we focus on the design of data markets in the presence of privacy concerns and information leakage. In Chapter 5, we study the pricing schemes with non-convex costs, with the energy market as a motivating example.

Chapter 2: Communication-Aware Scheduling in Cloud AI Systems

In this chapter, we study the scheduling of precedence-constrained tasks. Scheduling precedence-constrained tasks is a classical problem that has been studied for more than fifty years in the domain of parallel processing, and it recently has attracted extensive attention due to its applications in large scale ML platforms. However, little progress has been made in the setting where there are non-uniform communication delays between tasks. Results for the case of identical machines with non-uniform communication delays were derived nearly thirty years ago, and there has been some recent progress when communication delays are fixed. Yet no results for related machines with non-uniform communication delays have followed even though non-uniform communication delays are crucial for capturing the characteristics in cloud AI systems.

We propose a new scheduler, Generalized Earliest Time First (GETF), and prove that it computes a makespan that is at most of length $O(\log m/\log\log m)\mathrm{OPT}^{(i)} + C$ in the case of related machines and machine-dependent communication times, where C is the amount of communication time in a chain (path) in the precedence graph. Additionally, we generalize our result to the objective of total weighted completion time and show that GETF produces a schedule $\mathcal{S}$ whose total weighted completion time is at most $O(\log m/\log\log m)$ $\mathrm{wOPT}^{(i)} + \sum_j \omega_j C(\mathcal{S}, j)$, where $\mathrm{wOPT}^{(i)}$ is the optimal total weighted completion time, ω_j is the weight in the objective, and $C(\mathcal{S}, j)$ is the communication requirement in a chain in the precedence graph. These two results address long-standing open problems. This chapter summarizes the results in [20–22].

Chapter 3: Energy-Aware Scheduling in Cloud AI Systems

In this chapter, we study the scheduling problem of precedence-constrained tasks to balance between performance and energy consumption. To this point, scheduling to balance performance and energy has been limited to settings without dependencies between jobs. While many heuristics have been proposed for scheduling precedence-constrained ML tasks to optimize both performance and energy consumption, little progress has been made towards the goal of designing a scheduler to optimize both measures with a worst-case theoretic guarantee. We consider a system with multiple servers capable of speed scaling and seek to schedule precedence constrained jobs to minimize a linear combination of performance and energy consumption. Inspired by the single server setting, we propose the concept of *pseudo-size* for individual tasks, which is a measure of the importance of a task in the precedence graph that is learned from workload data. A task of a large pseudo-size is prioritized to run at a fast speed while those of a small pseudo-size are set to run slowly to conserve energy. We then propose a two-stage scheduling framework which uses the learned pseudo-size approximation and achieves a provable approximation bound on the linear combination of performance and energy consumption, where the quality of the bound depends on that of the approximation of task pseudo-sizes, for both makespan and total weighted completion time. This chapter summarizes the results in [22].

Chapter 4: The Privacy Paradox in Data Acquisition

While users claim to be concerned about privacy, often they do little to protect their privacy in their online actions. One prominent explanation for this "privacy paradox" is that when an individual shares her data, it is not just her privacy that is compromised; the privacy of other individuals with correlated data is also compromised. This information leakage encourages oversharing of data and significantly impacts the incentives of individuals in online platforms. In this chapter, we study the design of mechanisms for data acquisition in settings with information leakage and verifiable data. Concretely, we provide the first characterization of an optimal mechanism for data acquisition when agents are concerned about privacy and their data is correlated with each other. As a result, information leakage due to data correlation not only contributes to an agent's privacy cost, but also to the privacy costs of others with correlated data. Additionally, the mechanism allows, for the first time, a trade-off between the bias and variance of the estimator, where the worst-case is over the unknown correlation between costs and data, when privacy cost is considered. This offers the analyst freedom to tailor towards an emphasis on either bias or variance of the estimator depending on the contextual goals. Finally, we further study the mechanism via a characterization of monotonicity and non-monotonicity properties of the marketplace. This chapter summarizes the results of [23].

Chapter 5: Optimal Pricing in Markets with Non-Convex Costs

In this chapter, we consider a market run by an operator, who seeks to satisfy a given consumer demand for a commodity by purchasing the needed amount from a group of competing suppliers with non-convex cost functions. The operator knows the suppliers' cost functions and announces a price/payment function for each supplier, which determines the payment to that supplier for producing different quantities. Each supplier then makes an individual decision about how much to produce, in order to maximize its own profit. The key question is how to design the price functions.

To that end, we propose a new pricing scheme, which is applicable to general non-convex costs, and allows using general parametric pricing functions. Optimizing for the quantities and the price parameters simultaneously, and the ability to use general parametric pricing functions allows our scheme to find prices that are typically economically more efficient and less discriminatory than those of the existing schemes, while still supporting a competitive equilibrium. In addition, we supplement the proposed method with a computationally efficient polynomial-time approximation algorithm, which can be used to approximate the optimal quantities and prices for general non-convex cost functions. Our framework extends to the case of networked markets, which, to the best of our knowledge, has not been considered in previous works. Lastly, we evaluate the proposed method through extensive numerical examples, and show how it compares with the existing methods. This chapter summarizes the results in [24].

Part I

Resource Allocation

Chapter 2

COMMUNICATION-AWARE SCHEDULING IN CLOUD AI SYSTEMS

2.1 Introduction

In this chapter, we study scheduling precedence-constrained tasks onto a set of heterogeneous machines with non-uniform (machine-dependent) communication delays between the machines in order to minimize the makespan or the total weighted completion time. Initially, work on this topic was motivated by the goal of scheduling jobs on multi-processor systems [25]. Today this problem is timely due to the prominence of large-scale, general-purpose machine learning platforms. For example, in systems such as Google's TensorFlow [4], Facebook's PyTorch [5], and Microsoft's Azure Machine Learning (AzureML) [6], machine learning workflows are expressed via a computational graph, where jobs are made up of tasks, represented as vertices, and precedence relationships between the tasks, represented as edges. This "precedence graph" abstraction allows data scientists to quickly develop and incorporate modular components into their machine learning pipeline (e.g., data preprocessing, model training, and model evaluation) and then easily specify a workflow. The graphs that specify the workflows in platforms such as TensorFlow, PyTorch, and AzureML can be made up of hundreds or even thousands of tasks, and the jobs may be run on systems with thousands of machines. As a result, the performance of the platforms depends on how these precedence-constrained tasks are scheduled across machines.

The goal of scheduling jobs composed of precedence-constrained tasks has been studied for more than fifty years, starting with the work of [26]. The simplest version of this scheduling problem focuses on scheduling a single job with n precedence-constrained tasks on m identical parallel machines with the goal of minimizing the *makespan*: the time until the last task completes. More generally, the goal of minimizing the *total weighted completion time* is considered, where the total weighted completion time is a weighted average of the completion time of each task in the job. Note that makespan is a special case of total weighted completion time as a dummy task with weight one can be added as the final task of the job, with all other tasks given weight zero. For the goal of minimizing the makespan, Graham showed that a simple list scheduling algorithm can find a schedule of length within a multiplicative factor of $(2 - 1/m)$ of the optimal. This result is still the best guarantee known for this simple setting. Since then, research has sought to generalize the setting considered in two important ways: (i) to non-identical machines and (ii) to the case where communication is needed between tasks.

Addressing these two issues has been one of the major goals of the field since Graham's initial result fifty years ago. Since that time, considerable progress has mostly been made on generalizations to heterogeneous machines. The focus has been on *(uniformly) related machines*, a model where each machine i has a speed s_i, each task j has a size w_j, and the time to run task j on machine i is w_j/s_i. Under the related machine model, a sequence of results in the 1980s and 1990s culminated in a result that showed how to use list scheduling algorithms in combination with a partitioning of machines into groups with "similar" speeds in order to achieve an $O(\log m)$-approximation algorithm for makespan [9]. This result was also extended in the same work to total weighted completion time by proposing a time-indexed linear programming technique. The extension yields an $O(\log m)$-approximation for total weighted completion time. The idea of using a *group assignment* rule to partition machines into groups of machines with similar speeds and then to assign tasks to a group is a powerful one and has shown up frequently in the years since; it recently led to a breakthrough when the idea of partitioning machines was adapted further and combined with a variation of list scheduling to obtain a $O(\log m/\log\log m)$-approximation algorithm for both makespan and total weighted completion time [10].

Despite the progress made in generalizing from identical machines to heterogeneous machines, there has been little progress toward the goal of incorporating communication delays. Subsequent to this work, some progress has been made in the case of fixed communication delays (see Related literature). However, machine-dependent communication delays are crucial for capturing issues such as data locality and the difference between intra-rack and inter-rack communication. The state-of-the-art result in the case of machine-dependent communication delays is [27], which studies machine-dependent communication costs in the setting of *identical machines*. In this context, a greedy algorithm called Earliest Time First (ETF) has been shown to produce schedules with a makespan bounded by $(2 - 1/m)\mathrm{OPT}^{(i)} + C$, where $\mathrm{OPT}^{(i)}$ is the optimal schedule length when ignoring communication time and C is the maximum amount of communication of a chain (path) in the precedence graph.

However, the analysis for the case of identical machines in [27] is quite complex and it has proven difficult to generalize to the related machines setting. As a result, there has been no progress outside the context of identical machines since [27].

Given the challenge of designing schedulers that are approximately optimal for related machines with machine-dependent communication time, most work studying the design of scheduling policies in this context has relied on developing scheduling heuristics and evaluating these heuristics numerically [e.g., 28–33]. For a recent survey, see [34, 35] and the references therein.

Contributions. In this paper, we propose a new scheduler, Generalized Earliest Time First (GETF), and prove that it computes a makespan that is at most of length $O(\log m/\log\log m)\text{OPT}^{(i)} + C$ in the case of related machines and machine-dependent communication times, where C is the amount of communication time in a chain (path) in the precedence graph. Additionally, we generalize our result to the objective of total weighted completion time and show that GETF produces a schedule $\mathcal{S}$ whose total weighted completion time is at most $O(\log m/\log\log m)$ $\text{wOPT}^{(i)} + \sum_j \omega_j C(\mathcal{S}, j)$, where $\text{wOPT}^{(i)}$ is the optimal total weighted completion time, ω_j is the weight in the objective, and $C(\mathcal{S}, j)$ is the communication requirement in a chain in the precedence graph. These two results address long-standing open problems. Note that the makespan result matches state-of-the-art bounds for the special cases (i) when there is zero communication time and (ii) when the machines are identical. In the case of total weighted completion time, no previous result exists for the case of identical machines with communication time, but the result matches the best known bound for the case with related machines and zero communication time.

The key technical advance that enables our new result is a dramatically simplified analysis of ETF in the setting of identical machines. The state-of-the-art result in this setting is [27], which is established using a long, complex argument. In contrast, the core idea in our proof of Theorem 2.4.1 is a short, simple proof of a *Separation Principle* which can be used to provide a novel proof of the approximation ratio for ETF in the case of identical machines. The proof is simple and general enough that it can be extended from identical machines to related machines by adapting recent advances from [10].

Related literature. In recent years, the design and optimization of large-scale general-purpose machine learning platforms has been an overarching goal, bridging many communities in both industry and academia. The emergence of platforms such as TensorFlow, PyTorch, and AzureML illustrate the power of such systems to democratize tools from machine learning, making them accessible and scalable for anyone.

Since the emergence of such systems, there has been a torrent of work that seeks to optimize the scheduling and assignment of the precedence-constrained graphs in such systems. Heuristics have emerged for managing straggler tasks [e.g., 28, 29, 36, 37]; scheduling tasks with different computational properties, e.g., jobs with MapReduce-type structures [38–43], scheduling approximation jobs [28, 37, 44], and managing communication times [e.g., 35, 45]. Many of these heuristics have led to system designs that have had a significant industrial impact.

Such designs typically address the challenges associated with precedence constraints in ad hoc ways based on simplifying assumptions about the structures of the graphs. In contrast, there is a long history of analytic work seeking to design schedulers for precedence-constrained tasks with provable worst-case guarantees. As we have already mentioned, the initial results on this topic for makespan were provided by Graham, who gave a $(2 - 1/m)$-approximation algorithm based on list scheduling for $P|prec|C_{max}$ [26]. A decade later, it was shown by [46] that it is NP-hard to approximate $P|prec|C_{max}$ within a factor of 4/3. This left a gap which has been essentially closed recently, when [47] proved that it is NP-hard to achieve an approximation factor less than 2, given the assumption of a new variant of the Unique Game Conjecture introduced by [48]. In the case of total weighted completion time objective $P|prec|\sum_j \omega_j C_j$, the negative results carry over from the makespan objective since makespan objective can be viewed as a special case of total weighted completion time objective. Moreover, under the assumption of the stronger version of the Unique Game Conjecture, it is shown in [48] that it is even hard to approximate within a factor of $2 - \epsilon$ for the problem with one machine. On the positive side, [49] gave a 7-approximation, and [50, 51] later improved it to a 4-approximation. The current best known result is a $(2 + 2\ln 2 + \epsilon)$-approximation by [10] via a time-indexed linear programming relaxation technique.

The results mentioned above all focus on identical machines with zero communication delays. When related machines are considered, the problem becomes more challenging. An early result on this topic is [9], which proposed a Speed-based List Scheduling (SLS) algorithm that obtains an approximation of $O(\log m)$ for $Q|prec|C_{max}$. A time-indexed linear programming technique has been proposed in the same work that gives a $O(\log m)$ bound for $Q|prec|\sum_j \omega_j C_j$. Recently, an improvement to $O(\log m/\log\log m)$ for both objectives was proven in [10]. The best known lower bound for the problem of related machines is from [52], which shows that it is impossible for a polynomial time algorithm to approximate the minimal makespan to any constant factor assuming the hardness of an optimization problem on k-partite graphs.

In contrast, when communication delay is considered, much less is known. To our knowledge, no approximation ratio is known for $P|prec, c_{i,j}|C_{max}$, and this open problem was noted by [53]. The only algorithm with a guaranteed worst-case performance bound in this setting is ETF [27], which provides a bound of $(2 - 1/m)\text{OPT}^{(i)} + C$ on the makespan in the case of identical machines. Prior to our paper, no algorithm with a worst-case approximation guarantee for either makespan or total weighted completion time is known for the case of related machines with non-uniform communication delays, i.e., $Q|prec, c_{i,j}|C_{max}$ and $Q|prec, c_{i,j}| \sum_j \omega_j C_j$. Some progress was made subsequently to publication of the extended abstract of this work [20]. When communication delays are fixed, [11] proposed a $O(\log c \cdot \log m)$-approximation algorithm in the case of identical machines, i.e., $P|prec, c|C_{max}$. Under the related machines model, i.e., $Q|prec, c|C_{max}$, [12] proposed a $O(\log m \log c / \log \log c)(OPT+c)$-approximation algorithm where OPT is the optimal makespan for the problem when duplication is allowed. Further, they were able to bound the duplication advantage to compute a no-duplication schedule. However, both papers consider fixed communication delays. Additionally, they both emphasize that their results do not apply to machine-dependent communication delays and highlight the case of machine-dependent communication delays as an important open question.

2.2 Problem Formulation

We study a model that generalizes $Q|prec, c_{i,j}| \sum_j \omega_j C_j$ by including machine-dependent communication times. Our goal is to derive bounds on the total weighted completion time and the makespan, which is an important special case of the total weighted completion time that uses a particular choice of ω_j.

Specifically, we consider the task of scheduling a job made up of a set V of n tasks on a heterogeneous system composed of a set M of m machines with potentially different processing speeds and communication speeds. The tasks form a directed acyclic graph (DAG) $G = (V, E)$, in which each node j represents a task and an edge (j', j) between task j and task j' represents a precedence constraint. We interchangeably use node or task, as convenient. Precedence constraints are denoted by a partial order $\prec$ between two nodes of any edge, where $j' \prec j$ means that task j can only be scheduled after task j' completes. Let w_j represent the processing demand of task j. The amount of data to be transmitted between task j' and task j is represented by the edge weight $w_{j',j}$ of (j', j).

The system is heterogeneous in two aspects: processing speed and communication speed. For processing speed, we consider the classical *related machines* model: a machine i has speed s_i, and it takes w_j/s_i uninterrupted time units for task j to complete on machine i. Specifically, computer resources such as CPUs and GPUs have varying speeds; hence schedulers must be able to handle heterogeneous servers. The communication speed $s_{i',i}$ between any two machines i', i is heterogeneous across different machine pairs. We index the machine to which task j is assigned by $h(j)$. If $i = h(j)$ and $i' = h(j')$, then communication time between task j' and j in the DAG is $w_{j',j}/s_{i',i}$.

For simplicity, we consider a setting where the machines are fully connected to each other, so any machine can communicate with any other machine. This is without loss of generality as one can simply set the communication speed between any two disconnected machines to 0. We also assume that the DAG is connected. Again, this is without loss of generality because, otherwise, the DAG can be viewed as multiple DAGs and the same results can be applied to each. As a result, our results trivially apply to the case of multiple jobs. Additionally, our model assumes that each machine (processing unit) can process at most one task at a time, i.e., there is no *time-sharing*, and the machines are assumed to be *non-preemptive*, i.e., once a task starts on a machine, the scheduler must wait for the task to complete before assigning any new task to this machine. This is a natural assumption in many settings, as interrupting a task and transferring it to another machine can cause significant processing overhead and communication delays due to data locality, e.g., [54].

The goal of the scheduler in our model is to minimize the *total weighted completion time* of the job, denoted by $\sum_j \omega_j C_j$, where C_j is the completion time of task j and ω_j is the weight associated with task j. We also consider the *makespan*, denoted by C_{max}, which is the time when the the final task in the DAG completes.

Note that the problem we consider is an offline scheduling problem. This is a classical problem with relevance to modern ML platforms, which use batch scheduling of precedence-constrained tasks in their pipelines, e.g., [4]. It is also known to be challenging. Specifically, minimizing the makespan (and hence also minimizing the total weighted completion time) of jobs with precedence constraints is known to be NP-complete [55]. Thus, we aim to design a polynomial-time algorithm that computes an *approximately* optimal schedule. We say that an algorithm is a ρ-approximation algorithm if it always produces a solution with an objective value within a factor of ρ of optimal in polynomial time.

Our main results use three important concepts. First, our results provide bounds in terms of $\mathrm{OPT}^{(i)}$ and $\mathrm{wOPT}^{(i)}$, which are the optimal makespan and the optimal total weighted completion time if the communication delays were zero, respectively. Note that $\mathrm{OPT}^{(i)}$ and $\mathrm{wOPT}^{(i)}$ are a lower bound of the corresponding objectives of the problem when communication delays are not included. Second, we provide bounds in terms of the communication time of a *terminal chain* of the schedule. A *chain* in the DAG is a sequence of immediate predecessor-successor pairs, whose first node is a node with no predecessor and last node is a leaf node with no successors. Third, we provide bounds in terms of the communication time of a terminal chain of a subset of the DAG that is naturally formed in the scheduling process. Formally, for any given schedule, a terminal chain $\mathbb{C}$ of length N can be constructed in the following fashion. We start with one of the tasks that ends last in the given schedule, denoted as c_N. Among all the immediate predecessors of node c_N, we pick one of the tasks that finishes last and define it as c_{N-1}. In such a way, we can construct a chain of tasks $c_1 \prec c_2 \prec \ldots \prec c_N$ until the first node c_1 in the chain does not have a predecessor. There may be many such terminal chains, and our results apply to any arbitrary terminal chain for the given schedule.

2.3 Generalized Earliest Time First (GETF) Scheduling

In this section, we introduce a new algorithm – Generalized Earliest Time First (GETF) – for scheduling tasks with precedence constraints in settings where servers have heterogeneous service rates and communication times. For GETF, we provide provable worst-case approximation guarantees for both the goal of minimizing the makespan and minimizing the total weighted completion time. At its core, GETF is a greedy algorithm. Like ETF, it seeks to run tasks that can be started earliest, thus minimizing the idle time created by the precedence constraints in a greedy way. However, this simple heuristic does not take into account the potential difference between the service rates of different machines. For this, GETF is similar to SLS. It uses a group assignment function $f(\cdot)$ to determine sets of "similar" machines, and then assigns tasks to different groups of machines. Within the groups of similar machines, GETF uses the ETF greedy allocation rule.

Algorithm 2.1 Generalized Earliest Time First (GETF)

INPUT: group assignment rule $f(\cdot)$, tie-breaking rule
OUTPUT: schedule $\mathcal{S}$ with machine assignment mapping $h(\cdot)$ and starting time mapping $t(\cdot)$

$R \leftarrow \{1, 2, \ldots, n\}$
while $R \neq \emptyset$ **do**
  $A = \{j : j \in R, \nexists j' \text{ s.t. } j' \in R \text{ and } j' \prec j\}$
  For $j \in A, t'_j =$ earliest starting time on machine m'_j s.t. $m'_j \in f(j)$
  $B = \{j : j = \arg\min_{j' \in A} t(j')\}$
  Choose j from B to start on machine m'_j with a starting time t'_j based on the given tie-breaking rule
  $h(j) = m'_j, t(j) = t'_j$
  $R \leftarrow R \setminus \{j\}$
end while

GETF is parameterized by a group assignment function $f(\cdot)$ and a tie-breaking rule, and proceeds in two stages. At every iteration, GETF finds a set A of all the tasks that are ready to process and are not yet scheduled. For every task in A, GETF calculates the earliest starting time if it was only allowed to schedule on machines in the assigned group. Then, GETF computes B, the set of tasks in A with the earliest starting times, and chooses one of the tasks to process on a machine based on the tie-breaking rule. The pseudocode for GETF is presented in Algorithm 2.1 and Figure 2.1 in Section 2.3.3 illustrates the operation of GETF on a simple example (Example 2.3.1).

GETF can be instantiated with different group assignment and tie-breaking rules. To understand how these rules work, consider a situation where the m machines are divided into K groups $M_1, M_2, \ldots, M_K$ by a group assignment rule. Let $f(j)$ denote the group of machines to which task j can be assigned, $j = 1, \ldots, n$. Given this notation, a schedule under GETF consists of two mappings: a mapping $h(\cdot)$ from each task to its assigned machine and a mapping $t(\cdot)$ from each task to its starting time. Further, for any schedule with $h(\cdot)$ produced by GETF, $h(\cdot)$ of the produced schedule should be consistent with group assignment function $f(\cdot)$, i.e., $h(j) \in f(j)$ for each task j.

The choice of the group assignment rule has a significant impact on the performance of GETF. Indeed, different group assignment functions are used for the goals of minimizing the makespan and total weighted completion time. While our results hold for any tie-breaking rule, different tie-breaking rules could provide meaningful improvements in real-world workloads. As it could be helpful to keep a specific tie-breaking rule in mind while considering the algorithm and proofs, the reader may find it helpful to consider random tie-breaking. Our technical results are based on the specific group assignment functions described in the following subsections.

2.3.1 A Group Assignment Rule for Makespan

The group assignment rule $f_{\mathsf{mksp}}(\cdot)$ for the goal of minimizing the makespan that we focus on is adapted from SLS, which is designed for the setting *without* communication time. Specifically, machines of similar speeds are grouped together as follows.

First, all the machines with speed less than a $\frac{1}{m}$ fraction of the speed of the fastest machine are discarded. Then, the remaining machines are divided into K groups $M_1, M_2, \ldots, M_K$ where $K = \lceil \log_\gamma m \rceil$, $\gamma = \log m / \log \log m$. Note that $K = O(\log m / \log \log m)$. Given the removal of the slowest machines, we can assume that any remaining machine has speed within a factor of $\frac{1}{m}$ of the fastest machine. Without loss of generality, we assume the speed of the fastest machine is m and the group M_k contains machines with speeds in range $[\gamma^{k-1}, \gamma^k)$.

It may seem strange that some machines are discarded, but note that the total speed of discarded machines is not bigger than the speed of the fastest machine. So, if we consider the scheduling problem with zero communication time, removing these machines at most doubles the makespan in the worst case.

After dividing machines into K groups in the preprocessing step, we need to assign the machines. This step is more involved than the division. The design of the group assignment rule $f_{\mathsf{mksp}}(\cdot)$ is based on the solution of a linear program (LP), which is a relaxed version of the following mixed integer linear program (MILP).

While the MILP is only designed to produce a group assignment rule, its optimal solution does not necessarily provide a feasible schedule. In the MILP, $x_{i,j} = 1$ if task j is assigned to machine i; otherwise $x_{i,j} = 0$. For each task j, C_j denotes the completion time of task j. Constraint (2.1a) ensures that every task is processed on some machine. For any task j, processing time $w_j \sum_i \frac{x_{i,j}}{s_i}$ is bounded by its completion time as in constraint (2.1b). Constraint (2.1c) enforces the precedence constraints between any predecessor-successor pair (j', j). Constraint (2.1d) guarantees that the total load assigned to machine i is $w_j \sum_i \frac{x_{i,j}}{s_i}$ and it should not be greater than the makespan. Finally, constraint (2.1e) states that the makespan should not be smaller than the completion time of any task.

$$\min_{x_{i,j},C_j,T} \quad T$$

$$\sum_i x_{i,j} = 1 \qquad \forall j \tag{2.1a}$$

$$w_j \sum_i \frac{x_{i,j}}{s_i} \le C_j \qquad \forall j \tag{2.1b}$$

$$C_{j'} + w_j \sum_i \frac{x_{i,j}}{s_i} \le C_j \quad j' \prec j \tag{2.1c}$$

$$\frac{1}{s_i} \sum_j w_j x_{i,j} \le T \qquad \forall i \tag{2.1d}$$

$$C_j \le T \qquad \forall j \tag{2.1e}$$

$$x_{i,j} \in \{0,1\} \ \forall i, j. \tag{2.1f}$$

Since we cannot solve the MILP efficiently, we relax it to form an LP by replacing constraint (2.1f) with $x_{i,j} \ge 0$. Let x^*, C^*, T^* denote the optimal solution of this LP. Note that T^* provides a lower bound on $\mathrm{OPT}^{(i)}$, the optimal makespan for the same problem with zero communication time.

For a set $M_k \subseteq M$ of machines, let $s(M_k)$ denote the total speed of machines in M_k, i.e.,

$$s(M_k) = \sum_{i \in M_k} s_i.$$

Define $x^*_{M_k,j}$ as the total fraction of task j assigned to machines in set M_k:

$$x^*_{M_k,j} = \sum_{i \in M_k} x^*_{i,j}.$$

For any task j, define ℓ_j as the largest group index such that at least half of the tasks are fractionally assigned to machines in groups $M_\ell, \dots, M_K$:

$$\ell_j = \max_{\ell} \ell \quad \text{s.t.} \quad \sum_{k=\ell}^{K} x^*_{M_k,j} \ge \frac{1}{2}.$$

We note that any choice of constant above works for the purpose of our worst case analysis of GETF, but the choice can potentially have an impact on its empirical performance. Thus the choice of the parameter should be further optimized when applied in practice. Each task j is assigned to the group $f_{\mathsf{mksp}}(j)$ that maximizes the total speed of machines in that group among candidates $M_{l_j}, \dots, M_K$, i.e.,

$$f_{\mathsf{mksp}}(j) = \operatorname*{arg\,max}_{M_k : \ell_j \le k \le K} \quad s(M_k).$$

2.3.2 A Group Assignment Rule for Total Weighted Completion Time

The group assignment rule $f_{\mathsf{twct}}(\cdot)$ for the goal of minimizing the total weighted completion time is similar in spirit to $f_{\mathsf{mksp}}(\cdot)$, but is based on modified solutions of a different LP. We divide machines into groups in the same way as in Section 2.3.1. Without loss of generality, we assume that $\frac{w_j}{s_i} \geq 1$ for any task j to be processed on any machine i. Thus, we can divide the time horizon into the following time-indexed intervals of possible task completion times: $[1,2], (2,4], (4,8], \ldots, (\tau_{Q-1}, \tau_Q]$ where $Q = \log(\sum_j \frac{w_j}{\min_i s_i})$ and $\tau_q = 2^q$ for $0 \leq q \leq Q$. Then, the MILP that forms the basis for the group assignment rule can be formulated as follows:

$$\min_{x_{i,j,q}, C_j} \quad \sum_j \omega_j C_j$$

$$\sum_i \sum_q x_{i,j,q} = 1 \qquad \forall j \tag{2.2a}$$

$$w_j \sum_i \frac{1}{s_i} \sum_q x_{i,j,q} \leq C_j \qquad \forall j \tag{2.2b}$$

$$C_{j'} + w_j \sum_i \frac{1}{s_i} \sum_q x_{i,j,q} \leq C_j \qquad j' \prec j \tag{2.2c}$$

$$\sum_{t=1}^{q} \sum_i x_{i,j,t} - \sum_{t=1}^{q} \sum_i x_{i,j',t} \leq 0 \qquad \forall q, j' \prec j \tag{2.2d}$$

$$\sum_q \tau_{q-1} \sum_i x_{i,j,q} < C_j \qquad \forall j \tag{2.2e}$$

$$\frac{1}{s_i} \sum_j w_j \sum_{t=1}^{q} x_{i,j,t} \leq \tau_q \qquad \forall i, q \tag{2.2f}$$

$$x_{i,j,q} \in \{0,1\} \quad \forall i, j, q. \tag{2.2g}$$

Again, the MILP is only designed to find a group assignment rule, and thus its optimal solution does not necessarily produce a feasible schedule. Here, $x_{i,j,q} = 1$ if task j is assigned to machine i and it completes in the qth interval $(\tau_{q-1}, \tau_q]$. For each task j, C_j denotes the completion time of task j and ω_j represents its weight in the objective of total weighted completion time. Constraint (2.2a) enforces that each task will be assigned to some machine. Constraint (2.2b) guarantees that the completion time of a task is not smaller than its processing time. Constraints (2.2c) and (2.2d) together enforce the precedence constraint for every predecessor-successor pair. Constraint (2.2e) guarantees that the completion time of task j is not smaller than the left boundary of the qth interval $(\tau_{q-1}, \tau_q]$. The total load assigned to machine i up to

qth interval is $\frac{1}{s_i}\sum_j w_j \sum_{t=1}^{q} x_{i,j,t}$, and it should not be greater than the upper bound τ_q as enforced in constraint (2.2f).

To define the group allocation rule, we relax constraint (2.2g) to form an LP. As in the previous section, let x^*, C^* denote the optimal solution for this LP. Note that $\sum_j \omega_j C_j^*$ provides a lower bound for wOPT$^{(i)}$. For any task j, define $q(j)$ as the minimum value of q such that both $\sum_{t=1}^{q}\sum_i x_{i,j,t}^* \geq \frac{1}{2}$ and $C_j^* \leq 2^q$ are satisfied. Intuitively, $q(j)$ can be viewed as a rough estimate of the completion time of task j. Define $\alpha(j)$ as the total fraction of task j over any machine in the first $q(j)$ intervals with respect to solution x^*:

$$\alpha_j = \sum_{t=1}^{q(j)}\sum_i x_{i,j,t}^*.$$

We construct a set of feasible solutions $\tilde{x}$ based on the optimal solution x^* for the LP:

$$\tilde{x}_{i,j} = \sum_{q=1}^{q(j)} \frac{x_{i,j,q}^*}{\alpha_j} \quad \forall i, j. \tag{2.3}$$

Notice that the group assignment rule $f_{\mathsf{twct}}(\cdot)$ is of the same form as $f_{\mathsf{mksp}}(\cdot)$, with $\tilde{x}$ replacing x^*. For task j, define $\tilde{\ell}_j$ as before but with respect to $\tilde{x}$ instead of x^*:

$$\tilde{\ell}_j = \max_{\ell} \ell \quad \text{s.t.} \quad \sum_{k=\ell}^{K} \tilde{x}_{M_k,j} \geq \frac{1}{2}.$$

The group assignment rule $f_{\mathsf{twct}}(\cdot)$ for the goal of minimizing the total weighted completion time follows as below:

$$f_{\mathsf{twct}}(j) = \operatorname*{arg\,max}_{M_k:\tilde{\ell}_j \leq k \leq K} s(M_k).$$

2.3.3 A Comparison of GETF and SLS

The description of GETF above highlights that it combines the greedy heuristic of ETF with the speed-based assignment heuristic of SLS. This enables GETF to provide guarantees for settings with both heterogeneous processing rates and communication delays. In contrast, SLS does not provide guarantees in settings with communication time. This is a result of the fact that SLS is based on list scheduling and does not always schedule the earliest task first, thus making it impossible to bound the overall idle time in between tasks.

To illustrate the difference between GETF and SLS, we provide a simple example of scheduling a job made up of four tasks.

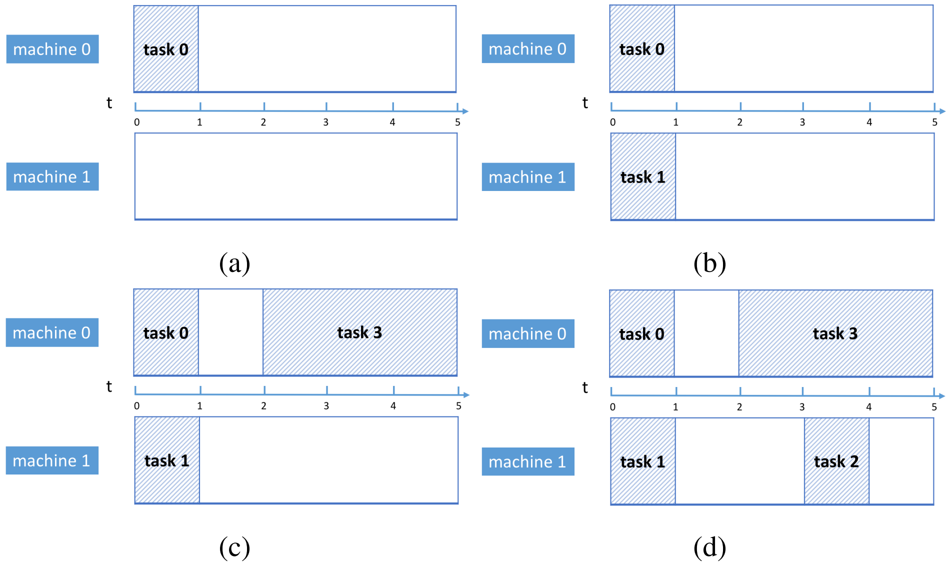


Figure 2.1: An illustration of GETF running on Example 2.3.1. (a)-(d) show the first four iterations.

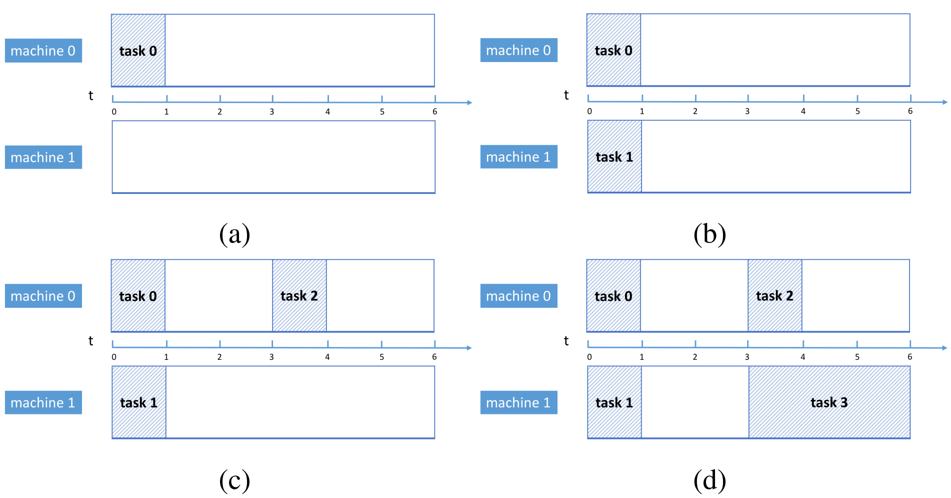


Figure 2.2: An illustration of SLS running on Example 2.3.1. (a)-(d) show the first four iterations.

Example 2.3.1. *We consider a job made up of four tasks,* $0, 1, 2, 3$ *with processing demands* $1, 1, 1,$ *and* 3 *that are to be scheduled on a set of two identical machines with the same processing speed equal to* 1*. The weight for the edges in the graph are listed as below:* $w_{0,2} = w_{0,3} = w_{1,2} = 2, w_{1,3} = 1$*. We assume* $s_{i,j} = 1$ *for* $i \neq j$*; otherwise* $s_{i,i} = 2$ *for* $i = 0, 1$*.*

The schedules of GETF and SLS are illustrated in Figures 2.1 and 2.2. Note that, since the servers are identical, the group assignment rule does not play a role in these examples. Given a priority list $(0, 1, 2, 3)$*, a possible schedule produced by SLS puts tasks* 0 *and* 2 *on machine* 0 *and assigns the rest of the tasks to machine* 1 *as demonstrated in Figure 2.2. A terminal chain for the given schedule is task* 1 *followed by task* 3*, and the idle time of length* 2 *between the end of task* 1 *and the start of task* 3 *on machine* 1 *is not bounded by the communication time between task* 1 *and* 3*. In contrast, task* 3 *starts earlier on machine* 0 *in a schedule produced by GETF, see Figure 2.1. List scheduling does not always schedule the earliest task at each step, thus making the idle time on machine* 1 *not necessarily bounded by communication time between task* 1 *and task* 3*. Our proofs in Section 2.4.1 highlight that maintaining a tight bound on the communication time between tasks is crucial to achieving a good approximation ratio in settings with machine-dependent communication time.*

2.4 Results

Our main results bound the approximation ratio of GETF in settings with related machines and heterogeneous communication time for the goals of minimizing the makespan and minimizing the total weighted completion time.

2.4.1 Makespan

In the case of minimizing the makespan, our main result provides a bound in terms of the communication time of a terminal chain of the schedule. Specifically, let $\mathbb{C} : c_1 \prec c_2 \prec \ldots \prec c_N$ be a terminal chain for the schedule and define C as the communication time over such a chain in the worst case, i.e.,

$$C = \sum_{j=2}^{N} \frac{w_{c_{j-1},c_j}}{\bar{s}(c_{j-1}, c_j)},$$

where $\bar{s}(c_{j-1}, c_j)$ is defined as the slowest speed between $h(c_{j-1})$, the machine assigned to c_{j-1} and any machine in the group $f(c_j)$, i.e.,

$$\bar{s}(c_{j-1}, c_j) = \min_{i \in f(c_j)} s_{h(c_{j-1}),i}.$$

Note that C can be computed efficiently and minimized over all the terminal chains using dynamic programming and that the tie-breaking rule can have an impact on C due to its impact on terminal chains.

Theorem 2.4.1. *For any schedule $\mathcal{S}$ produced by GETF with group assignment rule, $f_{\mathsf{mksp}}(\cdot)$*

$$C_{max}(\mathcal{S}) \leq O(\log m/\log\log m)OPT^{(i)} + C,$$

where $OPT^{(i)}$ is the optimal schedule length obtained if communication time for all pairs were zero.

Theorem 2.4.1 represents the first result for makespan in the setting of related machines and heterogeneous communication time, addressing a problem that has been open since ETF was introduced for identical machines thirty years ago. Additionally, it matches the state-of-the-art results for the case without communication time, where the best known approximation ratio is $O(\log m/\log\log m)$ [10], and the case with communication time but identical machines, where the best known approximation ratio is $(2 - \frac{1}{m})\mathrm{OPT}^{(i)} + C$ [27].

Concretely, in the special case of identical machines, the group assignment rule $f_{\mathsf{mksp}}(\cdot)$ is no longer required when implementing GETF since all machines share the same speed, so there is only one group of machines. Thus, GETF reduces to ETF. The theorem makes use of C' which is defined as

$$C' = \frac{1}{m}\sum_{j=2}^{N}\sum_{i=1}^{m}\frac{w_{c_{j-1},c_j}}{s_{h(c_{j-1}),i}}.$$

Note that C' differs from C since it is an average over the terminal chain. The result we obtain in this case is the following, which matches the current state-of-the-art result of [27].

Proposition 2.4.1. *Consider a setting with m identical machines. For any schedule $\mathcal{S}$ produced by GETF,*

$$C_{max}(\mathcal{S}) \leq \left(2 - \frac{1}{m}\right)OPT^{(i)} + C',$$

where $OPT^{(i)}$ is the optimal schedule length obtained if communication time for all pairs were zero.

2.4.2 Total Weighted Completion Time

Similarly to the makespan case, we provide a bound with respect to the communication time of chains. However, since total weighted completion time depends on the completion time of every task (instead of just one task as in the case of makespan), the communication time of terminal chains of many subsets of the DAG show up in the bound. More formally, assume that the tasks are indexed with respect to their order in the schedule determined by GETF, denoted by $\mathcal{S}$. At iteration j, task j is to be scheduled. Let $G(\mathcal{S}, j)$ denote a DAG formed by a set of the tasks that have been scheduled so far and the corresponding edges within these tasks. Define $\mathcal{S}(j)$ to be a subset of the given schedule $\mathcal{S}$ up to iteration j, i.e., it is a schedule for DAG $G(\mathcal{S}, j)$. This definition ensures that task j is one of the tasks that ends last in the schedule $\mathcal{S}(j)$. Now, let $\mathbb{C}(\mathcal{S}, j) : c_1 \prec c_2 \prec \cdots \prec c_{N_j}$ be a terminal chain that ends with task $j = c_{N_j}$ in the schedule $\mathcal{S}(j)$, and define $C(\mathcal{S}, j)$ as the communication time over such a chain in the worst case, i.e.,

$$C(\mathcal{S}, j) = \sum_{j'=2}^{N_j} \frac{w_{c_{j'-1}, c_{j'}}}{\bar{s}(c_{j'-1}, c_{j'})}.$$

This definition of $C(\mathcal{S}, j)$ generalizes the notion of C used in Theorem 2.4.1 for makespan and plays a similar role in the theorem below.

Theorem 2.4.2. *For any schedule $\mathcal{S}$ produced by GETF with group assignment rule $f_{\mathsf{twct}}(\cdot)$,*

$$\sum_j \omega_j C_j \leq O(\log m / \log\log m) wOPT^{(i)} + \sum_j \omega_j C(\mathcal{S}, j),$$

where $wOPT^{(i)}$ is the optimal total weighted completion time obtained if communication time for all pairs was zero.

Theorem 2.4.2 is the first result on total weighted completion time for the setting of related machines with heterogeneous communication time and it matches the bounds in cases where previous results exist. In particular, if the weights are chosen so as to recover makespan, then the bound matches that of Theorem 2.4.1. Similarly, results for identical machines can be recovered as done in the case of makespan. However, note that the group assignment rule used for GETF here is different than that in Theorem 2.4.1. The rule used in Theorem 2.4.2 applies more generally but, while both group assignment rules yield the same worst-case performance bound for makespan, we expect that the rule used in Theorem 2.4.1 will lead to a smaller makespan in most practical settings as it is designed for the purpose of minimizing the makespan.

2.5 Proofs

In this section, we present our proofs of Theorems 2.4.1 and 2.4.2. The general form of both arguments is similar; however, the case of total weighted completion time is more involved. The first step of our argument is to show a general upper bound, which is valid for GETF regardless of choices of group assignment function $f(\cdot)$, and tie-breaking rule. This *Separation Principle* can be used to easily establish the result for makespan in the case of identical machines (Proposition 2.4.1), and represents a significant simplification compared to existing proofs of that result in the literature. We then tighten the general bound by taking advantage of the choices of $f(\cdot)$ described in Section 2.3 for makespan and total weighted completion time. Finally, we establish a connection between the makespan and total weighted completion time in the same settings by introducing a time-indexed LP that enables us to bound the total weighted completion time.

2.5.1 A Separation Principle

The Separation Principle presented here is a key component of our proof of Theorem 2.4.1. The core of nearly all proofs in this area is the construction of a chain, which is then used to bound the overall makespan. This idea goes back to the first list scheduling algorithms proposed by [26]. The key to our argument is to bound the amount of communication time between any predecessor-successor pairs in a terminal chain. However, as we discuss in Section 2.3, it is not possible to do this under list scheduling algorithms.

Our approach also differs considerably from the approach used to study ETF in [27], where the authors divide $[0, C_{max}]$ into two sets of time intervals, one for the time when all the machines are busy and the other that one chain covers. Extending this approach to related machines does not appear possible. In contrast, in our argument, the construction of a terminal chain is simple and so we can identify the set of time intervals between tasks in the terminal chain and take advantage of the greedy nature of GETF to bound these times directly.

A key feature of the the Separation Principle below is that it separates the analysis of the terminal chain from the analysis of the group assignment rule, which provides another valuable simplification of the previous proof approaches.

Theorem 2.5.1 (Separation Principle)**.** *For any choice of group assignment function $f(\cdot)$ and tie-breaking rule, GETF produces a schedule $\mathcal{S}$ of makespan*

$$C_{max}(\mathcal{S}) \leq P + \sum_{k=1}^{K} D_k + C,$$

where

$$P = \sum_{c_j \in \mathbb{C}} \frac{w_{c_j}}{s_{h(c_j)}},$$

$$D_k = \frac{\sum_{j:k\in f(j)} w_j}{s(M_k)},$$

$$C = \sum_{j=1}^{N-1} \frac{w_{c_j,c_{j+1}}}{\bar{s}(c_j, c_{j+1})}.$$

Note that the upper bound in this result is valid regardless of the choice of group assignment rule and tie-breaking rule. P is the sum of processing times along a terminal chain and D_k can be viewed as total load assigned to machines in group M_k. Both P and D_k, $k = 1, 2, \ldots, K$, are not dependent on the communication constraint, which enables us to take advantage of any good choice of group assignment rule $f(\cdot)$ for general DAG scheduling, even in the case of zero communication time.

Proof. Our proof proceeds in four steps:

(i) Define a terminal chain $\mathbb{C}$. Recall that a chain $\mathbb{C}$, $c_1 \prec c_2 \prec \ldots \prec c_N$ is a terminal chain when task c_N completes at the end of the overall schedule.

(ii) Partition the overall makespan into $K + 1$ parts. The idea of this step is to decouple $[0, C_{max}]$ into one part where the tasks in the terminal chain are being processed and K other parts associated with each machine group. Dependent on the choices of group assignment rule, we can further bound these $K + 1$ parts.

(iii) Bound the idle time in between tasks. The greedy nature of GETF makes it possible to bound the length of the idle time intervals between tasks by communication delays of task pairs.

(iv) Combine (ii) and (iii) to bound the overall makespan in terms of the communication time of the terminal chain.

(i) *Define a terminal chain* $\mathbb{C}$. To find a terminal chain of length N, we start with one of the tasks that ends last, denoted as c_N. According to the definition of $h(\cdot)$ and $t(\cdot)$, task c_N is assigned to machine $h(c_N)$ in group $f(c_N)$ with a starting time $t(c_N)$. Among all the immediate predecessors of task c_N, we pick one of the tasks that finishes last and define it as c_{N-1}. In such a fashion, we construct a chain $\mathbb{C}$ of tasks $c_1 \prec c_2 \prec \ldots \prec c_N$ of length N such that c_1 does not have any predecessor.

(ii) *Partition* $[0, C_{max}]$ *into* $K{+}1$ *parts,* $\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_K$. Recall that $K = O(\log m / \log \log m)$ is the number of groups for machines by the group assignment rule as we describe in the previous section. Let $\mathcal{T}_0$ denote the union of the time intervals during which tasks of chain $\mathbb{C}$ are being processed. Consider the time interval between the end of task c_{j-1} and the start of task c_j for $j = 2, 3, \ldots, N$, and assign it to $\mathcal{T}_k$ where $M_k = f(c_j)$. As a set of time intervals, $\mathcal{T}_k$ can be possibly empty or have more than one time interval. Essentially, $\mathcal{T}_k$ is a set of time intervals that tasks in the terminal chain $\mathbb{C}$ assigned to machines in group M_k have to wait before being processed. In such a fashion, we define $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K$ since $f(\cdot)$ maps each task to one of the K machine groups. The length of the union of $\mathcal{T}_i$ for $i = 0, 1, \ldots, K$ is the makespan.

(iii) *Bound the idle time in between tasks.* Consider a task c_j assigned to machine $h(c_j)$. For each machine $i \in f(c_j)$, let $E(c_{j-1}, c_j, i)$ denote a union of disjoint empty time intervals on machine i between the end time of task c_{j-1} and the start time of task c_j. Between the end time of task c_{j-1} and the start time of task c_j, there can be multiple tasks being processed on machine i in serial, possibly resulting in more than one idle time interval on machine i during that time interval $E(c_{j-1}, c_j, i)$. Precedence constraints between task pairs can also possibly make a successor wait before it gets started. Regardless of the reason for idle time between tasks, each task can not possibly start earlier on any machine in the assigned group due to the greedy feature of GETF. Thus the length of $E(c_{j-1}, c_j, i)$ is bounded above by the communication time between task c_{j-1} and task c_j, i.e.,

$$|E(c_{j-1}, c_j, i)| \leq \frac{w_{c_{j-1}, c_j}}{s_{h(c_{j-1}), i}} \quad \forall i \in f(c_j).$$

This is true because if it were not the case, then task c_j could have started earlier on machine i. Note that the end time of task c_j could possibly be earlier if it were allowed to be scheduled on a faster machine with a slightly bigger communication delay, since the processing speeds of machines in the same group vary.

Let e_i be idle time on machine i in group M_k during the time interval $\mathcal{T}_k$, and let $\bar{e}_k$ be maximum idle time on any machine in group M_k during the time intervals $\mathcal{T}_k$, i.e., $e_i \leq \bar{e}_k$ for all $i \in M_k$. Thus,

$$\sum_{k=1}^{K} \bar{e}_k \leq \sum_{j=2}^{N} \frac{w_{c_{j-1},c_j}}{\min_{i' \in f(c_j)} s_{h(c_{j-1}),i'}} \leq \sum_{j=2}^{N} \frac{w_{c_{j-1},c_j}}{\bar{s}(c_{j-1}, c_j)}. \tag{2.4}$$

(iv) *Bound the makespan.* For $1 \leq k \leq K$, the total speed of machines in group M_k is

$$s(M_k) = \sum_{i \in M_k} s_i.$$

Denote the total length of the intervals in $\mathcal{T}_k$ by t_k. There must be at least a sum of $(t_k - e_i)\, s_i$ units of processing done on each machine i in group M_k during the time intervals $\mathcal{T}_k$. Thus for $1 \leq k \leq K$,

$$\sum_{i \in M_k} (t_k - e_i)\, s_i \leq \sum_{j: f(j) = M_k} w_j.$$

Therefore,

$$t_k \leq \frac{\sum_{j:f(j)=M_k} w_j}{s(M_k)} + \frac{\sum_{i \in M_k} e_i s_i}{s(M_k)}. \tag{2.5}$$

We now bound C_{max}:

$$\begin{aligned}
C_{max} = \quad & \sum_{k=1}^{K} t_k + t_0 \\
\leq \quad & \sum_{k=1}^{K} \left(\frac{\sum_{j:f(j)=k} w_j}{s(M_k)} + \frac{\sum_{i \in M_k} e_i s_i}{s(M_k)} \right) + \\
& \sum_{c_j \in \mathbb{C}} \frac{w_{c_j}}{s_{h(c_j)}} \qquad (2.6a) \\
\leq \quad & P + \sum_{k=1}^{K} D_k + \sum_{k=1}^{K} \bar{e}_k \frac{\sum_{i \in M_k} s_i}{s(M_k)} \\
= \quad & P + \sum_{k=1}^{K} D_k + \sum_{k=1}^{K} \bar{e}_k \\
\leq \quad & P + \sum_{k=1}^{K} D_k + C, \qquad (2.6b)
\end{aligned}$$

where (2.6a) is due to (2.5) and (2.6b) is due to (2.4). □

2.5.2 Proof of Theorem 2.4.1

In order to apply the Separation Principle to prove Theorem 2.4.1, we need to prove bounds on P and $\sum_{k=1}^{K} D_k$ in the case of the group assignment rule defined in Section 2.3. For this, we consider the scheduling problem with zero communication time. Note that the design of group assignment function $f_{\mathsf{mksp}}(\cdot)$ is based on the optimal solution x^* of the relaxed LP for a scheduling problem with zero communication time, hence the upper bounds for both P and $\sum_{k=1}^{K} D_k$ are associated with the optimal objective of the relaxed LP in the setting with zero communication time as well.

The bounds of P and $\sum_{k=1}^{K} D_k$ are given in the following two lemmas, which are adapted from results in [10]. Theorem 2.4.1 follows directly from these two lemmas, the Separation Principle, and the fact that $T^* \leq \mathrm{OPT}^{(i)}$, where T^* is the optimal solution to the LP.

Lemma 2.5.1. $P \leq 2\gamma T^*$.

Proof. Recall that $x^*_{M',j} = \sum_{i\in M'} x^*_{i,j}$ and ℓ_j as the largest group index such that at least more than half of tasks are assigned to machines in groups $M_\ell, \ldots, M_K$.

For every task j and any machine $i \in f(j)$, by definition of the largest index ℓ_j,

$$\sum_{k=1}^{\ell_j} x^*_{M_k,j} > \frac{1}{2}. \tag{2.7}$$

Thus,

$$\begin{aligned}
\sum_{i'\in M} \frac{x^*_{i',j}}{s_{i'}} &= \sum_{k=1}^{K} \sum_{i'\in M_k} \frac{x^*_{i',j}}{s_{i'}} && \text{(2.8a)}\\
&\geq \sum_{k=1}^{\ell_j} \sum_{i'\in M_k} \frac{x^*_{i',j}}{s_{i'}} \\
&\geq \frac{1}{2}\gamma^{-\ell_j} && \text{(2.8b)}\\
&\geq \frac{1}{2\gamma s_i}, && \text{(2.8c)}
\end{aligned}$$

where (2.8b) is due to (2.7) and the fact that processing speed of machine i' in group M_k for task j is at most γ^{ℓ_j} for $k \leq \ell_j$, and (2.8c) is due to the fact that processing speed of machine i in group $f(j)$, whose group index is not smaller than ℓ_j, is at least $\gamma^{\ell_j - 1}$. Using this, we can bound P as follows:

$$\begin{aligned}
P &= \sum_{c_j \in \mathbb{C}} \frac{w_{c_j}}{s_{h(c_j)}} \\
&\leq 2\gamma \sum_{c_j \in \mathbb{C}} w_{c_j} \sum_{i' \in M} \frac{x^*_{i',c_j}}{s_{i'}} && \text{(2.9a)} \\
&\leq 2\gamma \sum_{c_j \in \mathbb{C}} C^*_{c_j} && \text{(2.9b)} \\
&\leq 2\gamma T^*, && \text{(2.9c)}
\end{aligned}$$

where (2.9a) is due to (2.8), (2.9b) is due to constraint (2.1d) of the LP, and (2.9c) is due to constraint (2.1c) of the LP. □

Lemma 2.5.2. $\sum_{k=1}^{K} D_k \leq 2KT^*$.

Proof. For any task j, by definition of ℓ_j, $\sum_{k=\ell_j}^{K} x^*_{M_k,j} \geq \frac{1}{2}$. Thus,

$$\begin{aligned}
\frac{1}{2s(f(j))} &\leq \sum_{k=\ell_j}^{K} \frac{x^*_{M_k,j}}{s(f(j))} \\
&\leq \sum_{k=\ell_j}^{K} \frac{x^*_{M_k,j}}{s(M_k)} && \text{(2.10)} \\
&\leq \sum_{k=1}^{K} \frac{x^*_{M_k,j}}{s(M_k)}.
\end{aligned}$$

Inequality (2.10) is due to the fact that the assigned group $f(j)$ maximizes the total speeds of machines in that group among the candidates $M_{\ell_j}, \ldots, M_K$. Thus,

$$\begin{aligned}
\sum_{k=1}^{K} D_k = \sum_{k=1}^{K} \frac{\sum_{j: f(j)=M_k} w_j}{s(M_k)} &= \sum_{j \in V} \frac{w_j}{s(f(j))} \\
&\leq 2 \sum_{j \in V} w_j \sum_{k=1}^{K} \frac{x^*_{M_k,j}}{s(M_k)} \\
&= 2 \sum_{k=1}^{K} \frac{1}{s(M_k)} \sum_{j \in V} w_j x^*_{M_k,j} \\
&\leq 2 \sum_{k=1}^{K} T^* && \text{(2.11)} \\
&= 2KT^*.
\end{aligned}$$

The total load assigned to machines in group M_k is $\sum_{j\in V} w_j x^*_{M_k,j}$ while its total speed is $S(M_k)$. Summing over machines in group M_k on both sides for constraint (2.1d) leads to (2.11). □

2.5.3 Proof of Proposition 2.4.1

We now show how the Separation Principle can be used to provide a new, simpler proof of the state-of-the-art approximation ratio of ETF in the case of identical machines. Recall that the group assignment function is not required for GETF in this case.

To prove Proposition 2.4.1, we use the same approach as we used for proving the Separation Principle. However, we can tighten the analysis in the final step of the argument. Specifically, the proof can be broken into three steps, instead of four:

(i) Define a terminal chain $\mathbb{C}$. This step is identical to the definition of a terminal chain in the proof of the Separation Principle.

(ii) Bound the idle time in between tasks. As the machines are identical in terms of processing speed, communication speed between different machine pairs are still heterogeneous due to the possible geolocations of machines.

(iii) Combine (i) and (ii) to bound the overall makespan in terms of the communication time of the terminal chain.

Compared with the proof of the Separation Principle, Step (i) defines a terminal chain in the exactly same way. In Step (ii), bounding the idle time in the case of identical machines is also similar. Step (iii) requires more work. Here, we further tighten the bound by eliminating the processing time of the terminal chain to improve the constant factor.

(i) *Define a terminal chain* $\mathbb{C}$. This step is identical to the definition of a terminal chain in the proof of the Separation Principle.

(ii) *Bound the idle time in between tasks.* Let $I(c_{j-1}, c_j)$ be the time interval between the end time of task c_{j-1} and the start time of c_i for $j = 2, 3, \ldots, N$. As we explained in the Separation Principle, there can possibly be multiple idle time intervals on a machine during the time interval $I(c_{j-1}, c_j)$. For each machine $i \in M$, define $E(c_{j-1}, c_j, i)$ as a union of disjoint empty time intervals on machine i during the time interval $I(c_{j-1}, c_j)$. For any machine i, the length of $E(c_{j-1}, c_j, i)$

is bounded above by the communication time between task c_{j-1} and task c_j, i.e.,

$$|E(c_{j-1}, c_j, i)| \leq \frac{w_{c_{j-1},c_j}}{s_{h(c_{j-1}),i}} \quad \forall i \in M, j = 2, 3, \ldots, N.$$

Otherwise task c_j could have started earlier on machine i.

(iii) *Bound the makespan.* During the time intervals $I(c_{j-1}, c_j)$ for $j = 2, 3, \ldots, N$, there must be at least $\sum_{j=2}^{N} \sum_{i=1}^{m} (|I(c_{j-1}, j_i)| - |E(c_{j-1}, c_j, i)|)$ processing units done, and it is bounded by a sum of the processing units for all the tasks except those in the terminal chain. This leads to the following bound:

$$\sum_{j=2}^{N} \sum_{i=1}^{m} \left(|I(c_{j-1}, c_j)| - |E(c_{j-1}, c_j, i)|\right) \leq \sum_{j=1}^{n} w_j - \sum_{j=1}^{N} w_{c_j}. \tag{2.12a}$$

Finally, applying (2.12a), we have

$$\begin{aligned} C_{max} &= \sum_{j=2}^{N} |I(c_{j-1}, c_j)| + \sum_{j=1}^{N} w_{c_j} \\ &\leq \frac{1}{m} \sum_{j=1}^{n} w_j + \frac{m-1}{m} \sum_{j=1}^{N} w_{c_j} + \\ &\quad \frac{1}{m} \sum_{j=2}^{N} \sum_{i=1}^{m} |E(c_{j-1}, c_j, i)| \\ &\leq \left(2 - \frac{1}{m}\right) \mathrm{opt}^{(i)} + C'. \end{aligned} \tag{2.13a}$$

The total processing time $\sum_{j=1}^{n} w_j$ divided by the number of machines m is the smallest possible makespan, i.e., $\frac{1}{m} \sum_{j=1}^{n} w_j \leq \mathrm{OPT}^{(i)}$. At the same time, the makespan of any schedule should at least cover the processing time of any chain $\mathbb{C}$ in the DAG. These two facts lead to the last inequality (2.13a).

2.5.4 Proof of Theorem 2.4.2

To establish the bound on the total weighted completion time for the group assignment rule $f_{\mathsf{twct}}(\cdot)$, we first apply the Separation Principle to separate the requirements on communication and processing times. Second, we break the tasks into subsets based on the task completion times and, for each subset, we form an LP for those tasks alone. For each such LP, we construct a feasible solution $\tilde{x}, \tilde{C}$ and $\tilde{T}$ to bound processing time of the tasks. The feasibility of $\tilde{x}, \tilde{C}$ and $\tilde{T}$ enables us to take advantage of Lemmas 2.5.1 and 2.5.2 with only a loss of an additional constant factor.

Given a schedule $\mathcal{S}$ for a DAG G, we use the same notation as in Section 2.4.2, $G(\mathcal{S}, j)$, to denote subsets of DAG. For each DAG $G(\mathcal{S}, j)$, there is a terminal chain $\mathbb{C}(\mathcal{S}, j)$ with task j as the ending task in the schedule $\mathcal{S}(j)$. Similarly, define $P(\mathcal{S}, j)$ as a sum of the processing time along the terminal chain $\mathbb{C}(\mathcal{S}, j)$,

$$P(\mathcal{S}, j) = \sum_{c_j \in \mathbb{C}(\mathcal{S}, j)} \frac{w_{c_j}}{s_{h(c_j)}}, \tag{2.14}$$

and let $D_k(\mathcal{S}, j)$ denote the total load assigned to machines in group M_k in DAG $G(\mathcal{S}, j)$,

$$D_k(\mathcal{S}, j) = \frac{\sum_{j:j \in G(\mathcal{S}, j), k \in f(j)} w_j}{s(M_k)}. \tag{2.15}$$

For every DAG $G(\mathcal{S}, j)$ associated with schedule $\mathcal{S}_j$ for $1 \leq j \leq n$, we are able to apply Separation Principle and then combine these inequalities as follows:

$$\sum_j \omega_j C_j \leq \sum_j \omega_j \left(P(\mathcal{S}, j) + \sum_k D_k(\mathcal{S}, j) \right) + \sum_j \omega_j C(\mathcal{S}, j).$$

Both $P(\mathcal{S}, j)$ and $D_k(\mathcal{S}, j)$ are independent of the communication constraints, which enables us to take advantage of any group assignment rule.

Using the group assignment rule $f_{\mathsf{twct}}(\cdot)$ helps further tighten the bound. To show this, we first divide the n tasks into Q sets based on $q(j)$, which can be viewed as a rough estimate of the completion time of task j. For the qth interval, we define $\mathcal{J}_q$ as a set of tasks such that $q(j) = q$:

$$\mathcal{J}_q = \{j : q(j) = q\}.$$

In this way, we have divided the n tasks into Q sets: $\mathcal{J}_1, \mathcal{J}_2, \ldots, \mathcal{J}_Q$.

Next, for $1 \leq q \leq Q$, we construct a set of feasible solutions for LP (2.1), $\tilde{x}, \tilde{C}$ and $\tilde{T}$, for every set of tasks in $\mathcal{J}_q$, based on the optimal solution of LP (2.2), i.e., x^* and C^*. Note that $\tilde{x}$ here is the same as in equation (2.3). Since precedence constraints are preserved in constraints of the LPs, we can concatenate these schedules together to obtain a feasible schedule for all of the tasks.

Lemma 2.5.3. *Consider a set of tasks $\mathcal{J}_q$ for a fixed q. A feasible solution for LP* (2.1) *is defined by*

$$\tilde{x}_{i,j} = \sum_{t=1}^{q} \frac{x^*_{i,j,t}}{\alpha_j} \quad \forall i, j \in \mathcal{J}_q \tag{2.17a}$$

$$\tilde{C}_j = 2C^*_j \qquad \forall j \in \mathcal{J}_q \tag{2.17b}$$

$$\tilde{T} = 2^{q+1}. \tag{2.17c}$$

Proof. To show feasibility of such a candidate solution, we verify that $q, \tilde{x}, \tilde{C}$ and $\tilde{T}$ satisfy all the constraints in LP (2.1). Substitute $\tilde{x}$ into the left side of constraint (2.1a) for any task $j \in \mathcal{J}_q$, and it is clear that $\sum_i \tilde{x}_{i,j} = 1$. To validate that constraint (2.1b) is satisfied, note that $\alpha_j \geq 1/2$ by definition and so a direct substitution on the left hand side yields the right hand side due to (2.2b). Similarly, constraint (2.2c) ensures that constraint (2.1c) is satisfied and constraint (2.2f) ensures that constraint (2.1d) is satisfied. Finally, we obtain $C_j^* \leq 2^q$ by definition of $q(j)$ and thus constraint (2.1e) holds. □

Due to the similarity between group assignment rule $f_{\mathsf{mksp}}(\cdot)$ and $f_{\mathsf{twct}}(\cdot)$, we can further tighten the bound using Lemmas 2.5.1 and 2.5.2 from Section 2.5.2 directly. Combining Lemmas 2.5.1 and 2.5.3, we conclude that the total load along any chain $\mathbb{C}$ in the DAG formed by $\mathcal{J}_q$ is upper bounded by

$$\begin{aligned}\sum_{j \in \mathbb{C}} \frac{w_j}{s_{h(j)}} &\leq 2\gamma\tilde{T} \\ &= 2\gamma \cdot 2^{q+1}.\end{aligned}$$

Next, since the terminal chain $\mathbb{C}(\mathcal{S}, j)$ can be represented as a concatenation of chains in the DAGs formed by tasks in $\mathcal{J}_q$ for $1 \leq q \leq q(j)$, we have

$$\begin{aligned}P(\mathcal{S}, j) &\leq \sum_{t=1}^{q(j)} 2\gamma \cdot 2^{t+1} \\ &\leq 8\gamma \cdot 2^{q(j)}.\end{aligned}$$

Using Lemmas 2.5.2 and 2.5.3 together gives the following inequality:

$$\begin{aligned}\sum_{j \in \mathcal{J}_q} \frac{w_j}{s(f_{\mathsf{twct}}(j))} &\leq 2K\tilde{T} \\ &= 2K \cdot 2^{q+1}.\end{aligned}$$

The left side can be viewed as $\sum_k D_k$ for a DAG formed by tasks in $\mathcal{J}_q$. Since the tasks in DAG $G(\mathcal{S}, j)$ form a subset of $\cup_{t=1}^{q(j)} \mathcal{J}_q$, the following inequality holds:

$$\begin{aligned}\sum_k D_k(\mathcal{S}, j) &\leq \sum_{t=1}^{q(j)} \sum_{j' \in \mathcal{J}_q} \frac{w_{j'}}{s(f_{\mathsf{twct}}(j'))} \\ &\leq \sum_{t=1}^{q(j)} 2K \cdot 2^{t+1} \\ &\leq 8K \cdot 2^{q(j)},\end{aligned}$$

which immediately yields

$$P(\mathcal{S}, j) + \sum_k D_k(\mathcal{S}, j) \le 8(\gamma + K) \cdot 2^{q(j)}.$$

Finally, the remaining piece of the proof is to upper bound $2^{q(j)}$ with a multiplicative factor of its optimal completion time C_j^* in the LP (2.2). By definition of $q(j)$, for task j either

$$\sum_{t=1}^{q(j)-1} \sum_i x_{i,j,t}^* < \frac{1}{2} \tag{2.20}$$

or

$$C_j^* > 2^{q(j)-1}. \tag{2.21}$$

If inequality (2.20) holds, then

$$\begin{aligned} 2^{q(j)-1} &= \tau_{q(j)-1} \\ &\le 2\tau_{q(j)-1} \left(\sum_{t=q(j)}^{Q} \sum_i x_{i,j,t}^* \right) && (2.22a) \\ &\le 2 \left(\sum_{t=q(j)}^{Q} \tau_{t-1} \sum_i x_{i,j,t}^* \right) \\ &\le 2 \left(\sum_t \tau_{t-1} \sum_i x_{i,j,t}^* \right) \\ &\le 2C_j^*. && (2.22b) \end{aligned}$$

Inequality (2.22a) is due to (2.20) and the definition of $q(j)$, and constraint (2.2e) in the LP (2.2) leads to (2.22b). If inequality (2.21) is true, then

$$2^{q(j)-1} < C_j^* \le 2C_j^*.$$

In both cases, $2^{q(j)-1}$ is upper bounded by $2C_j^*$. Thus, we achieve

$$P(\mathcal{S}, j) + \sum_k D_k(\mathcal{S}, j) \le 32(\gamma + K) \cdot C_j^*.$$

Since $\sum_j \omega_j C_j^*$ is lower bounded by $\text{wOPT}^{(i)}$, we conclude that

$$\begin{aligned} \sum_j \omega_j C_j &\le \sum_j \omega_j \left(P(\mathcal{S}, j) + \sum_k D_k(\mathcal{S}, j) \right) + \sum_j \omega_j C(\mathcal{S}, j) && (2.23a) \\ &\le 32(\gamma + K) \sum_j \omega_j C_j^* + \sum_j \omega_j C(\mathcal{S}, j) && (2.23b) \\ &\le O(\log m / \log\log m) \cdot \text{wOPT}^{(i)} + \sum_j \omega_j C(\mathcal{S}, j), && (2.23c) \end{aligned}$$

which completes the proof.

2.6 Concluding Remarks

This paper studies the problem of scheduling tasks with precedence constraints on related machines with machine-dependent communication times, and addresses two long-standing open problems in the area. We introduce a new scheduler, GETF, and prove worst-case approximation ratios for it in the case of (i) scheduling to minimize the makespan and (ii) scheduling to minimize the total weighted completion time. These results represent the first progress on this problem since [27] provided a bound on the makespan under ETF in the case of identical servers and non-uniform communication time. No previous bounds exist for the case of total weighted completion time when non-uniform communication time is considered.

A variety of open questions are raised by the work in this paper. Most importantly, while we have provided theoretical bounds on the performance of GETF, it is also important to investigate how GETF performs in real settings via an implementation study. GETF could be particularly powerful in the context of large-scale machine learning platforms, where workflows are typically specified as DAGs. As part of such a study, it would be interesting to understand how to best choose a tie-breaking rule, how to adjust the group assignment rules for the best performance, and how various choices for these rules compare with heuristics that have been suggested in the literature. Further, it will be important to see if it is possible to obtain some theoretical results characterizing how the optimal choices for these rules depend on properties of real-world workloads. Moreover, it will also be interesting to extend the results of this work to stochastic settings, e.g., when task sizes are unknown.

On the analytic side, it will be interesting to discover other applications of the Separation Principle. It may be possible to revisit other scheduling problems for precedence-constrained tasks and obtain more general results because of the separation this result provides. Further, it is possible to consider other performance measures, such as energy usage and resource augmentation, using the Separation Principle.

Chapter 3

ENERGY-AWARE SCHEDULING IN CLOUD AI SYSTEMS

3.1 Introduction

This chapter seeks to develop energy-aware scheduling policies for precedence-constrained tasks that arise in modern machine learning platforms. The problem of how to optimally schedule a job made up of tasks with precedence constraints has been studied for decades. The initial work on this scheduling problem arose in the context of scheduling jobs on multi-processor systems [25]. Today this problem attracts attention due to the prominence of large-scale, general-purpose machine learning platforms, e.g., Google's TensorFlow [4], Facebook's PyTorch [5], and Microsoft's Azure Machine Learning (AzureML) [6]. Machine learning jobs on the cloud are often expressed as a directed acyclic graphs (DAG) of precedence-constrained tasks, and how these jobs are scheduled to run on clusters of machines on the cloud is very crucial to the performance of the system [4]. Another timely example of scheduling precedence-constrained tasks is the parallelization of training and evaluation of large complex neural networks in heterogeneous clusters that consist of CPUs, GPUs, TPUs, etc. This *device placement* problem has attracted considerable attention in recent years [56–58].

Traditionally, computational efficiency has been the only focus of works studying how to schedule precedence-constrained tasks, e.g., the goal is to complete the tasks as soon as possible given a fixed set of heterogeneous machines. The most common metric in the literature is total weighted completion time, i.e., a weighted average of completion time of tasks. The mean response time is a special case of total weighted completion time (via assigning equal weights to all the tasks), as is the makespan (via adding a dummy node of weight one as the final task with all other tasks assigned to weight zero). For these performance measures, significant progress has been made in recent years. New results have emerged providing policies with poly-logarithmic approximation ratios in increasingly general settings, including settings focused on makespan and total weighted completion time, settings with heterogeneous related machines, and settings with uniform and machine-dependent communication times [10–12, 21, 59]. Details on these results can be found in the Related Literature section.

However, the increasing scale of machine learning jobs has brought questions about the energy usage of such jobs to the forefront. Today, the emissions of training an AI model can be as high as five times the lifetime emission of a car [13]. The computation required for deep learning has been doubling every 3.4 months, resulting a 300,000x increase from 2012 to 2018 [14, 15]. Indeed, the energy cost for an individual data center is on the order of million dollars, and it has become a significant portion of operating cost for cloud platforms [60]. However, there is an inherent conflict between boosting performance and reducing energy consumption, i.e., a larger power budget, in general, allows a higher performance in practice. Thus, it is urgent to study how we can efficiently schedule machine learning jobs with both performance and energy consumption in mind. Balancing these performance measures and energy usage is crucial to industry as well as societal goals of making cloud computing carbon neutral [61].

There has been considerable progress toward understanding how to schedule to balance performance and energy measures in simple settings, e.g., single and multi-server settings without dependencies between tasks [62–65]. A focus within this line of work is on the question of co-designing scheduling of tasks and speed scaling of servers. On an algorithmic level, there are three common choices to conserve power consumption.

(a) Static speed: a server adopts a constant speed of choice to balance between performance and energy budgets.

(b) Gated static speed (a.k.a. sleep states, shutting down): a server transitions to low power mode (with speed set to zero) when it is idle; otherwise the server chooses a constant speed.

(c) Speed scaling: a server has freedom to adjust speeds dynamically when executing tasks.

Though the gated speed scheme is easier to implement in practice, speed scaling brings more potential to conserve energy by adjusting speeds of a task depending on the priority of the task. If one task needs to be prioritized, then the scheduler assigns a fast speed to run the task at the cost of a high power consumption.

While there has been progress on studying speed scaling in simple scheduling problems, the question of how to balance energy usage with traditional performance metrics, such as total weighted completion time, in more complex settings where there are dependencies among tasks is a challenging open question. In fact, scheduling precedence-constrained tasks is NP-hard even when ignoring power consumption, i.e., both the goal of partitioning the jobs across machines and of scheduling the jobs among a group of machines are NP-hard. Further, speed scaling adds considerable difficulty to the question of how to schedule tasks optimally to balance energy and performance. When speed scaling is not considered, the relaxed version of the scheduling problem can be formulated as a mixed integer linear program (MILP). Thus, there are many off-shelf solvers that work well for these problems on a relatively small-scale in practice. In the presence of speed scaling, solving for the optimal schedule even for small-scale problems becomes even more complex and computationally expensive in practice because the optimization is no longer linear. Given the hardness of the problem, a natural question is: *can we design a scheduler and speed scaling policy for precedence-constrained tasks that is provably near-optimal for a linear combination of performance and energy consumption?*

Related Literature. In recent years, due to the successful deployment of machine learning algorithms under different contextual scenarios, the design and optimization of large-scale machine learning platforms attracts extensive attention from both academia and industry community. One of the fundamental challenge is to balance between system performance and energy usage while scheduling machine learning jobs with precedence constrains.

Significant progress has been recently made towards the goal of maximizing performance while scheduling precedence-constrained tasks if energy concerns are ignored. Under the related machines model, i.e., $Q|prec|\sum \omega_j C_j$, a Speed-based List Scheduling algorithm was proposed to obtain a $O(\log m)$-approximation [9]. Later, an improvement to $O(\log m/\log\log m)$ was made in 2017 by [10] for both objectives: makespan and total weighted completion time. If communication delays are assumed to be fixed (uniform), two groups of researchers independently made progress towards a multiplicative approximation algorithm with logarithmic guarantees by adopting different approaches [11, 12, 59]. When it comes to incorporate non-uniform communication delays for the first time, i.e., $Q|prec, c_{i,j}|\sum \omega_j C_j$, Generalized Earliest Time First (GETF) was proposed in [21] to achieve a worst-case bound for both makespan and total weighted completion time, and it reduces to the state of the art results in the case of zero communication delays.

There has been a torrent of works studying power management if precedence constraints are ignored, including settings with sleep states [66–68], settings that consider speed scaling, settings with single servers [63, 69–72], and settings with multiple servers [64, 65, 73]. A comparison of different energy conservation schemes have also received considerable attention as well [62, 74]. Further, diverse performance measures have been considered in the literature, including deadline feasibility [63, 75, 76], flow time [77–79], etc. We refer to [80] for a comprehensive survey of related scheduling problems.

However, when both performance and energy goals are considered for tasks with precedence constraints, much less is known. The closest work to ours is by [81]. They consider the problem of scheduling precedence-constrained tasks to minimize the makespan subject to an energy constraint, and obtain a poly-logarithmic approximation algorithm by reducing the problem to the problem $Q|prec|C_{max}$. However, their technique does not apply to our setting for two reasons. First, we consider a general objective, which is a linear combination of performance and energy consumption, instead of focusing on the constrained budget problem; Second, total weighted completion time is a much more general choice for performance measure given that makespan is a special case of it. Thus it remains unknown if we can design a scheduler for precedence-constrained tasks to minimize a combination of performance and energy measure under these general settings.

Further, there are tremendous works on the design of heuristics for the goal of balancing both performance and energy measures [82–88]. We focus on theoretical results, and thus do not dive into the details of these heuristics.

3.2 Model

We study the problem of scheduling a job made of a set $\mathcal{V}$ of n tasks on a system consisting of a set $\mathcal{M}$ of m machines. The tasks form a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which each node j represents a task and an edge (j', j) between task j and task j' represents a precedence constraint. We interchangeably use node or task, as convenient. Precedence constraints are denoted by a partial order $\prec$ between two nodes of any edge, where $j' \prec j$ means that task j can only be scheduled after task j' completes. Let p_j represent the processing size of task j. If task j is assigned to run at a speed s_j on machine i, then it would take $\frac{p_j}{s_j}$ time units to run on machine i.

For simplicity, we assume that the DAG is connected. This is without loss of generality because, otherwise, the DAG can be viewed as multiple DAGs and the same results can be applied to each individual connected component. As a result, our results trivially apply to the case of multiple jobs. Additionally, our model assumes that each machine can process at most one task at a time, i.e., there is no *time-sharing*, and the machines are assumed to be *non-preemptive*, i.e., once a task starts on a machine, the scheduler must wait for the task to complete before scheduling any new task to this machine. This is a natural assumption in many settings, as interrupting a task and transferring it to another machine can cause significant processing overhead and communication delays due to data locality. Further, we assume that servers consume zero power when in idle, i.e., servers can run at zero speed.

Performance measure. The objective function we consider is $T + \lambda E$, a linear combination of a performance measure T and an energy/power usage E. The system operator can emphasize on either performance or energy metric as desired via a choice of weight λ. For this work, we adopt total weighted completion time as the performance measure, and our results apply to both makespan and mean response time as well. For the energy metric, our focus E is on total energy usage, which is the sum of energy usage of all tasks in the DAG, i.e., $E = \sum_{j \in \mathcal{V}} E_j$.

Speed scaling. The scheduler has the ability to scale speed of servers in order to trade off performance and energy. In our model, a server chooses speed s_j for task j. For a task running at speed s_j, its energy consumption E_j is modeled as the product of the instantaneous power $f(s_j)$ and running time t_j of that task, i.e., $E_j = f(s_j) \cdot t_j$. A common form of the power function in the literature is a polynomial, i.e., $f(s) = s^\alpha$ where $\alpha > 1$. A quadratic form is most common, so we focus on that in this work, though our main results apply more generally. Note that for any convex choice of instantaneous power function, given any optimal schedule, the server always runs at a constant speed during the execution of a single task; otherwise we can always adopt the average speed for running the task without any sacrifice on performance measure but potentially conserve more energy.

3.3 The Optimization Problem

We formally define the problem via an optimization formulation. Let $x_{i,j}$ be a a binary decision variable that indicates whether task j is assigned to machine i, i.e,

$$x_{i,j} = \begin{cases} 1, & \text{if task } j \text{ runs on machine } i, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

Let C_j denote the completion time of task j, and s_j denote the assigned speed of task j. The scheduling problem (with total weighted completion time as the performance measure) can be formulated as follows:

$$\min_{x_{i,j},C_j,s_j,T,E} \quad T + \lambda E$$

$$x_{i,j} \in \{0,1\} \qquad \forall i,j \tag{3.2a}$$

$$\sum_i x_{i,j} = 1 \qquad \forall j \tag{3.2b}$$

$$C_{j'} + \frac{p_j}{s_j} \leq C_j \qquad j' \prec j \tag{3.2c}$$

$$\sum_j \omega_j C_j \leq T \qquad \forall j \tag{3.2d}$$

$$\sum_j f(s_j)\frac{p_j}{s_j} = E \tag{3.2e}$$

$$\sum_i x_{i,j}x_{i,j'} = q_{j,j'} \qquad \forall j,j' \text{ if } j! = j' \tag{3.2f}$$

$$C_{j'} - C_j \geq \frac{p_{j'}}{s_{j'}} \text{ or } C_j - C_{j'} \geq \frac{p_j}{s_j} \qquad \text{if } q_{j,j'} = 1\ . \tag{3.2g}$$

The optimal value of this problem is denoted as OPT. Constraint (3.2b) requires every task to be scheduled on some machine. Constraint (3.2c) guarantees that for any successor predecessor pair, the successor task will not start until the predecessor completes. In Constraint (3.2d), T represents the total weighted completion time, and it can be reduced to other performance measures by assigning appropriate weights, such as makespan and mean response time. In Constraint (3.2e), E is the sum of multiplication of power function per unit time and running time. Constraint (3.2f) guarantees that $q_{j,j'} = 1$ if task j and task j' are assigned to the same machine. Any machine should not process more than one task at a time, as in Constraint (3.2g). By addition of an auxiliary binary variable $b_{j,j'}$, we can rewrite Constraint (3.2g) as follows:

$$q_{j,j'}\left(C_j - C_{j'} + \frac{p_{j'}}{s_{j'}}\right) \leq b_{j,j'}\left(T - C_{j'} + \frac{p_{j'}}{s_{j'}}\right) \forall j,j' \tag{3.3a}$$

$$b_{j,j'}C_{j'} \leq \left(C_j - \frac{p_j}{s_j}\right) q_{j,j'} \qquad \forall j,j' \tag{3.3b}$$

$$b_{j,j'} \leq q_{j,j'} \qquad \forall j,j' \tag{3.3c}$$

$$b_{j,j'} \in \{0,1\} \qquad \forall j,j'. \tag{3.3d}$$

When task j and task j' are assigned to run on the same machine, i.e., $q_{j,j'} = 1$, the auxiliary variable represents the ordering of these two tasks, i.e., $b_{j,j'} = 0$ if $j \prec j'$.

We emphasize that there are many possible formulations for the optimization problem. For example, we can derive another optimization formulation with a time-indexed program, but it would remain nonlinear due to speed constraints. Thus, we adopt the above formulation for simplicity. Given the optimization formulation, it is straightforward to conclude that searching for the global optimal of such a complex non-convex problem is hard and computationally expensive. As a result, we focus on approximation algorithms.

3.4 Algorithm Design

Inspired by the single server case, we introduce the notion of *pseudo-size* to quantify importance of tasks in the DAG. Intuitively, a task of significance should be prioritized to run while a task of low priority should be set to run at a slow speed. Our approximation algorithm combines the intuition of pseudo-size approximation and any approximation algorithm for the case of identical machines without energy concerns, e.g., [50], to produce a schedule that balances between performance and energy goals. The pseudo-size of tasks depend on various features of the precedence graph, such as degree of nodes, number of children, etc, as well as the given number of machines. In practice, experts can extract these features and feed them to an off-shelf learning algorithm to obtain an approximation.

The introduction of pseudo-size is critical in two ways. First, conditional on an approximation of pseudo-size, we are able to reduce the general problem, where it is hard to find an optimal solution, to a much simpler scheduling problem in the case of identical machines. This not only mitigates the required computations in process, but also makes it possible to compute a theoretical bound on the final schedule. Second, since the concept of pseudo-size stems from one single server scenario, it comes with an intuitive interpretation in the physical world, which is not a given in the world of learning algorithms. In practice, pseudo-size of a task quantifies magnitude of externalities it has on other tasks in the graph.

Next, we first formally define the notion of *pseudo-size*, and then propose a family of approximation scheduling algorithms based on the pseudo-size approximation.

3.4.1 The Single Server Case

We start by diving into the one server problem. As shown in [62], the optimal speed for running a task without precedence constraints is proportional to square root of the number of tasks that are waiting for the task. The same intuition holds true for one server even if there are precedence constraints among the tasks. The characterization of optimal speeds in the one server case is summarized as follows.

Example 3.4.1. *Consider n tasks with dependency to be scheduled to run on a single machine, i.e., optimization problem* (3.2) *with $m = 1$. For any given feasible ordering in which tasks are indexed with respect to the the given ordering, the optimal speed for running task j is*

$$\sqrt{(n-j+1)\cdot\frac{\omega_j}{\lambda}},$$

where $(n-j+1)$ is the number of tasks waiting for its completion.

Proof. For any feasible ordering of tasks on the machines, we assume that tasks are indexed with respect to the given ordering, i.e., $1 \prec 2 \prec \cdots \prec n$. Thus the objective function can be simplified to

$$\left(\frac{n\cdot p_1}{s_1}+\lambda\cdot p_1\cdot s_1\right)+\left(\frac{(n-1)\cdot p_2}{s_2}+\lambda\cdot p_2\cdot s_2\right)+\cdots+\left(\frac{p_n}{s_n}+\lambda\cdot p_n\cdot s_n\right).$$

Clearly, the objective is minimized only when speed of task j is set to be equal to $\sqrt{(n-j+1)/\lambda}$. For task j, $(n-j+1)$ is the number of tasks (including the task itself) that are depending on its completion. □

For a large λ that translates to an emphasis on the total energy consumption, optimal speeds tend to have a smaller magnitude and vice versa. When λ is chosen to be one, then total weighted completion time of tasks is equal to the sum of energy consumption in any given optimal schedule, i.e., equal budgets are allocated to performance measure and energy consumption for any optimal schedule. In this case, optimal speeds of a task is exactly square root of the number of dependent tasks provided that the weight ω_j of task j is 1. Via a choice of λ, the system designer can adjust the budget for performance and energy consumption as desired.

3.4.2 Pseudo-Size

The pseudo-size of a task is a measure of importance of the task with respect to its running speed. If a task has many tasks waiting for its completion, then the scheduler tends to prioritize the task via assigning a fast running speed. This parallels with the term $(n-j+1)$ in the single server case.

Definition 3.4.1. *For a given DAG $\mathcal{G}$ to be scheduled on a set of m machines, the pseudo-size β_j of task j is defined as the scaled square of the optimal speed s_j for* *that task multiplied by the scaling parameter γ_j, i.e.,*

$$\beta_j = \gamma_j \cdot (s_j)^2, \tag{3.4}$$

where $\gamma_j \triangleq \frac{\lambda}{\omega_j}$ is a constant scalar introduced by system parameters.

The scaling parameter depends on the choice of performance measure as well as the weight λ. For any relatively large-scale problem, optimal speed is almost impossible to determine for such a problem. As a result, a good approximation via an approximation of pseudo-size becomes significant.

3.4.3 Learning-Augmented Energy-Aware List Scheduling

In this section, we propose *Learning-Augmented Energy-Aware List Scheduling* for both makespan and total weighted completion time based on an approximation of task pseudo-size. The concept of task pseudo-size offers a new and intuitive perspective to learn the optimal speeds via an approximation of pseudo-size. The data-driven approach enables us to adapt to different scenarios quickly, e.g., different structures of DAGs, heterogeneous task sizes and number of machines, etc. Additionally, as a two-stage algorithm, Learning-Augmented Energy-Aware List Scheduling enables us to separate the evaluation of these two stages, and combine them together to form a worst-case bound under some conditions. Our algorithm proceeds as follows.

Algorithm 3.2 Learning-Augmented Energy-Aware List Scheduling.

INPUT: DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and number of machines m.
OUTPUT: schedule $\mathcal{S}$ and speed scaling policy $\{s_j\}$.

First Stage

1: Approximate task pseudo-size $\{\beta_j\}$ via learning.

Second Stage

2: Derive the associated running speed $\{s_j\}$.
3: Transform the initial problem into identical machine case by assigning a new task size $p'_j \leftarrow p_j/s_j$ for every task.
4: Apply list scheduling. ▹ LS varies depending on choice of C_{max} and $\sum \omega_j C_j$.

We divide our algorithm into two stages. During the first stage, we take advantage of any off-shelf data-driven algorithms to learn an approximation of task pseudo-size from workload data. The learned model might vary depending on choice of different performance measures. During the second stage, we exploit the learned pseudo-size and derive the associated speed for every single task in the DAG. Given the running speeds, we are able to transform the problem into the case of identical machines and adopt any appropriate listing scheduling to compute a final schedule. Note that the priority list adopted for makespan and total weighted completion time can be different in order to achieve a good theoretic guarantee. In short, any greedy list scheduling can be employed for makespan while list scheduling has to be further refined in the case of total weighted completion time. Next, we further describe the list scheduling adopted from [50] for total weighted completion time.

Total Weighted Completion Time. Given a pseudo-size approximation $\{\beta_j\}$, we transform the problem into the identical machines problem by assigning a new task size for task j:

$$p'_j \leftarrow p_j \cdot \sqrt{\gamma_j/\beta_j}. \tag{3.5}$$

The pseudo-size of a task quantifies magnitude of externalities that it has on other tasks waiting for its completion. Equation (3.5) scales task sizes to account for externalities. After the transformation, the list scheduling adopted for total weighted completion time is built upon a linear program [50].

$$\min_{C_j} \sum_{j \in \mathcal{V}} \omega_j C_j$$

$$C_j \geq C_{j'} + p'_j \qquad \forall j' \prec j \tag{3.6a}$$

$$\sum_{j \in \mathcal{F}} p'_j C_j \geq \frac{1}{2m} \left(\sum_{j \in \mathcal{F}} p'_j \right)^2 + \frac{1}{2} \sum_{j \in \mathcal{F}} p'^2_j \qquad \forall \mathcal{F} \subset \mathcal{V} \tag{3.6b}$$

The objective of the linear program is to minimize total weighted completion time. Constraint (3.6a) enforces precedence constraints among any predecessor-successor pair. Constraint (3.6b) is a weaker version of no time sharing requirement, i.e., one machine can only process at most one task at a time.

This linear program doe not serve to find an optimal solution for the identical machine problem. Instead, we further build upon its solutions to construct a priority list. Let $\{C_j^{LP}\}$ denote the optimal solutions for LP (3.6). The priority list is constructed with respect to α-points of tasks based on the LP solution, which are defined as below for $0 \leq \alpha \leq 1$. The α-point M_j^{LP} of task j based on the LP solution is defined as follows:

$$M_j^{LP} \triangleq C_j^{LP} - \alpha \cdot p'_j.$$

Once we compute the α-points of tasks, we obtain the priority list $\mathcal{L}$ by indexing these tasks with respect to magnitude of α-points in a non-decreasing order, i.e., a task with a large α-point value has a high priority. In practice, we can search for a near-optimal α to further optimize over the overall objective, but for the purpose of this work, we can assume that $\alpha = \frac{1}{2}$ to achieve the final bound. We conclude this section by giving the details of list scheduling in the case of total weighted completion time as in Algorithm 3.3.

Algorithm 3.3 Learning-Augmented Energy-Aware List Scheduling: Total Weighted Completion Time.

INPUT: DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and number of machines m.
OUTPUT: schedule $\mathcal{S}$ and speed scaling policy $\{s_j\}$.

First Stage
1: Approximate task pseudo-size $\{\beta_j\}$ via learning.

Second Stage
2: Transform the initial problem into the identical machine problem by assigning a new task size $p'_j \leftarrow p_j/s_j$ where $s_j = \sqrt{\beta_j/\gamma_j}$ for each task.
3: Construct a priority list $\mathcal{L}$ via LP (3.6). ▷ List scheduling starts.
4: **for** task j in the priority list $\mathcal{L}$ **do**
5: Start task j on any available machine at the earliest possible time.
6: **end for**

3.5 Results

Learning-augmented algorithms usually leverage machine learning tools to make a prediction in a data-driven fashion, and then feed this prediction as the actual input to an algorithm. The goal is to design such an algorithm that incorporates these predictions and provides theoretic guarantees based on quality of learning-based predictions. These learning-augmented algorithms have been applied in various settings, e.g., predicting task size for a scheduler [89], improving online algorithms [90, 91], ski-rental etc. Similarly in this work, we take the inspiration of pseudo-size from the single server case, and construct a scheduler and speed scaling policy via building upon the pseudo-size approximation. This allows our algorithm to adapt to workload data. Assuming a good approximation of task pseudo-size, we are able to provide worst-case bounds on a linear combination of performance and energy consumption when either makespan or total weighted completion time is considered. Even when an pseudo-size approximation is bad, our results characterize how the overall worst-case performance depends on quality of task pseudo-size approximation.

3.5.1 Total Weighted Completion Time

The scheduling algorithm makes use of the pseudo-size in two stages. First, we use learning-based algorithms to learn an approximation of the pseudo-size $\{\beta_j\}$. Second, we exploit the pseudo-size approximation to transform the problem into a scheduling problem with identical machines via assigning a new task size of $p_j \cdot \sqrt{\gamma_j/\beta_j}$ for task j. Then we can take advantage of the approximation algorithm in [50] for the case of identical machines. We denote the schedule produced by the above algorithm as schedule S. Let $T(S)$ and $E(S)$ denote the performance and energy consumption of the schedule S respectively. We use $f(S)$ to denote the linear combination of performance and energy consumption for schedule S, i.e., $f(S) = T(S) + \lambda \cdot E(S)$. Our main result is the following learning-augmented approximation ratio.

Theorem 3.5.1 (Total Weighted Completion Time)**.** *Given a good pseudo-size approximation for tasks in a DAG, i.e.,* $(1-\epsilon_j^-)\cdot\beta_j^* \le \beta_j \le (1+\epsilon_j^+)\cdot\beta_j^*$*, then any schedule S created by the above algorithm satisfies:*

$$f(S) \le \max\left\{\max_j \frac{4}{\sqrt{1-\epsilon_j^-}}, \max_j \sqrt{1+\epsilon_j^+}\right\} \cdot OPT.$$

Note that the approximation ratio is small as long as the pseudo-size is approximated well.

3.5.2 Makespan

If makespan is adopted for performance measure rather than total weighted completion time, during the second stage of the scheduling algorithm, we transform the problem into the identical machine problem and then apply a list scheduling algorithm. For a produced schedule S, we prove the following learning-augmented approximation ratio.

Theorem 3.5.2 (Makespan)**.** *Given a good pseudo-size approximation for tasks in a DAG, i.e.,* $(1-\epsilon_j^-)\cdot\beta_j^* \le \beta_j \le (1+\epsilon_j^+)\cdot\beta_j^*$*, then any schedule S created by the* *above algorithm satisfies:*

$$f(S) \leq \max\left\{\max_j \frac{2-1/m}{\sqrt{1-\epsilon_j^-}}, \max_j \sqrt{1+\epsilon_j^+}\right\} \cdot OPT.$$

To the best of our knowledge, this provides the first bound when minimizing a linear combination of performance and energy consumption, which is more general compared to settings focused on minimizing the makespan for a given energy budget as in [81]. Moreover, their techniques in [81] does not apply to the case of total weighted completion time.

3.6 Concluding Remarks

In this chapter, we consider the problem of scheduling precedence-constrained tasks to minimize a linear combination of performance and energy consumption. Our main results provide worst-case guarantees when performance is either makespan or total weighted completion time. Even without a general pseudo-size approximation, one can easily achieve a good approximation for certain DAGs, e.g., a DAG with all joins or forks. A variety of research questions have been raised by our results. Most importantly, how good is an approximation of task pseudo-size for general DAGs via learning from workload data? Moreover, for particular structures of DAGs, is it possible to further refine the bounds by incorporating properties of these DAGs? On the analytic side, it would be also interesting to study how our learning-augmented algorithm compares with pure learning algorithms in practice.

3.A Appendix

3.A.1 A Proof of Theorem 3.5.1

Proof. Our proof proceeds in three steps.

(i) *Bound total energy consumption E.* Assuming a good approximation of task pseudo-size, i.e., $(1-\epsilon_j^-)\cdot\beta_j^* \leq \beta_j \leq (1+\epsilon_j^+)\cdot\beta_j^*$, we can translate this approximation into a bound on speeds adopted in the algorithm:

$$\sqrt{1-\epsilon_j^-}\cdot s_j^* \leq s_j \leq \sqrt{1+\epsilon_j^+}\cdot s_j^*.$$

Given that energy consumption of task j is

$$E_j = f(s_j)\cdot t_j = p_j \cdot s_j \leq \sqrt{1+\epsilon_j^+}\cdot E_j^*,$$

we get an upper bound on the total energy consumption:

$$E(S) \le \max_j(\sqrt{1+\epsilon_j^+}) \cdot E^*. \tag{7}$$

(ii) *Bound total weighted completion time T.* According to the main results in [50], for any schedule generated by the LP (3.6), we get

$$C_j - p'_j \le 2M_j^{LP} + 2(C_j^{LP} - p'_j),$$

which can be further relaxed as below:

$$C_j \le 4C_j^{LP} - 2p'_j. \tag{8}$$

Let $T\,(p_1, \cdots, p_n)$ denote the optimal total weighted completion time for the identical machines problem with task sizes $(p_1, \cdots, p_n)$. Thus

$$\begin{aligned}
\sum_j \omega_j C_j &\le 4 \sum_j \omega_j C_j^{LP} && \text{(9a)}\\
&\le 4 \cdot T\left(p'_1, \cdots, p'_n\right) && \text{(9b)}\\
&\le 4 \cdot T\left(\frac{p_1}{s_1^*} \cdot \frac{1}{\sqrt{1-\epsilon_1^-}}, \cdots, \frac{p_n}{s_n^*} \cdot \frac{1}{\sqrt{1-\epsilon_n^-}}\right) && \text{(9c)}\\
&\le 4 \cdot T\left(\frac{p_1}{s_1^*} \cdot \max_j \frac{1}{\sqrt{1-\epsilon_j^-}}, \cdots, \frac{p_n}{s_n^*} \cdot \max_j \frac{1}{\sqrt{1-\epsilon_j^-}}\right) && \text{(9d)}\\
&= \max_j \frac{4}{\sqrt{1-\epsilon_j^-}} \cdot T\left(\frac{p_1}{s_1^*}, \cdots, \frac{p_n}{s_n^*}\right) && \text{(9e)}\\
&= \max_j \frac{4}{\sqrt{1-\epsilon_j^-}} \cdot \sum_j \omega_j C_j^*. && \text{(9f)}
\end{aligned}$$

Inequality (9a) is a further relaxation of Inequality (8). As LP (3.6) only covers partial constraints for the scheduling problem of identical machines, its optimal objective gives a lower bound as in Inequality (9b). Consider the scheduling problem in the case of identical machines: if we increase task sizes for each task, then total weighted completion time will increase as well. This explains why Inequality (9c) and Inequality (9d) hold true. If we scale the task size by a constant factor, then total weighted completion time will scale correspondingly as in Equality (9e). This can be easily verified if we go over the constraints in the optimization formulation for the scheduling problem in the case of identical machines. Equality (9f) says that the general scheduling problem, when either performance or energy consumption are considered, can be reduced to the problem of minimizing total weighted completion time alone when task sizes are re-scaled.

(iii) *Combine (i) and (ii) to bound the overall objective.* By combining Inequality (7) and Inequality (9), we achieve

$$T(S) + \lambda E(S) \leq \max \left\{ \max_j \frac{4}{\sqrt{1-\epsilon_j^-}}, \max_j \sqrt{1+\epsilon_j^+} \right\} \cdot OPT.$$

□

3.A.2 A Proof of Theorem 3.5.2

Proof. Our proof proceeds in three steps.

(i) *Bound total energy consumption E.* This step is no different than that in the case of total weighted completion time. Assuming a good approximation of task pseudo-size, i.e., $(1-\epsilon_j^-) \cdot \beta_j^* \leq \beta_j \leq (1+\epsilon_j^+) \cdot \beta_j^*$, we can translate this approximation into a bound on speeds adopted in the algorithm:

$$\sqrt{1-\epsilon_j^-} \cdot s_j^* \leq s_j \leq \sqrt{1+\epsilon_j^+} \cdot s_j^*. \tag{10}$$

Given that energy consumption of task j is

$$E_j = f(s_j) \cdot t_j = p_j \cdot s_j \leq \sqrt{1+\epsilon_j^+} \cdot E_j^*, \tag{11}$$

we get an upper bound on the total energy consumption:

$$E(S) \leq \max_j (\sqrt{1+\epsilon_j^+}) \cdot E^*. \tag{12}$$

(ii) *Bound makespan T.* We start with constructing a chain backwards. In the given schedule S, we start with the task that ends last. Among its predecessors, we choose one of the tasks with the latest completion time. There can be multiple candidate tasks that end at the same time. Though the overall performance might vary depending on different tie-breaking rules in practice, random tie-breaking rules are good enough for the purpose of this proof. We note that there is no idle gaps in between the execution of any pair of two predecessor-successor tasks in such a chain; otherwise, the successor task could have started earlier. We continue doing so until we reach the top of the DAG. In such a way, we construct a chain $\mathcal{J}$ such that every machine is busy during the complement time of execution process of this chain of tasks. Thus,

$$T(S) \leq \sum_{j\in\mathcal{J}} p'_j + \frac{\sum_{j\in\mathcal{V}} p'_j - \sum_{j\in\mathcal{J}} p'_j}{m}$$
$$= (1-\frac{1}{m})\sum_{j\in\mathcal{J}} p'_j + \frac{1}{m}\sum_{j\in\mathcal{V}} p'_j.$$

Let $C_{max}(p_1,\cdots,p_n)$ denote the optimal makespan for the identical machines problem with task sizes $(p_1,\cdots,p_n)$. Then

$$T(S) \leq (1-\frac{1}{m})\cdot C_{max}\left(p'_1,\cdots,p'_n\right) + C_{max}\left(p'_1,\cdots,p'_n\right) \tag{14a}$$
$$\leq (2-\frac{1}{m})\cdot C_{max}\left(\frac{p_1}{s^*_1}\cdot\frac{1}{\sqrt{1-\epsilon^-_1}},\cdots,\frac{p_n}{s^*_n}\cdot\frac{1}{\sqrt{1-\epsilon^-_n}}\right) \tag{14b}$$
$$\leq (2-\frac{1}{m})\cdot C_{max}\left(\frac{p_1}{s^*_1}\cdot\max_j\frac{1}{\sqrt{1-\epsilon^-_j}},\cdots,\frac{p_n}{s^*_n}\cdot\max_j\frac{1}{\sqrt{1-\epsilon^-_j}}\right) \tag{14c}$$
$$= \max_j\frac{2-1/m}{\sqrt{1-\epsilon^-_j}}\cdot C_{max}\left(\frac{p_1}{s^*_1},\cdots,\frac{p_n}{s^*_n}\right). \tag{14d}$$

As in Inequality (14a), optimal makespan should be lower bounded by either running time of any chain of tasks in the DAG or completion time as if there were no precedence constraints. Consider the scheduling problem in settings with identical machines: if we increase size of any task, the optimal makespan should not become any shorter, if not any longer. This is why Inequality (14b) and Inequality (14c) hold true. Further, if we scale task size by a constant factor, then optimal makespan should scale correspondingly. Again, this can be easily seen if we write the linear optimization formulation of the scheduling problem in the case of identical machines.

(iii) *Combine (i) and (ii) to bound the overall objective.* By combining Inequality (12) and Inequality (14d), we achieve

$$T(S) + \lambda E(S) \leq \max\left\{\max_j\frac{2-1/m}{\sqrt{1-\epsilon^-_j}}, \max_j\sqrt{1+\epsilon^+_j}\right\}\cdot OPT. \tag{15}$$

□

Part II

Market Design

Chapter 4

THE PRIVACY PARADOX IN DATA ACQUISITION

4.1 Introduction

There is a fundamental discrepancy between privacy attitudes and the behaviors of users online: users claim to be concerned about their privacy, but do little to protect privacy in their actions. More specifically, users express their concerns about privacy, including the ambiguous distribution of data and its use by third party [92–94]; however, when choosing services, users mainly focus on the popularity, convenience, price, etc, despite the potential risk of data misuse [95, 96]. This phenomenon is known as the *privacy paradox* [97], and understanding the reasons behind this paradox and its consequences for the design of online platforms is an important goal for both computer scientists and economists.

The privacy paradox is at the root of the behavior of individuals in modern online data marketplaces. Online platforms gather data on billions of individuals in order to personalize advertising and customize other aspects of their systems. However, such usage tends to provide little direct benefits for the users, a fact that is used as indirect evidence to argue that users provide a small value on privacy [98]. Such an argument ignores the impact that an individual's participation decision has on others. In particular, when an individual shares her data, it is not just her privacy that is compromised; the privacy of other individuals whose data is correlated with hers is also compromised. Thus, these other individuals are more likely to share their own data given that some has already been leaked [99]. This simple, but often overlooked issue is at the root of the privacy paradox. Information leakage due to correlation has been shown to lead to oversharing since each individual overlooks their own privacy concerns as a result of the negative externalities created by others' revelation decisions. Thus, information leakage leads to the potential for significant economic and social inefficiency in data marketplaces.

In this paper, *we study the impact of privacy concerns and information leakage on the design of data markets.* Specifically, we study the task of designing mechanisms for obtaining verifiable data from a population for a statistical estimation task, such as estimating the expected value of some function of the underlying data.

The goal of designing mechanisms for optimal data acquisition is a core piece of the emerging literature on data marketplaces. A common motivating example is a setting where a healthcare platform is doing statistical analysis on its population of users. While some data is measured accurately from their smart devices, other desired data may be about characteristics users do not wish to provide or may vary over time. Thus the healthcare platform has to conduct a survey among the users to obtain such information accurately (e.g., giving the individual a smart device or having the individual fill out a form); however when administering such a survey, the responses are likely biased. For example, if weight is the target, then the respondents may be biased towards low-weight samples. Thus, the task of designing mechanisms to limit the bias and reduce the variance of estimates obtained from such surveys is crucial. However, such a task is challenging due to the fact that the analyst does not know the distribution of the data and has a limited budget.

A growing line of work has focused on the design of such optimal data acquisition mechanisms, e.g., [100–103]. Initiated by [100, 103], this line of work has led to the design of mechanisms for unbiased estimation with minimal variance in a variety of settings. However, in this literature it is assumed that all individuals will participate, thus unbiased estimation is possible. The trade-off between bias and variance has been ignored to this point with the exception of Chen and Zheng [102], which still assumes all individuals will participate and does not consider privacy concerns. Further, this line of work has not considered the issues created by information leakage due to correlation between the participants. Information leakage creates significant incentives for increased data sharing and thus mechanisms that do not consider it directly will suffer from undetected bias and increased variance in the obtained estimates. Modeling the incentives created by leakage potentially provides the analyst the opportunity to obtain an estimator of the same quality using a smaller budget, due to the externalities created by data correlation.

Contributions. In this paper, we provide the first characterization of an optimal mechanism for data acquisition in the setting where agents are concerned about privacy and their data is correlated with each other. As a result, information leakage due to data correlation not only contributes to an agent's privacy cost, but also to the privacy costs of others with correlated data. Additionally, the mechanism allows, for the first time, a trade-off between the bias and variance of the estimator when privacy cost is considered. This offers the analyst freedom to tailor towards an emphasis on either bias or variance of the estimator depending on the contextual goals.

Specifically, we propose a novel model for data acquisition. The novelty of our model is a result of three important components. First, we introduce the privacy cost to model impacts of data correlation. Unlike modeling data correlation on an individual level in [99], we divide the agents into different groups and assume that agents within the same group share a same correlation strength. This gives us the power to work with any granularity of choice with regard to data correlation. For example, if every group has a relatively small size of agents, then our model of data correlation shifts towards a near individual level. In addition, an agent suffers a larger privacy cost if she joins the platform than that if she does not join. The choice of our privacy cost function enables us to model all these desired properties. Second, in reality, not every agent always decides to join the platform. Thus, we introduce the notion of participation rate as the ratio of the number of agents who join the platform to the number of total agents. This further allows us to study equilibrium with respect to participation rate, which is crucial since the mechanism impacts the participation rate, which in turn impacts the bias and variance. Third, given that not every agent always joins the platform, it is not always realistic to aim for an unbiased estimator. Instead, we minimize a linear combination of bias and variance of the estimator. Via a choice of constant weights for bias and variance, we are able to balance between these two metrics of the estimator as desired.

Our main theoretical results provide a closed form solution of payment and allocation rules under a choice of equilibrium participation rate in order to achieve a truthful mechanism (Theorem 4.3.1 & Theorem 4.4.1). More specifically, we aim to minimize a linear combination of bias and variance subject to budget and truthfulness constraints. By considering a linear combination, we are able to emphasize either bias or variance of the estimator as desired. Moreover, we provide conditions for the optimality of an unbiased estimator in the case when it is possible to achieve a full participation rate, i.e., every agent decides to join the platform.

Our results offer some interesting insights about mechanisms for data acquisition. First, an unbiased estimator is possible even if the budget is relatively small because we can meet the expected budget constraint using a small selection probability. However, an unbiased estimator is not always realistic in practice. As a result, it is important to optimize the bias-variance trade-off. Second, incorporation of privacy cost due to information leakage and sharing makes it possible for the analyst to underpay the agents to acquire the same data set. This can potentially lead to a relatively small payment for data, something that is frequently seen in practice.

Last but not least, the privacy cost from leakage encourages more agents to join the platform, which coincides with the data oversharing phenomenon frequently observed in platforms today.

The design of mechanisms for data acquisition is known to be challenging even if we focus on an unbiased estimator and ignore privacy cost due to data correlation. However, obtaining our results requires overcoming additional challenges. There are two technical innovations that enable our analysis. First, we need to introduce and characterize an equilibrium with respect to the participation rate, as the participation rate is an endogenous property of the mechanism design problem. This equilibrium adds considerable complexity to the analysis, but also provides insights about how the data acquisition mechanism depends on the popularity of the platform. Second, we introduce the notion of data correlation strength to characterize the privacy cost due to information leakage. This allows us to capture the impact of data correlation on the platform. Further, depending on a choice of group sizes, we are able to model information leakage at different granularity levels as desired.

Related literature. The design of data markets has attracted a significant amount of interest in recent years. There is growing body of work studying a variety of aspects of data markets, including monetizing information via either dynamic sales or optimal mechanisms, e.g., [104, 105], exploiting personal information to improve allocation of resources in online markets, e.g., [106–108], optimal acquisition of information, e.g., [100–102], etc. For a recent survey, see [109] and the references therein. Our work broadly falls into the last line of work, and focuses on the design of a mechanism for optimal data acquisition. The study of the optimal data acquisition has attracted a growing amount of attention across both economics and computer science. In this paper, we focus on designing a truthful mechanism in order to perform a statistical estimation task. This goal has received considerable attention. However, prior work does not consider the privacy cost of the participants. Our work aims to fill the gap by considering privacy cost as an important factor for individuals' decision-making.

In more detail, the prior work on optimal data acquisition can be divided into two categories depending on whether data is verifiable or not. In the first category, individuals' utilities often directly depend on the outcome of the statistical inference, and they thus have an incentive to misreport their data, e.g., [110–116]. This is possible since there is no ground truth to verify the data. In the second category, individuals are assumed to report their data truthfully due to the ability of the analyst to verify the data, e.g., [100–103, 117, 118]. In this paper, we consider the setting where data is verifiable and so focus our discussion on prior work in that category below.

The task in this context is to purchase data from individuals whose private costs are subject to an (expected) budget constraint. This model was introduced in [100, 103]. The model assumes that data cannot be fabricated and that private costs of the participants are correlated with the data. Moreover, the participants do not derive utility or disutility from the estimation outcome. Roth and Schoenebeck [100] minimizes a bound of the worst case variance while achieving an unbiased estimator while Abernethy, Chen, Ho, and Waggoner [103] considers general supervised learning. Following these initial papers, a closed form result that directly minimizes the worst case variance is given in [101]. However, unbiased estimators are not always possible in realistic settings. When a biased estimator is considered, Chen and Zheng [102] proposes a slightly different model, in which agents arrive in an online fashion and cost distribution is not known a priori. Under this model, Chen and Zheng [102] studies a trade-off between bias and variance of the estimator rather than only focus on the unbiased estimators. This trade-off is also a core component of our work.

The prior work discussed above mostly focuses on data acquisition without considering the privacy concerns of the participants. Even when the privacy concerns are considered in the prior work, privacy cost is often interpreted as a simple cost, which does not capture the information leakage due to data correlation. In contrast, privacy cost due to information leakage as a result of data correlation is a crucial concern in the model we consider in this paper. There is a recent line of work that examines data acquisition through the lens of differential privacy [117, 119–122]. As agents might not be willing to report their data due to their privacy concerns, monetary incentives are given to encourage individuals to participate in order to balance between privacy of individuals and accuracy of estimation. However, in this line of work, the impact of information leakage is not considered and thus the practical impact of the privacy paradox is not considered.

The information leakage that results from correlation between individuals' is at the root of the privacy paradox and is a crucial factor for data markets, as has been recognized by recent work in economics, e.g.,[99, 123]. More specifically, Acemoglu, Makhdoumi, Malekian, and Ozdaglar [99] recently introduced the concept of information leakage to account for privacy cost due to other individuals' data sharing.

Inspired by this concept, we incorporate the privacy cost due to heterogeneous data correlation in our model and aim to design a truthful mechanism to balance between bias and variance subject to an expected budget constraint. The privacy concern of individuals, specifically the privacy cost due to data correlation, is essential in the design of such a mechanism. This is a critical feature of our model that is not present in any prior work on data acquisition. The setting and model we consider differ considerably from [99]. We consider the problem of optimal data acquisition and the incentives created by information leakage, which have not been considered previously in this context.

4.2 System Model

We consider an online platform consisting of an analyst and many agents. At a high level, the analyst aims to design a pricing mechanism to purchase private data from agents in order to perform a statistical estimation task, e.g., estimate the mean of the agents' data. Ideally, the analyst would like to purchase all the private data to obtain an unbiased estimator. However, given a limited budget, the analyst has to design a pricing mechanism to wisely select the data in order to balance between the bias and variance of the estimator.

To this end, we consider a family of pricing mechanisms that presents a menu to the agents. The menu consists of pairs of payments and probabilities of having agents' data selected for use in the estimation task. Given the menu, the agents report their costs and decide individually if they would like to join the platform or not. Once the agents make the decisions, the analyst selects data from an agent on the platform to purchase with the given probability selected from the menu. The analyst's goal in designing the mechanism is to determine the menu of payments and selection probabilities in order to perform a statistical estimation subject to an expected budget constraint. The form of this mechanism is classical, and adopted from, e.g., [100, 101, 103].

Considering privacy cost is critical to the design of such a mechanism. For the agents on the platform, they obtain benefit from participation but reveal some personal information through their interactions with the platform, which leads to the privacy cost. Further, when an agent's data is used, it negatively impacts not only the privacy of the agent revealing the data, but also the privacy of other agents whose data is correlated. Thus, even though some agents do not join the platform and do not directly share their data, they may still suffer a privacy cost through their

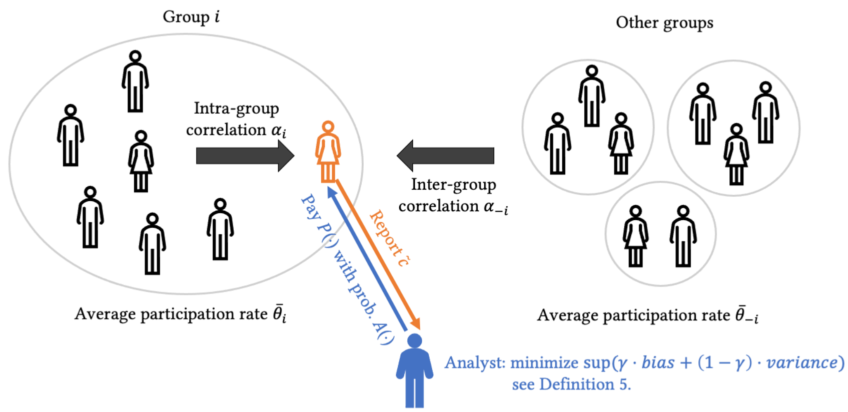


Figure 4.1: An illustration of the model from the perspective of a participating agent in group i.

peers' interactions with the platform due to data correlation. This privacy cost via *information leakage* is an important and novel feature of the model presented here.

We describe the full details of the model and the family of mechanisms we consider in the remainder of this section. Figure 4.1 provides an overview of the model.

4.2.1 Agent Model

We consider s agents that hold data of interest to the analyst. The set of agents is denoted by $\mathcal{S}$. Every agent owns a data point. By reporting her data to the platform's survey, the agent incurs an overall cost c, which is known to her but not to the analyst. The overall cost consists of a combination of reporting cost and privacy cost, where the reporting cost results from the act of reporting the data while the privacy cost comes from both data sharing and data correlation. We discuss these costs further in Section 4.2.1.

We now present how the mechanism works. First, each agent is presented a price menu by the analyst that consists of a payment rule $P(\cdot)$ and selection probability $A(\cdot)$, both of which depend on the reported cost $\tilde{c}$ of the agent. We interchangeably use selection probability or allocation rule, as convenient. Second, given the menu, an agent decides if she would like to join the platform. An agent who decides to join the platform is asked to report her cost, which determines the payment and selection probability. The payment $P(\cdot)$ is given to the agent if she joins the platform *and* her data is selected (used) by the analyst. More specifically, her data is selected with probability $A(\cdot)$, and she receives the payment only if her data is selected.

Whether an agent decides to join the platform relies on weighing the benefit of participation $w(\cdot)$, plus a potential payment, against the privacy cost that occurs as a result of her own or her peers' interactions with the platform. More specifically, for an agent on the platform, her privacy cost incurred includes both the cost from the agent herself sharing her data and the privacy cost due to her friends' sharing of possibly correlated data. We use $h(\cdot)$ to denote this combination. In contrast, if an agent does not join the platform, she still suffers a privacy cost $g(\cdot)$ due to information leakage as a result of correlation between her data and the data of those agents who do join. We use $\mathcal{N}$ to denote the set of agents who join the platform. For agent k, her utility is as follows:

$$u_i(\tilde{c}|c) = \begin{cases} -g(\cdot), & \text{if } k \notin \mathcal{N}, \\ -h(\cdot) + w(\cdot), & \text{w/ prob. } 1 - A(\cdot), \text{ if } k \in \mathcal{N}, \\ P(\cdot) - c + w(\cdot), & \text{w/ prob. } A(\cdot), \text{ if } k \in \mathcal{N}. \end{cases} \tag{4.1}$$

Up to this point, we have described the model in a general way, without giving details on the form of the benefit of participation ($w(\cdot)$) and privacy cost ($g(\cdot)$ and $h(\cdot)$). In the remainder of this section, we introduce relevant models of these functions, which illustrate a variety of parameterized forms motivated by different potential settings.

Participation Benefit

The participation benefit is the non-negative value received by agents who take advantage of the service provided by the platform. Intuitively, the more participants the platform attracts, the more valuable the platform's service becomes. Take Facebook as an example, the more friends of a user use Facebook, the more valuable Facebook as a social network is to this user. However, in light of the limited budget in practice, it is almost impossible to have every agent join the platform. As a result, we define the average participation rate of the population $\bar{\theta}$ to be the ratio of number of agents on the platform to the population of agents. The average participation rate $\bar{\theta}$ can be viewed as a measure of the popularity of the platform.[1] Moreover, let $w(\bar{\theta})$ denote the participation benefit as a continuous function of the average participation rate. The participation benefit is assumed to be non-decreasing in the average participation rate of the population $\bar{\theta}$.

[1]The participation rate is determined endogenously via the equilibrium, which we discuss in Section 4.3.

Correlation Strength

Whether an agent decides to join the platform or not, an important source for her privacy cost comes from the information leakage due to data correlation. A stronger correlation naturally leads to more leakage and induces a larger privacy cost. Intuitively, if an agent's data is highly correlated with the rest of the agents, then she has a relatively large correlation strength. Moreover, some agents might share a stronger correlation with each other within a group of agents than with others outside the group. For instance, on a healthcare related platform, users who carry a common disease might share a similar pattern and thus their data is possibly highly correlated with each other. In order to capture the inter-group versus intra-group difference, we divide s agents into I groups, and agents within the same group i share a common correlation strength $\boldsymbol{\alpha}_i$. The correlation strength vector $\boldsymbol{\alpha}_i$ is further defined as $\boldsymbol{\alpha}_i \triangleq (\alpha_i, \alpha_{-i})$, where α_i and α_{-i} are used to, respectively, denote the correlation strength induced by agents inside group i and those outside group i. Note that vectors are bold while scalars are not.

Privacy cost

Privacy cost is critical to an agent's decision about whether to join the platform. For an agent who does not join the platform, her privacy cost $g(\cdot)$ comes entirely from information leakage through her peers' data sharing on the platform due to data correlation. In contrast, for an agent who joins the platform but does not report her data, not only her peers' actions but also her own interactions with the platform result in her privacy cost, which is denoted by $h(\cdot)$. Next, we introduce the parameterized form of privacy costs in detail.

Intra-group & Inter-group Privacy Loss. To differentiate the privacy cost due to correlation inside and outside the group, we further decompose privacy cost into a sum of intra-group cost and inter-group cost. Let the participation rate $\boldsymbol{\theta}_i$ denote a vector of participation rate within the group $\bar{\theta}_i$ and that of the rest groups $\bar{\theta}_{-i}$, i.e., $\boldsymbol{\theta}_i \triangleq [\bar{\theta}_i, \bar{\theta}_{-i}]$. For convenience, we later use $\bar{\theta}$ to denote the average participation rate of the overall population. For an agent in group i, her privacy cost is equal to a sum of intra-group cost and inter-group cost: $g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) = g(c, \bar{\theta}_i; \alpha_i) + g(c, \bar{\theta}_{-i}; \alpha_{-i})$ if the agent does not join the platform; otherwise her privacy cost of joining the platform is $h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) = h(c, \bar{\theta}_i; \alpha_i) + h(c, \bar{\theta}_{-i}; \alpha_{-i})$.

Up to this point, we have introduced a parameterized form of privacy functions $g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ and $h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$. To derive more clear engineering insights, we make two assumptions regarding these privacy functions. These assumptions are intuitive and consistent with the applications we consider.

Assumption 4.2.1. *[Monotonicity & Boundedness] Both $g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ and $h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ are continuously non-decreasing in cost c, participation rate $\bar{\theta}_i$ and $\bar{\theta}_{-i}$, and correlation strength α_i and α_{-i}. Further, the difference $h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ is continuously increasing in cost c, and $g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ is no greater than $h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$. Both of them are bounded by c: $g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) \leq h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) \leq c$.*

To motivate the monotonicity assumption, first recall that cost is composed of reporting costs and privacy costs, as mentioned in Section 4.2.1. Thus, an agent with a high cost tends to value her privacy more, which is captured by the monotonicity with respect to costs. Second, intuitively, privacy costs increase as more agents join the platform. Indeed, the more agents join the platform and share their data, the more information that the analyst can infer about agents, including those who do not join the platform. Third, the correlation strength characterizes how each agent's data is correlated with other agents' in the population. For an agent, a stronger correlation strength indicates that data sharing by other agents of the population could potentially cause more leakage, thus more privacy losses.

To understand the assumption $g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) \leq h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) \leq c$, we note that every agent suffers a privacy cost induced by her peers' activities on the platform due to data correlation even if she does not join the platform. In addition, for those who indeed join the platform, they tend to leak more information about themselves through their own activities on the platform. As a result, it is reasonable to assume that the privacy cost $g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ of an agent, if she does not join the platform, is smaller than her privacy cost $h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ if she joins but does not report her data. In other words, sharing on the platform potentially leads to more privacy cost. The difference of these two parts models the privacy cost due to individual sharing. Naturally, for any agent, the privacy cost due to individual sharing tends to be higher if the agent has a larger overall cost. Since the overall cost consists of reporting cost and privacy cost, any privacy cost function is modeled as a fraction of the overall cost c, i.e., privacy cost is upper bounded by c.

Assumption 4.2.2. *The privacy cost $h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ is linear in cost c with parameter $b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$, i.e.,*

$$h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) = c \cdot b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i). \tag{4.2}$$

We assume that the privacy cost function of participation $h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ is linear in cost c. This assumption is a consequence of the work in Ghosh and Roth [119], which shows that cost functions of the form $c_i\varepsilon$ are appropriate for settings using differential privacy, where ε quantifies the amount of privacy leaked. Note that the parameter $b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ inherits the monotonicity with respect to the participation rate and correlation strength from that of the privacy cost function $h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$.

Agent's Utility

After introducing the key components of agent's utility, we summarize the utility function as follows:

(i) If the agent does not join the platform, she only experiences a privacy cost $g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ induced by information leakage due to data correlation.

(ii) If the agent joins the platform but is not selected to report her data, her utility is $h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + w(\bar{\theta})$, where $w(\bar{\theta})$ is the participation benefit.

(iii) If the agent joins the platform and is selected to report her data, she incurs her overall cost c, which includes her privacy cost. Based on the agent's reported cost $\tilde{c}$, she receives a payment $P(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ and thus her utility is $P(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - c + w(\bar{\theta})$.

As the agent's privacy cost function is parameterized by the correlation strength parameter, so should the payment function $P(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$, selection probability $A(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$, utility $u(\tilde{c}, c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$. However, for simplicity, we later use $P_i(\tilde{c})$, $A_i(\tilde{c})$, and $u_i(\tilde{c}|c)$, respectively, to denote payment, selection probability, and utility function. For agent k, her utility is as follows:

$$u_i(\tilde{c}|c) = \begin{cases} -g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i), & \text{if } k \notin \mathcal{N}, \\ -h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + w(\bar{\theta}), & \text{w/ prob. } 1 - A_i(\tilde{c}), \text{ if } k \in \mathcal{N}, \\ P_i(\tilde{c}) - c + w(\bar{\theta}), & \text{w/ prob. } A_i(\tilde{c}), \text{ if } k \in \mathcal{N}. \end{cases} \tag{4.3}$$

Let the expected utility for an agent (reporting $\tilde{c}$ while having a cost c) in group i who joins the platform be denoted by $\bar{u}_i(\tilde{c}|c)$. Consequently,

$$\begin{aligned} \bar{u}_i(\tilde{c}|c) =& A_i(\tilde{c}) \left[P_i(\tilde{c}) - c + w(\bar{\theta})\right] + (1 - A_i(\tilde{c})) \left[-h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + w(\bar{\theta})\right]. \\ =& A_i(\tilde{c}) \left[P_i(\tilde{c}) - c + h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)\right] - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + w(\bar{\theta}). \end{aligned} \tag{4.4}$$

4.3 Mechanism Design

The analyst aims to perform an estimation task by designing the menu for the mechanism, which is composed of payment function and allocation rule for agents. To study this problem, we first introduce two standard desirable properties of the mechanism: *truthfulness* (Definition 4.3.1) and *budget feasibility* (Definition 4.3.2), and then define the overall objective (a trade-off between bias and variance of the estimator) (Definition 4.3.5) subject to these two properties. Following that setup, we introduce the equilibrium participation rate (Definition 4.3.7), which is a situation where no agent wants to alter her participation decision given the current participation rate. We then derive structural results on the payment function and allocation rule that induce participants' truthful reporting in the equilibrium in Section 4.3.3 (Theorem 4.3.1). Finally, we present monotonicity properties of the mechanism in Section 4.3.4.

4.3.1 Problem Statement

To define the analyst's mechanism design problem, we first describe two classical, desirable properties: truthfulness and an expected budget constraint. First, we require the mechanism to be truthful, which is common, e.g., [101, 102].

Definition 4.3.1 (Truthfulness)**.** *A mechanism is truthful if for every participant with cost c, she can maximize her expected utility if she truthfully reports her cost, i.e.,*

$$u_i(c|c) \geq u_i(\tilde{c}|c), \quad \forall \tilde{c} \neq c, \forall i. \tag{4.5}$$

Definition 4.3.1 guarantees that rational agents on the platform will truthfully report their costs. Substituting the expected utility as in Equation (4.4), we get

$$A_i(c) \cdot [P_i(c) - c + h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)] \geq A_i(\tilde{c}) \cdot [P_i(\tilde{c}) - c + h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)], \quad \forall \tilde{c} \neq c, \forall i. \tag{4.6}$$

Second, we constrain the mechanism to use a limited budget B, which limits the payments of the analyst to the agents for their data. Before presenting the budget constraint in details, we introduce some notations. Every agent owns a data point x and cost c. These pairs (x, c), owned by agents in group i ($i \in [I]$), follow a joint distribution $\mathcal{D}_i$. Define $\mathcal{D} \triangleq \{\mathcal{D}_i\}_{i\in[I]}$. Let f_i denote the marginal distribution of the cost. We assume that $f \triangleq \{f_i\}_{i\in[I]}$ is known to the analyst while $\mathcal{D} \triangleq \{\mathcal{D}_i\}_{i\in[I]}$

is unknown.[2]

Definition 4.3.2 (Expected Budget Constraint). *A mechanism with payment function P and selection probability A satisfies the expected budget constraint B if and only if*

$$\sum_{k:i=i(k)} \mathbb{E}_{c\sim f_i}[P_i(c)A_i(c)\cdot \mathbb{1}(k\in \mathcal{N}_i)] \le B. \tag{4.7}$$

We enforce the budget constraint on the expected payment in Definition 4.3.2, which is a common approach in the prior work, e.g., [100–102]. Note that the expectation is taken with respect to the marginal distribution of agent's cost f, and the agents who join the platform as only they potentially receive a payment by the allocation rule. For convenience, we use $i(k)$ to denote the group index of agent k.

We now formally define bias and variance of the estimator of the analyst. Recall that the analyst seeks to optimize a tradeoff between bias and variance in our setting. Specifically, the analyst wishes to learn an underlying parameter μ of the whole population, and he obtains an estimator $\hat{\mu}$ based on participants' data. The estimator $\hat{\mu}$ is viewed as a random variable, and the randomness of the estimator comes from the joint distribution $\mathcal{D}$ and the allocation rule A. We view the estimator as drawn from a distribution $\mathcal{T}(\mathcal{D}, A)$.

Definition 4.3.3 (Bias). *Given an allocation rule A and an instance of the true distribution $\mathcal{D}$, the bias of an estimator $\hat{\mu}$ is defined as follows:*

$$\mathbb{B}(\hat{\mu};\mathcal{D},A) = \left|\mathbb{E}_{\hat{\mu}\sim\mathcal{T}(\mathcal{D},A)}\left[\hat{\mu}-\mu\right]\right|. \tag{4.8}$$

Definition 4.3.4 (Variance). *Given an allocation rule A and an instance of the true distribution $\mathcal{D}$, the variance of an estimator $\hat{\mu}$ is defined as follows:*

$$\mathbb{V}(\hat{\mu};\mathcal{D},A) = \mathbb{E}_{\hat{\mu}\sim\mathcal{T}(\mathcal{D},A)}\left[(\hat{\mu}-\mathbb{E}[\hat{\mu}])^2\right]. \tag{4.9}$$

Since the analyst does not know the joint distribution $\mathcal{D}$, he cannot directly optimize over bias and variance of the estimator. Instead, we consider the goal of minimizing the worst-case linear combination of bias (Definition 4.3.3) and variance (Definition 4.3.4) over all instantiations of $\mathcal{D}$ that are consistent with the marginal cost distribution f.

[2] This prior $\{f_i\}$ could be constructed from previous interactions between agents and data buyers. Knowledge of the cost (or valuation) distribution is a standard assumption in Bayesian mechanism design (e.g., Myerson [124]), and is assumed in the works of Roth and Schoenebeck [100] and Chen, Immorlica, Lucier, Syrgkanis, and Ziani [101].

Definition 4.3.5 (Worst-Case Bias-Variance Trade-off)**.** *Given an allocation rule A, an instance of the true distribution $\mathcal{D}$, and a combination parameter γ, the worst-case bias-variance trade-off is the supremum of the linear combination of bias and variance:*

$$\sup_{f \text{ consistent with } \mathcal{D}} \quad \gamma \cdot \mathbb{V}(\hat{\mu}; \mathcal{D}, A) + (1-\gamma)\mathbb{B}(\hat{\mu}; \mathcal{D}, A). \tag{4.10}$$

Using the above, we formally define the mechanism design problem as follows.

Definition 4.3.6 (Mechanism Design Task)**.** *Given an estimator $\hat{\mu}$, cost distribution f, correlation strength parameters, and fixed parameter γ, the analyst aims to minimize a worst-case bias-variance trade-off by designing payment rule P and allocation rule A subject to truthfulness and budgetary constraints:*

$$\begin{aligned}
\inf_{A,P} \sup_{f \text{ consistent with } \mathcal{D}} \quad & \gamma \cdot \mathbb{V}(\hat{\mu}; \mathcal{D}, A) + (1-\gamma)\mathbb{B}(\hat{\mu}; \mathcal{D}, A) \\
s.t. \quad & \sum_{k: i=i(k)} \mathbb{E}_{c\sim f_i}[P_i(c)A_i(c) \cdot \mathbb{1}(k \in \mathcal{N}_i)] \leq B \\
& A_i(c)\,[P_i(c) - c + h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)] \geq \\
& A_i(\tilde{c})\,[P_i(\tilde{c}) - c + h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)], \quad \forall \tilde{c} \neq c, \forall i.
\end{aligned} \tag{4.11}$$

Note that each agent suffers a privacy cost due to data correlation even if she does not join the platform. This negative utility offers the analyst freedom to compensate the agent a partial cost rather than the whole cost while motivating the agent to join the platform (individual rationality). As a result, we do not have to constrain the mechanism to satisfy a positive value of participation, which is a common requirement in the prior work in this area. This is a novel, significant consequence of considering privacy leakage.

4.3.2 Equilibrium Characterization

To characterize the equilibrium that results under a given mechanism design, we consider the agents' decisions given a fixed participation rate. Recall that the participation rate is an aggregation of agents' decisions on whether to participate, and that an agent's utility depends on the participation rate. We emphasize that an agent's decision does not depend on other agents' individual decisions, rather only on the participation rate as an aggregate. This is natural for large platforms where a single agent's decision has little impact on others'.

Before defining the equilibrium concept, we first introduce some notation. Recall that there are I groups of agents parameterized by data correlation strength. We use q_i to represent the likelihood of a random agent coming from group i in the population. For agents in group i, the cost follows a continuous distribution f_i with a support set $C \triangleq [c_{\min}, c_{\max}]$. We use $\bar{\theta}_i$ to denote the participation rate for group i. Let $\boldsymbol{\theta} \triangleq [\bar{\theta}_i]_{1 \le i \le I}$, an array of average participation rate for all the groups, denote the participation rate profile. Naturally we have the average participation rate in the population satisfying $\bar{\theta} = \sum_{1 \le i \le I} q_i \cdot \bar{\theta}_i$.

For a given participation rate profile, an agent makes decisions by weighing her utility under different choices. For an agent with cost c in group i, we use a binary variable $d_i(c|\boldsymbol{\theta})$ to denote her decision. If the agent joins the platform, $d_i(c|\boldsymbol{\theta}) = 1$; otherwise $d_i(c|\boldsymbol{\theta}) = 0$. We consider a non-atomic model where a single agent's decision has no effect on the aggregate participation rate when the agent set is large enough [125]. This is a realistic assumption for a large platform. Every agent is assumed to be rational, i.e., she decides to join the platform only if her utility of non-participation is not lower than that of participation. This translates to the following mathematical expression:

$$d_i(c|\boldsymbol{\theta}) = \begin{cases} 1, & \text{if } \max_{\tilde{c}} \bar{u}_i(\tilde{c}|c) \ge -g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i), \\ 0, & \text{otherwise.} \end{cases} \tag{4.12}$$

Since both payment and selection probability depend on the reported cost, an agent tends to report the cost that maximizes her expected utility of participation $\bar{u}_i(\tilde{c}|c)$.

For an agent in group $i, 1 \le i \le I$, her decision on whether to participate is closely related to the participation rate $\bar{\theta}_i$ of group i and that of the rest groups $\bar{\theta}_{-i}$. Meanwhile, we notice that there are many possible participation rate profiles $\boldsymbol{\theta}$ that correspond to one specific average participation rate $\bar{\theta}(= \sum_{1 \le i \le I} q_i \cdot \bar{\theta}_i)$. Thus, the average participation rate $\bar{\theta}$ alone is not enough to capture the agents' decisions. This motivates us to leverage the participation rate profile $\boldsymbol{\theta}$, instead of average participation rate $\bar{\theta}$ of all the groups, as the equilibrium concept.

We now define the equilibrium concept formally. This notion of equilibrium guarantees that agents' decisions are consistent with the participation rate profile. Note that we consider a strictly positive participation rate at equilibrium, i.e., $\bar{\theta}_i^* > 0, \forall i$, with an adequate budget; otherwise a zero participation rate for certain groups essentially leads to a biased estimator towards the rest of the groups with positive participation rates.

Definition 4.3.7 (Equilibrium)**.** *A participation rate profile $\boldsymbol{\theta}^* = [\bar{\theta}_i^*]_{1 \le i \le I}$ is an equilibrium, if for each group $i, 1 \le i \le I$, the fraction of participating agents is* *exactly $\bar{\theta}_i^*$, i.e.,*

$$\int_{c_{\min}}^{c_{\max}} \mathbb{1}\left[d_i(c|\boldsymbol{\theta}^*) = 1\right] \cdot f_i(c)dc = \bar{\theta}_i^*, \quad \forall i. \tag{4.13}$$

4.3.3 Payment Function

We now analyze the structure of the payment function in the mechanism associated with the equilibrium $\boldsymbol{\theta}^*$. The analyst's mechanism design problem in Definition 4.3.6 involves a truthfulness constraint and a budgetary constraint. We first present the payment function that satisfies the truthfulness constraint for a given non-increasing allocation rule. Later, in Section 4.4, we further optimize over allocation rule to solve for the general optimization problem.

We present the payment function for group i, $1 \le i \le I$, and then characterize the requirements for a truthful mechanism. We begin with some definitions. For a fixed desired participation rate profile $\boldsymbol{\theta}$, we introduce $\hat{c}_i$ of group i, which satisfies the equality in (4.14). Later in this section, we explain in Corollary 4.3.1 that $\hat{c}_i$ can be viewed as a cost threshold for agents in group i, and agents whose cost is no greater than this value would like to participate.

$$\bar{\theta}_i = \int_{c_{\min}}^{\hat{c}_i} f_i(c)dc, 1 \le i \le I. \tag{4.14}$$

Next, we give the payment function of group i as follows

$$P_i(\tilde{c}) = \tilde{c} - h(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + \frac{1}{A_i(\tilde{c})}\left((1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \int_{\tilde{c}}^{c_{\max}} A_i(z)dz + \tau(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i)\right), \tag{4.15}$$

where

$$\tau(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i) = h(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - (1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \int_{\hat{c}_i}^{c_{\max}} A_i(z)dz - w(\bar{\theta}). \tag{4.16}$$

The characterization of the payment function relies on allocation rule A and the associated participation rate profile $\boldsymbol{\theta}$. According to Myerson's Lemma [124], the allocation rule should be non-increasing in the reported cost $\tilde{c}$ in order to induce agents' truthfulness. Further, we present the requirements for a truthful mechanism in Theorem 4.3.1. Such a mechanism can uniquely induce agents' truthful reporting of costs. If the selection probability is strictly decreasing, the mechanism can induce agents' strict truthfulness.[3] We highlight this result in Theorem 4.3.1 and put its proof in Appendix 4.A.1.

Theorem 4.3.1. *The mechanism is truthful (strictly truthful, respectively) and induces participation profile $\boldsymbol{\theta}^*$ at equilibrium if and only if for every group $i \in [I]$, both of the following statements hold:*

(i) allocation rule $A_i(\tilde{c})$ is a non-increasing (decreasing, respectively) function of the reported cost $\tilde{c}$;

(ii) payment function is given as in Equation (4.15) *(as a function of A), where $\boldsymbol{\theta} \triangleq \boldsymbol{\theta}^*$ is the desired participation rate profile.*

For a mechanism as described in Theorem 4.3.1, the agent would like to truthfully report her overall cost, as truthful reporting can maximize her expected utility. The monotonicity of the allocation rule indicates that an agent with a high reported cost in the same group is less likely to be selected and get a payment. Furthermore, we would like to emphasize that the strictly decreasing property of the allocation rule induces strict truthfulness of the mechanism. Suppose that the allocation rule is fixed in a certain range of reported cost (i.e., not decreasing). By reporting any cost in this range, the agents in this range can get the highest utility, which does not satisfy strict truthfulness definition. Thus, a strictly decreasing allocation rule is necessary to induce strict truthfulness.

Corollary 4.3.1. *Under the mechanism in Theorem 4.3.1, the participation decisions of agents in group i satisfy $d_i(c|\boldsymbol{\theta}^*) = 1$ if $c \leq \hat{c}_i$ and $d_i(c|\boldsymbol{\theta}^*) = 0$ otherwise.*

This result highlights that, under the mechanism in Theorem 4.3.1, the decisions of the agents in each group demonstrate a threshold structure with respect to overall cost c. The agents in group i will participate and truthfully report her overall cost, if her overall cost is no greater than the threshold $\hat{c}_i$; otherwise she will not participate.

4.3.4 Properties of the mechanism

We now study the connections among three key components of the mechanism design: data correlation, group participation rate, and payment (see Properties

[3] By strict truthfulness, we are saying that the agent can maximize her utility of participation if and only if she truthfully reports her cost, i.e., $u_i(c|c) > u_i(\tilde{c}|c)$, for all $\tilde{c} \neq c$. That is, by removing the equality of (4.5) in Definition 4.3.1, we get the definition of strict truthfulness.

4.3.1-4.3.3 later in this section). Existing discussions (e.g., Acemoglu, Makhdoumi, Malekian, and Ozdaglar [99]) of the connections among data correlation, number of users, and payment, which demonstrated some interesting properties of data trading, motivate our study. For example, Acemoglu, Makhdoumi, Malekian, and Ozdaglar [99] shows that the total payment to users is non-monotone in the number of users in some scenarios. It is interesting to explore such connections under the settings of this work. Moreover, studying the connections helps uncover the impacts these issues have on the mechanism design under our settings when we incorporate information leakage.

Many of the properties below highlight the complexity of data marketplaces with information leakage. For example, our first property highlights that, even though a larger average participation rate means more participants, the total payment for the entire group does not necessarily increase. This is because the individual payment for these agents will decrease as well.

Property 4.3.1. *Given a fixed selection probability A, the total expected payment for agents in group i is non-monotonic in the average participation rate of group i.*

However, despite the non-monotonicity of the expected payment in the average participation rate, there are some intuitive monotonicity properties that do hold for the payment after imposing one additional natural assumption.

Property 4.3.2. *Suppose the difference $h(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ is non-increasing in correlation strength. The payment to the agents in group i who join the platform is decreasing in both intra-group correlation strength α_i and inter-group correlation strength α_{-i}.*

In the property above, we further require the difference function, i.e., $h(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$, to be non-increasing in both intra-group correlation strength and inter-group correlation strength. Intuitively, this difference can be viewed as the extra privacy cost for the agent with cost equal to $\hat{c}_i$ due to her own activities on the platform, and thus it might be even independent of data correlation strength. As a consequence, the payment for the agents who join the platform in group i is decreasing in both intra-group and inter-group correlation strength, i.e., α_i and α_{-i}.

Some intuition for the above property comes from the observation that, when there is strong correlation among the agents' data, an individual will suffer a large privacy leakage even if they do not join the platform. As a result, they are incentivized to join the platform even when the individual payment is relatively low. Naturally, given a fixed participation rate profile and allocation rule, the total expected payment for the entire group decreases as a result.

However, in contrast to the above intuition, we do not have monotonicity of the participation rate in this setting, as we show in the property below. Note that for most of this paper, we treat the participation rate as something fixed (and optimized) by the analyst. However, understanding its behavior is important for performing such an optimization.

Property 4.3.3. *In group i, the group participation rate is non-monotonic in both the intra-group correlation strength α_i and inter-group correlation strength α_{-i}.*

To develop some intuition for the above, we note that the analyst's mechanism design is subject to a budget constraint. Although stronger data correlation indicates lower payment, the total expected payment is also affected by the selection probability. If the selection probability increases, the analyst needs to induce a lower group participation rate so that the total expected payment does not exceed the budget. Otherwise, the analyst could induce a higher participation rate.

Our last property investigates what happens as the budget increases. In this case the analyst can exploit the additional budget to improve either bias or variance. On the one hand, increasing the selection probability helps reduce the uncertainty of collected data, thus decreases the variance. On the other hand, inducing an equilibrium with a higher participation rate helps cover a wider range of participants' data, thus decreases the bias.

Property 4.3.4. *As budget increases, the estimator achieves a better bias-variance trade-off, i.e., the optimal objective of the overall optimization problem as in Definition 4.3.6 reduces.*

4.4 Optimization of the Worst-Case Bias-Variance Trade-off

We now discuss the design of a mechanism that optimizes the worst-case bias-variance trade-off. First, we introduce the analyst's choice of estimator in Section 4.4.1. Second, we characterize the worst-case bias-variance trade-off of the estimator, given a selection rule and a participation rate profile at equilibrium, in Section 4.4.2. Finally, we provide our full mechanism in Section 4.4.3.

From a mathematical perspective, our bias-variance optimization problem optimizes over two types of variables: (i) the allocation rule, which is a collection of probabilities that participants are selected to report their data and get the payments, and (ii) the participation rate profile in equilibrium, which controls the fraction of agents in each group that join the platform in the first place. However, in practice, a platform is unlikely to be willing to sacrifice a high participation rate to improve the performance of any single statistical task. Thus, optimizing over the participation rate is not attractive. Thus, here, we treat the participation rate as given so that our work can apply to any platform at any growth stage, e.g., either a start-up platform with a low participation rate or a popular platform with a high participation rate.

In what follows, we first show that the optimal allocation rule under a fixed equilibrium participation rate profile is monotone in the agents' virtual costs, as defined in Definition 4.4.2 in Section 4.4.3, and provide a closed-form solution for this allocation rule. Given that optimizing over the participation rate is unrealistic, we instead present structural results and conditions for a full participation rate profile. This provides a connection to previous work on data acquisition since, when there is a full participation rate profile, the analyst sees a representative sample of the population and obtains an unbiased estimator.

4.4.1 Estimator

The objective of the analyst's mechanism design problem in Definition 4.3.6 is to minimize the worst-case bias-variance trade-off of a given estimator. In this section, we describe the estimator of our choice. We then characterize its worst-case bias-trade-off variance in Section 4.4.2.

The analyst is interested in estimating a population statistic using the mechanism. In this paper, we focus on estimating a mean over the population in question. For example, the analyst wishes to understand the average income, average BMI, or average ratings for a new movie.

We use the Horvitz-Thompson estimator [126], which is the unique unbiased linear estimator for the settings where an analyst must sample at different rates from different sub-populations; in our case, agents in different groups or with different costs may be selected with different probabilities $A(\cdot)$, and in turn define several distinct sub-populations. In such a setting, the standard sample mean estimator is biased towards the sub-populations from which the analyst samples most (relatively to the sub-population size). The Horvitz-Thompson estimator eliminates this bias by reweighing each data point by the inverse of the selection probability associated with its sub-population. In our setting, the Horvitz-Thompson estimator reweighs the data of each agent k by $1/A^k$, where A^k is the selection probability associated with agent k.

We define the Horvitz-Thompson estimator formally below in Definition 4.4.1. Recall that $\mathcal{N}$ denotes the set of participants (i.e., agents who join the platform) and is of size N. Let $\mathcal{O} \subset \mathcal{N}$ denote the set of participants who are selected to report their data. Let x_k be the data of agent k. Suppose that the agents' data is independently and identically distributed. The Horvitz-Thompson estimator is then an unbiased estimator for the mean of data of the participants in set $\mathcal{N}$, i.e., the expectation of the estimator is equal to the expected value of the participants' data.

Definition 4.4.1 (Horvitz-Thompson estimator)**.** *The Horvitz-Thompson estimator is given by*

$$\hat{\mu} = \frac{1}{N} \sum_{k \in \mathcal{N}} \frac{x_k \cdot \mathbb{1}\{k \in \mathcal{O}\}}{A^k}, \tag{4.17}$$

where A^k is the selection probability for agent k.

Although the estimator is unbiased with respect to the set of participants $\mathcal{N}$, it may be biased with respect to the overall agent set $\mathcal{S}$, when the average participation rate is less than one. When some agents do not participate, their data are not fed to the estimator and the estimator fails to capture the information of the non-participants. On the other hand, if the average participation rate is equal to one, i.e., all the agents are willing to participate, the estimator is naturally unbiased. In conclusion, the participation rate at equilibrium is an important factor in the issue of estimation bias. Since there is an intrinsic trade-off between bias and variance, the analyst might prefer low variance instead of low or zero bias in some cases, e.g., when the linear weight associated with the variance in Equation (4.11) is high and the budget is large enough to use a large selection probability. Later, in Section 4.4.3, we present conditions under which the analyst prefers to enforce a full participation rate to achieve an unbiased estimator.

4.4.2 Characterizing the worst-case bias-variance trade-off

We characterize the worst-case bias-variance trade-off of the estimator in this section. We start by introducing some key notation. For simplicity, we assume that the agent's data is binary from the set of $\{0, 1\}$. In Remark 4.4.1, we highlight that our analysis can be generalized to continuous data in $[0, 1]$. The probability of an agent being in group i is denoted by q_i. Let $\mathbf{q} \triangleq [q_i]_{1 \le i \le I}$. Let $p_i(c) \triangleq Pr[x = 1|c, i]$ denote the probability of an agent k with cost c in group i having data point $x_k = 1$. Recall that according to Corollary 4.3.1, an agent in group i will join in the platform if her cost c is no greater than the participation threshold $\hat{c}_i$. Let $A \triangleq \{A_i(c)\}_{1\le i\le I}$, consisting of the selection probabilities in all groups. Furthermore, we assume a positive correlation between data and cost, which is mathematically described as follows.

Assumption 4.4.1. *For each group i, $p_i(c)$ is a non-decreasing function of cost c.*

The assumption is often reasonable in practice. If the agent's data is of a higher value, then her cost is more likely to be higher as well, and vice versa. For example, agents with a higher income or higher BMI might care more about their privacy, hence their reporting costs tend to be higher for them as well.

Lemma 4.4.1. *Under Assumption 4.4.1, fix allocation rule A and participation rate profile $\boldsymbol{\theta}^*$. Conditional on $\mathcal{N}$, the supremum of linear combination of bias and variance of the estimator $\hat{\mu}$ in Definition 4.4.1 is*

$$
\begin{aligned}
T(A,\boldsymbol{\theta}^*) &= \sup_{f \text{ consistent with } \mathcal{D}} \gamma\cdot\mathbb{V}(\hat{\mu};\mathcal{D},A) + (1-\gamma)\mathbb{B}(\hat{\mu};\mathcal{D},A) \\
&= \sup_{p_i(c)\in[0,1],c\le\hat{c}_i} \frac{\gamma}{s\left(\bar{\theta}^*\right)^2}\left(\sum_i q_i\int_{c_{\min}}^{\hat{c}_i}\frac{p_i(c)}{A_i(c)}f_i(c)dc - \frac{1}{\bar{\theta}^*}\left(\sum_i q_i\int_{c_{\min}}^{\hat{c}_i}p_i(c)f_i(c)dc\right)^2\right) \\
&\quad + (1-\gamma)(1-\bar{\theta}^*)\cdot\left(1-\frac{1}{\bar{\theta}^*}\sum_i q_i\int_{c_{\min}}^{\hat{c}_i}p_i(c)f_i(c)dc\right).
\end{aligned}
\tag{4.18}
$$

Remark 4.4.1. *While we focus on the binary data case, we note that if the agents' data points x_k are taken from the interval $[0,1]$, our proof can immediately be adapted to show that*

$$
\begin{aligned}
T(A,\boldsymbol{\theta}^*) &= \sup_{f \text{ consistent with } \mathcal{D}} \gamma\cdot\mathbb{V}(\hat{\mu};\mathcal{D},A) + (1-\gamma)\mathbb{B}(\hat{\mu};\mathcal{D},A) \\
&\le \sup_{p_i(c)\in[0,1],c\le\hat{c}_i} \frac{\gamma}{s\left(\bar{\theta}^*\right)^2}\left(\sum_i q_i\int_{c_{\min}}^{\hat{c}_i}\frac{p_i(c)}{A_i(c)}f_i(c)dc - \frac{1}{\bar{\theta}^*}\left(\sum_i q_i\int_{c_{\min}}^{\hat{c}_i}p_i(c)f_i(c)dc\right)^2\right) \\
&\quad + (1-\gamma)(1-\bar{\theta}^*)\cdot\left(1-\frac{1}{\bar{\theta}^*}\sum_i q_i\int_{c_{\min}}^{\hat{c}_i}p_i(c)f_i(c)dc\right).
\end{aligned}
\tag{4.19}
$$

That is, the expression that we optimize over is an over-estimate of the worst-case bias-variance trade-off; in turn, our approach still provides an upper bound and does not underestimate the bias-variance trade-off when the data is non-binary.

The proof of Lemma 4.4.1 and Remark 4.4.1 are in Appendix 4.A.2. The basic idea is to derive the variance and the worst-case bias separately, and they both are fully characterized by the distribution of participants' data (i.e., $p_i(c)$ for $c \le \hat{c}_i$). Thus, optimizing a linear combination of bias and variance gives the worst-case linear combination of bias and variance.

Lemma 4.4.1 presents an analytical formulation of the objective function with an auxiliary variable $p_i(c)$. This facilitates our later analysis of the optimization problem, which is the minimization of (4.18) subject to truthfulness constraint and budget constraint.

To conclude the subsection, we provide some interpretation about how the allocation rule A and participation rate profile $\boldsymbol{\theta}^*$ affect the trade-off between bias and variance. If the value of allocation rule $A_i(c)$ is higher, the variance $\mathbb{V}(\hat{\mu}; \mathcal{D}, A)$ is lower. However, due to the budget constraint, a higher value of allocation rule indicates a lower participation rate, which means a possibly higher bias. Thus, due to the budget constraint, the analyst needs to carefully trade off bias and variance.

The analyst's objective is to minimize the above worst-case bias-variance trade-off by choosing the allocation rule A and the participation rate profile $\boldsymbol{\theta}^*$ in the equilibrium. In next section, we discuss the optimal allocation rule and participation rate profile in details.

4.4.3 Optimal Allocation Rule and Conditions of Full Participation Rate

We now derive solutions to the worst-case bias-variance trade-off optimization problem under the budget and truthfulness constraints. First, we present the optimal allocation rule given the participation rate profile at equilibrium. Second, we identify sufficient conditions under which the optimal participation rate is one and the analyst obtains an unbiased estimator.

Optimal Allocation Rule

We first study the design of the allocation rule, given a desired participation rate profile $\boldsymbol{\theta}^*$ at equilibrium. Recall that we focus on $\boldsymbol{\theta}^*$ in which $\bar{\theta}_i^* \ge \bar{\theta}_{\min}, \forall i$ for some positive value $\bar{\theta}_{\min} > 0$ to avoid the trivial case of complete non-participation of a group.

In this section, we first introduce the notion of virtual cost in Definition 4.4.2, which is analogous to the classical notion of virtual value in [124]. The virtual costs help characterize the payment function that, given a fixed allocation rule, induces agents to report their privacy costs truthfully. The characterization of payment function is in Theorem 4.3.1. We note that the virtual cost of an agent in group i depends on her privacy cost ($h(c, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)$) and the cost distribution (F_i and f_i) in group i.

Definition 4.4.2 (Virtual Cost)**.** *In group i, given participation rate profile $\boldsymbol{\theta}^*$, the virtual cost of an agent with cost c is*

$$\phi_i(c; \boldsymbol{\theta}^*) = c - h(c, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) + (1 - b(\boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)) \frac{F_i(c)}{f_i(c)}. \tag{4.20}$$

Here, F_i and f_i are cdf and pdf of cost in group i, respectively.

Recall that the vector $\boldsymbol{\theta}_i^*$ consists of $\bar{\theta}_i^*$ and $\bar{\theta}_{-i}^*$. The virtual cost $\phi_i(c; \boldsymbol{\theta}^*)$ in group i is decreasing in the within-group participation rate $\bar{\theta}_i^*$ and the outside-group rate $\bar{\theta}_{-i}^*$, as $h(\cdot)$ and $b(\cdot)$ are increasing in both $\bar{\theta}_i^*$ and $\bar{\theta}_{-i}^*$. Recall that a higher participation rate means more privacy cost. Furthermore, we assume that the virtual cost within a group is non-decreasing in cost c as in [101, 124] (Assumption 4.4.2). Note that a uniform distribution is one example satisfying the assumption.

Assumption 4.4.2. *[Regularity] The virtual cost $\phi_i(c; \boldsymbol{\theta}^*)$ in group i is non-decreasing. Furthermore, $f_i(c)$ is twice differentiable, in which case $F_i(c) f_i'(c) \leq 2(f_i(c))^2$.*

Now we are ready to present the optimal allocation rule of each group.

Theorem 4.4.1. *Under Assumptions 4.2.1-4.4.2, given a desired participation rate profile $\boldsymbol{\theta}^*$, the optimal allocation rule of group i is*

$$A_i(c) = \begin{cases} \chi, & \text{if } \phi_i(c; \boldsymbol{\theta}^*) \leq \hat{\phi}, \\ \frac{\eta}{\sqrt{\phi_i(c; \boldsymbol{\theta}^*)}}, & \text{if } \hat{\phi} < \phi_i(c; \boldsymbol{\theta}^*) \leq \phi_i(\hat{c}_i; \boldsymbol{\theta}^*). \end{cases} \tag{4.21}$$

The characterizations of the constants η, χ, and $\hat{\phi}$ depend on the system parameters including the budget B, number of agents s, distribution of virtual cost. See (48) *in Appendix 4.A.3.*

Theorem 4.4.1 shows that the optimal allocation rule of a group can have two possible structures: *Fixed then Decreasing* (FtD) or *Strictly Decreasing* (SD), depending on system parameters such as budget B. Figure 4.2 provides an illustration. The FtD structure (the blue and red curves in Figure 4.2) has an allocation rule for the

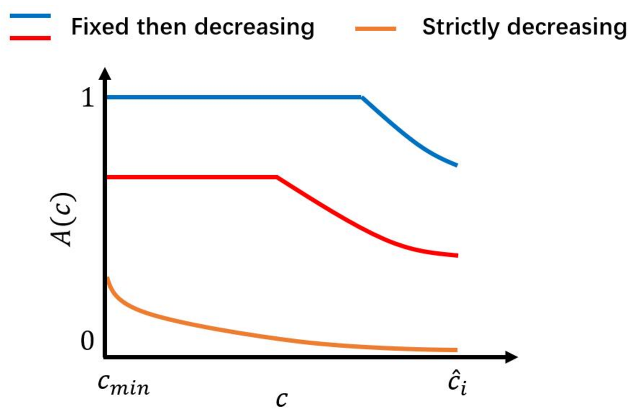


Figure 4.2: An illustration of the optimal allocation rule with two possible structures.

group that is firstly fixed in low-cost region (i.e., the region in which $\phi_i(c;\boldsymbol{\theta}^*) \leq \hat{\phi}$ holds) and then strictly decreasing in the cost in the high-cost region (inversely proportional to the square root of the virtual cost). Since the allocation rule is not strictly decreasing, the mechanism can only induce agents' weak (non-strict) truthfulness, according to Theorem 4.3.1. The other structure, SD (the orange curve in Figure 4.2), has an allocation rule that is strictly decreasing (inversely proportional to the square root of the virtual cost) in the whole region. The strictly monotonic structure induces strict truthfulness. Note that the form of the optimal allocation rule parallels that of previous work [101], showing that a similar form is still optimal in a more general setting. As we have discussed, the generalization beyond [101] to include the analyst's bias-variance trade-off, the agents' privacy costs, and data correlation add technical complexity and practicality. A priori, it is not clear that the optimal form would remain similar in the more general setting.

Note that the optimal allocation rule presented in Theorem 4.4.1 is defined for the participating agents, whose costs are lower than threshold $\hat{c}_i$ for group i. As in the case of non-participants, we only need to ensure monotonicity to induce truthfulness. A simple example is $A_i(c) = \epsilon$ for $c \geq \hat{c}_i$, where ϵ is a small enough non-negative value. Note that this example does not make the total expected payment exceed the budget, as the non-participants do not get the payments. In summary, the allocation rule of the cost lower than the threshold presented in Theorem 4.4.1 and the allocation rule of the cost higher than the threshold make up a complete menu shown to the agents, which induces agents' truthfulness and optimizes the bias-variance trade-off.

We find that, when the budget B is relatively small, the optimal allocation rule has a Strictly Decreasing structure. To state the result, first define

$$l(\boldsymbol{\theta}^*) \triangleq \sum_i q_i \bar{\theta}_i^* (h(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - w(\bar{\theta}^*)). \tag{4.22}$$

Corollary 4.4.1. *Under Assumptions 4.2.1-4.4.2, the optimal allocation rule is*

$$A_i(c) = \frac{1}{\sqrt{\phi_i(c; \boldsymbol{\theta}^*)}} \cdot \frac{B/s - l(\boldsymbol{\theta}^*)}{\sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \sqrt{\phi_i(c; \boldsymbol{\theta}^*)} f_i(c) dc}, \tag{4.23}$$

when the participation rate profile $\boldsymbol{\theta}^*$ satisfies the following inequality:

$$(B/s - l(\boldsymbol{\theta}^*))(2\gamma + \bar{\theta}^*(1-\bar{\theta}^*)(1-\gamma)s) < \gamma \sqrt{\phi_{\min}(\boldsymbol{\theta}^*)} \cdot \sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \sqrt{\phi_i(c; \boldsymbol{\theta}^*)} f_i(c) dc. \tag{4.24}$$

Furthermore, the mechanism is strictly truthful.

Corollary 4.4.1 is a special case of Theorem 4.4.1. The inequality in (4.24) indicates that the budget is relatively low. Suppose the budget is large enough. The analyst can keep the allocation rule in the low-cost region fixed, instead of strictly decreasing. In this case, the agents' reporting is weak truthful. However, if the budget is low as in (4.24), the optimal allocation rule is strictly decreasing, which induces agents' strict truthfulness. Strict truthfulness is a more desirable property than weak truthfulness.

Sufficient Conditions for Unbiased Estimation

Next, we present simple, easy to verify, sufficient conditions for unbiased estimation. We focus on the low-budget regime, which induces agents' strict truthfulness, according to Corollary 4.4.1. By substituting the allocation rule in Corollary 4.4.1 to the objective function (4.18), we arrive at an optimization problem over $\boldsymbol{\theta}^*$ as follows:

$$\max_{\boldsymbol{\theta}^*} \quad T^*(\boldsymbol{\theta}^*) = \frac{\gamma}{s} \left(U(\boldsymbol{\theta}^*) - \frac{1}{\bar{\theta}^*} \right), \tag{4.25}$$

where

$$U(\boldsymbol{\theta}^*) \triangleq \frac{1}{(\bar{\theta}^*)^2} \cdot \frac{\left(\sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \sqrt{\phi_i(c; \boldsymbol{\theta}^*)} f_i(c) dc \right)^2}{B/s - l(\boldsymbol{\theta}^*)}. \tag{4.26}$$

The objective function in (4.25) is complicated due to the complexity of $l(\boldsymbol{\theta}^*)$ in (4.26). Furthermore, the complicated characterization of its derivative with respect to group i's participation rate $\bar{\theta}_i^*$ adds to the complexity. Thus, we focus on identifying sufficient conditions under which the optimal participation rate is one, meaning that the analyst would like to have unbiased estimator.

To begin, we introduce some notation and assumptions. We denote the full participation rate profile by $\boldsymbol{\theta}_f \triangleq [1]^I \in \mathbb{R}^I$, each component of which is one. We define $\Delta_i \triangleq h(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$, which captures the additional privacy cost of an participant (compared with non-participation) whose cost is exactly the cost threshold in group i.

We now present sufficient conditions for achieving full participation and an unbiased estimator.

Proposition 4.4.1. *Suppose that, for all $\boldsymbol{\theta}^*$ in which $\bar{\theta}_i^* \geq \bar{\theta}_{\min} > 0, \forall i$, (i) the inequality in (4.24) holds, and (ii) $w'(\bar{\theta}^*) \geq D_i(\boldsymbol{\theta}^*, B, \gamma, s, \mathbf{q})$ holds for the function $D_i(\boldsymbol{\theta}^*, B, \gamma, s, \mathbf{q})$ defined in* (50) *of Appendix 4.A.5, where w' is the derivative of function $w(\cdot)$. Then, the optimal participation rate profile is $\boldsymbol{\theta}_f$ and the analyst obtains an unbiased estimator.*

Proposition 4.4.2. *Suppose that, for all $\boldsymbol{\theta}^*$ in which $\bar{\theta}_i^* \geq \bar{\theta}_{\min} > 0, \forall i$, (i) the inequality in (4.24) holds, and (ii) $\frac{\partial \Delta_i}{\partial \bar{\theta}_i^*} \leq \delta_i(\boldsymbol{\theta}^*, B, \gamma, s, \mathbf{q})$ holds for the function $\delta_i(\boldsymbol{\theta}^*, B, \gamma, s, \mathbf{q})$ defined in* (51) *of Appendix 4.A.6. Then, the optimal participation rate profile is $\boldsymbol{\theta}_f$ and the analyst obtains an unbiased estimator.*

The proofs of Proposition 4.4.1 and Proposition 4.4.2 are in Appendix 4.A.5 and Appendix 4.A.6, respectively. To understand the conditions in these two propositions, note that Condition (i) in both propositions indicates that the budget is "small", i.e., lower than a certain threshold, which is related to the expectation of virtual cost in the population. It ensures that the allocation rule is strictly decreasing, according to Corollary 4.4.1. Further, it holds if the budget itself is relatively low, or if the expectation of the virtual cost is relatively high. Recall that in the definition of virtual cost (Definition 4.4.2), the virtual cost is determined by both the cost and data correlation strength. In summary, Condition (i) indicates that the overall cost is relatively high, or the data correlation strength is relatively weak, or the budget is relatively low.

Next we discuss Condition (ii) in the above two propositions. Recall that the participation benefit $w(\bar{\theta})$ is increasing in the participation rate. Condition (ii) in Proposition 4.4.1 implies that the increment of this benefit is high as average participation rate increases, which indicates that the participation benefit is large. Meanwhile, recall that $\Delta_i = h(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ captures the addition part of the privacy cost that does not concern data correlation. Condition (ii) in Proposition 4.4.2 implies that the increment of this part of the privacy cost is small as the group participation rate increases, which indicates that the privacy cost of participation is closed to that of non-participation. Intuitively, if Conditions (ii) in the above two propositions holds, the analyst does not need to pay a lot to each agent to incentivize her participation. This is because the benefit of participation is significant (Condition (ii) in Proposition 4.4.1) or the negative impact of participation is small (Condition (ii) in Proposition 4.4.2). As a result, the expected payment of each agent is relatively low. This enables the analyst to increase the participation rate and incentivize more fraction of the population. This finally leads to a full participation rate to induce zero bias.

4.5 Concluding Remarks

In this paper, we study the design of an optimal mechanism for data acquisition in a setting where there is correlation among the data of participants that leads to information leakage. Information leakage is a crucial and under-explored feature of data markets and our results represent the first to characterize an optimal mechanism for data acquisition in such a setting. Additionally, our results provide the analyst the ability to optimally trade off bias and variance of the resulting estimator.

Further, our results provide important perspectives about the consequences of correlation and information leakage for data marketplaces. In particular, intuitively one might expect that these factors, which underlie the privacy paradox, would lead to inefficiencies where data marketplaces exploit participants and obtain data for lower prices than otherwise. The analysis in this paper shows that this is indeed the case. Combined with the results of [99], there is a compelling argument that information leakage is a significant factor in market inefficiencies that lead to oversharing and reduced payments in data marketplaces.

Thus, one may ask if it is possible to regulate data marketplaces to avoid such inefficiencies. Unfortunately, our results highlight that typical suggestions, such as avoiding monopolistic platforms, may not be effective. Since our results hold for any fixed participation rate, a platform can exploit information leakage regardless of its market size. Further, differential privacy cannot eliminate the underlying issues in this case, which comes from inter-group and intra-group correlation. Thus, an important open problem motivated by our work is: what approaches can be brought to mitigate the impact of information leakage and the privacy paradox in data marketplaces?

4.A Appendix

4.A.1 Proof of Theorem 4.3.1

First, we show that our mechanism is truthful if and only if the payment rule is of the form

$$P_i(\tilde{c}) = \tilde{c} - h(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + \frac{1}{A_i(\tilde{c})}\left((1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \int_{\tilde{c}}^{c_{\max}} A_i(z)dz + \tau(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i)\right),$$

given by Theorem 4.3.1, assuming that $\boldsymbol{\theta}$ is the true induced equilibrium participation profile. Second, we show that indeed, the equilibrium participation profile $\boldsymbol{\theta}^*$ is the desired profile $\boldsymbol{\theta}$ when using this payment rule if and only if

$$\tau(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i) = h(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}; \boldsymbol{\alpha}_i) - (1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \int_{\hat{c}_i}^{c_{\max}} A_i(z)dz - w(\bar{\theta}) \quad (4.16)$$

where $\hat{c}_i$ satisfies

$$\bar{\theta}_i = \int_{c_{\min}}^{\hat{c}_i} f_i(c)dc, 1 \le i \le I. \quad (4.14)$$

1. We first argue that our mechanism is (strictly) truthful if and only if the payment rule has the form given in Equation (4.15) and A_i is (strictly) monotone for all i.

   First, we show the "if" direction. Given a desired participation rate profile $\boldsymbol{\theta}$, we plug the payment function into the agent's utility from participation, and obtain the following expected utility $\bar{u}_i(\tilde{c}|c)$ when the agent with true cost c reports $\tilde{c}$:

   $$\begin{aligned}\bar{u}_i(\tilde{c}|c) &= A_i(\tilde{c})\left[P_i(\tilde{c}) - c + h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)\right] - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + w(\bar{\theta}) \\ &= A_i(\tilde{c})\left(\tilde{c} - h(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)\right) \\ &+ (1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \int_{\tilde{c}}^{c_{\max}} A_i(z)dz + A_i(\tilde{c})\left[-c + h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)\right] + K\end{aligned}$$

   where K does not depend on $\tilde{c}$.

   Now, note that we are in the non-atomic setting in which each agent is infinitesimal. In turn, a single agent's cost reporting and participation decisions does not affect the participation ratio, and $\frac{\partial \bar{\theta}_i}{\partial \tilde{c}} = 0$. As such, the derivative of $\bar{u}_i(\tilde{c}|c)$ with respect to $\tilde{c}$ is given by

   $$\begin{aligned}&A_i'(\tilde{c})(\tilde{c} - h(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) + A_i(\tilde{c})\left(1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)\right) \\ &- (1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i))\, A_i(\tilde{c}) - A_i'(\tilde{c})(c - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \\ &= A_i'(\tilde{c})(\tilde{c} - h(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - (c - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i))).\end{aligned}$$

In turn, when the allocation rule is decreasing with respect to cost, i.e., $A_i'(c) < 0$, we obtain that

$$\frac{\partial \bar{u}_i(\tilde{c}|c)}{\partial \tilde{c}} = A_i'(\tilde{c})(\tilde{c} - h(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - (c - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i))) \begin{cases} > 0, & \text{if } \tilde{c} < c, \\ = 0, & \text{if } \tilde{c} = c, \\ < 0, & \text{if } \tilde{c} > c. \end{cases}$$

Therefore, the agent maximizes her expected utility of participation if and only if she truthfully reports her cost, i.e., $\tilde{c} = c$. This corresponds to strict truthfulness. On the other hand, if the allocation rule is non-increasing, i.e., $A_i'(c) \leq 0$, we have

$$\frac{\partial \bar{u}_i(\tilde{c}|c)}{\partial \tilde{c}} = A_i'(\tilde{c})(\tilde{c} - h(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - (c - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i))) \begin{cases} \geq 0, & \text{if } \tilde{c} \leq c, \\ \leq 0, & \text{if } \tilde{c} > c. \end{cases}$$

Therefore, the agent maximizes her expected utility of participation if she truthfully reports her cost, i.e., $\tilde{c} = c$. This corresponds to general truthfulness, which incorporates strict truthfulness as a special case.

We now show that "only if" direction: any (strictly) truthful payment function must have the form given in Equation (4.15) and requires A_i to be (strictly) monotone. By truthfulness, we have $\bar{u}_i(c|c) = \max_{\tilde{c}} \bar{u}_i(\tilde{c}|c)$. Applying the envelope theorem to $\max_{\tilde{c}} \bar{u}_i(\tilde{c}|c)$ yields

$$\frac{\partial \bar{u}_i(c|c)}{\partial c} = \left.\frac{\partial \bar{u}_i(\tilde{c}|c)}{\partial c}\right|_{\tilde{c}=c} = -(1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \cdot A_i(c) - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i).$$

Taking the integral from c to $c_{\max}$, we further have

$$\begin{aligned} &\bar{u}_i(c_{\max}|c_{\max}) - \bar{u}_i(c|c) = \\ &(b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - 1) \int_c^{\max} A_i(z)dz + h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - h(c_{\max}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i). \end{aligned}$$

Since $\bar{u}_i(c|c) = A_i(c)\,[P_i(c) - c + h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)] - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + w(\bar{\theta})$, we obtain the following equation:

$$\begin{aligned} &\bar{u}_i(c_{\max}|c_{\max}) + (1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \int_c^{c_{\max}} A_i(z)dz + h(c_{\max}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) \\ &= A_i(c)\,[P_i(c) - c + h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)] - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + w(\bar{\theta}). \end{aligned}$$

Hence, we have

$$P_i(c) = c - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + \frac{1}{A_i(c)} \left((1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \int_c^{c_{\max}} A_i(z)dz + v \right),$$

where

$$v = \bar{u}_i(c_{\max}|c_{\max}) + h(c_{\max}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - w(\bar{\theta}).$$

Now, remembering that

$$\frac{\partial \bar{u}_i(\tilde{c}|c)}{\partial \tilde{c}} = A_i'(\tilde{c})(\tilde{c} - h(\tilde{c}, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - (c - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i))),$$

(strict) truthfulness implies that for all c, there exists $\epsilon > 0$ (small) such that for any $\tilde{c} \in (c - \epsilon, c)$, $\frac{\partial \bar{u}_i(\tilde{c}|c)}{\partial \tilde{c}} \geq 0$ (> 0 for strict) and for any $\tilde{c} \in (c, c + \epsilon)$, $\frac{\partial \bar{u}_i(\tilde{c}|c)}{\partial \tilde{c}} \leq 0$ (< 0 for strict). Since $c - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) = c(1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i))$ is increasing, this in particular requires that $A_i'(\tilde{c}) \leq 0$ (< 0 for strict truthfulness) on $(c - \epsilon, c)$ and $(c, c + \epsilon)$. Since this holds for all c, this in particular implies that for all $\tilde{c}$, $A_i'(\tilde{c}) \leq 0$ (resp < 0), which shows (strict) monotonicity of A.

2. It remains to show that our payment rule induces an equilibrium participation profile $\boldsymbol{\theta}^*$ equal to the desired participation profile $\boldsymbol{\theta}$ if and only if

$$\tau(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i) = h(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}; \boldsymbol{\alpha}_i) - (1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \int_{\hat{c}_i}^{c_{\max}} A_i(z) dz - w(\bar{\theta}), \tag{27}$$

where $\hat{c}_i$ is defined as the solution to

$$\bar{\theta}_i = \int_{c_{\min}}^{\hat{c}_i} f_i(c) dc, 1 \leq i \leq I, \tag{28}$$

and $\hat{c}_i$ is such that if all (and only the) agents in group i with cost at most $\hat{c}_i$ participate, then the participation rate in group i is $\bar{\theta}_i$. As such, to show the result, we simply need to show that the participating agents in group i are those with cost at most $\hat{c}_i$ if and only if $v = \tau(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i)$. To do so, note that an agent in group i with cost c who truthfully reports his cost has utility

$$\begin{aligned}\bar{u}_i(c|c) &= A_i(c) \left[P_i(c) - c + h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)\right] - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + w(\bar{\theta}) \\ &= (1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \int_{c}^{c_{\max}} A_i(z) dz + v - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + w(\bar{\theta}),\end{aligned}$$

and has a utility of $-g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ for non-participation. In turn, an agent in group i and with cost c participates if and only

$$(1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \int_{c}^{c_{\max}} A_i(z) dz + g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + w(\bar{\theta}) + v \geq 0.$$

Note that because w is continuous, $g(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - h(c, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)$ is continuous and increasing in c, and $A_i(z) \geq 0$ hence $\int_{c}^{c_{\max}} A_i(z) dz$ is continuous non-increasing in c, we have that $\bar{u}_i(c|c)$ is continuous and decreasing in c. In turn an agent participates exactly when his cost satisfies $c \leq \hat{c}_i$ if and only if

$$(1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i)) \int_{\hat{c}_i}^{c_{\max}} A_i(z)dz + g(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - h(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + w(\bar{\theta}) + v = 0,$$

which yields

$$v = \tau(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i).$$

This concludes the proof.

4.A.2 Proof of Lemma 4.4.1 and Remark 4.4.1

We first derive the variance. Recall that $\mathcal{O} \subset \mathcal{N}$ denotes the set of reporters selected in the participants. For simplicity of notations, we make the conditioning on $\mathcal{N}$ implicit in this proof. Let us fix the participation rate profile $\boldsymbol{\theta}^*$, the associated cost threshold $\hat{c}_i$ under which agents are willing to participate, and the corresponding participation rate $\bar{\theta}^*$ under this profile in the population. Note that there are $N = s\bar{\theta}^*$ participants. The estimator is an average of $N = s\bar{\theta}^*$ i.i.d. random variables, where each variable has variance

$$\sigma^2 = \mathbb{E}\left[\left(\frac{x_k \cdot \mathbb{1}\{k \in \mathcal{O}\}}{A^k}\right)^2\right] - \mathbb{E}\left[\frac{x_k \cdot \mathbb{1}\{k \in \mathcal{O}\}}{A^k}\right]^2.$$

Note that

$$\begin{aligned}
\mathbb{E}\left[\left(\frac{x_k \cdot \mathbb{1}\{k \in \mathcal{O}\}}{A^k}\right)^2\right] &= \mathbb{E}\left[\left(\frac{x_k}{A^k}\right)^2 \cdot \mathbb{1}\{k \in \mathcal{O}\}\right] \\
&= \mathbb{E}_{i,c}\left[\mathbb{E}_{x_k,\mathcal{O}}\left[\left(\frac{x_k}{A^k}\right)^2 \cdot \mathbb{1}\{k \in \mathcal{O}\}\middle| i, c\right]\right] \\
&= \mathbb{E}_{i,c}\left[\frac{\mathbb{E}\left[x_k^2|i,c\right]}{A_i(c)^2} \cdot \Pr\left[k \in \mathcal{O}|i,c\right]\right] \\
&= \mathbb{E}_{i,c}\left[\frac{\mathbb{E}\left[x_k|i,c\right]}{A_i(c)}\right],
\end{aligned}$$

where the second-to-last step uses the independence of x and $\mathcal{O}$ conditional on i, c, and the last step uses the facts i) conditional on k being a participant, $\Pr\left[k \in \mathcal{O}|i,c\right] = A_i(c)$ and ii) $x_k^2 = x_k$ as $x_k \in \{0,1\}$. When $x_k \in [0,1]$ instead, note that $\mathbb{E}\left[x_k^2|i,c\right] \leq \mathbb{E}\left[x_k|i,c\right]$, and we get an inequality instead, which proves Remark 4.4.1. A similar calculation yields (both in the binary and non-binary cases)

$$\mathbb{E}\left[\frac{x_k \cdot \mathbb{1}\{k \in \mathcal{O}\}}{A^k}\right] = \mathbb{E}_{i,c}\left[\mathbb{E}\left[x_k|i,c\right]\right].$$

Further, using the fact that an agent in group i participates if and only if $c \le \hat{c}_i$, note that the probability density of an agent being in group i and having cost c *conditional on the agent participating* is given by

$$\frac{q_i f_i(c)}{\sum\limits_{1\le i\le I} q_i \int_{c_{\min}}^{\hat{c}_i} f_i(c)dc} \;\text{ if }\; c_{\min} \le c \le \hat{c}_i,$$

and is equal to 0 if $c > \hat{c}_i$. In turn, we obtain that

$$\sigma^2 = \frac{\sum\limits_{1\le i\le I} q_i \int_{c_{\min}}^{\hat{c}_i} \frac{\mathbb{E}[x^2|i,c]}{A_i(c)} f_i(c)dc}{\sum\limits_{1\le i\le I} q_i \int_{c_{\min}}^{\hat{c}_i} f_i(c)dc} - \left(\frac{\sum\limits_{1\le i\le I} q_i \int_{c_{\min}}^{\hat{c}_i} \mathbb{E}[x|i,c] f_i(c)dc}{\sum\limits_{1\le i\le I} q_i \int_{c_{\min}}^{\hat{c}_i} f_i(c)dc} \right)^2 .$$

Since $N = \frac{1}{s\bar{\theta}^*}$ and that $\bar{\theta}^* = \sum\limits_{1\le i\le I} q_i \int_{c_{\min}}^{\hat{c}_i} f_i(c)dc$, the variance of the Horvitz-Thompson estimator is then given by

$$\frac{1}{s\bar{\theta}^*} \left(\frac{\sum\limits_{1\le i\le I} q_i \int_{c_{\min}}^{\hat{c}_i} \frac{\mathbb{E}[x^2|c,i]}{A_i(c)} f_i(c)dc}{\bar{\theta}^*} - \left(\frac{\sum\limits_{1\le i\le I} q_i \int_{c_{\min}}^{\hat{c}_i} \mathbb{E}[x|c,i] f_i(c)dc}{\bar{\theta}^*} \right)^2 \right).$$

Next, we characterize the worst-case bias (over the data of the non-participants). Recall that we assume positive correlation between data and costs. In turn, the worst-case bias corresponds to the case where the data of all the non-participants (whose costs and data are higher than those of participants in each group) is one. To see this, let μ be true population mean. Let $\mathbb{E}[x|d(c) = 0]$, resp. $\mathbb{E}[x|d(c) = 1]$, be the expectations of the data of the non-participants (decision variable $d(c) = 0$), resp. participants (decision variable $d(c) = 1$). Recall that the estimator $\hat{\mu}$ is an unbiased estimator of participants' data, i.e., $\mathbb{E}[\hat{\mu}] = \mathbb{E}[x|d(c) = 1]$. We have

$$s\bar{\theta}^*\mathbb{E}[\hat{\mu}] + s(1 - \bar{\theta}^*)\mathbb{E}[x|d(c) = 0] = s\mu,$$

which leads to

$$\mu - \mathbb{E}[\hat{\mu}] = (1 - \bar{\theta}^*)(\mathbb{E}[x|d(c) = 0] - \mathbb{E}[\hat{\mu}]).$$

Positive correlation between data and cost indicates that as the agents' costs increase, their data is more likely to increase as well. Meanwhile, we show in Corollary 4.3.1 that non-participants have higher costs. Thus, we know that the expectation of non-participants' data is no less than that of participants' data, i.e.,

$$\mathbb{E}[x|d(c)=0] \geq \mathbb{E}[x|d(c)=1] = \mathbb{E}[\hat{\mu}].$$

Therefore, when taking absolute values, we have

$$|\mu - \mathbb{E}[\hat{\mu}]| = (1-\bar{\theta}^*)(\mathbb{E}[x|d(c)=0] - \mathbb{E}[\hat{\mu}]).$$

The bias is maximized when the expectation of non-participants' data is the maximum value, i.e., $\mathbb{E}[x|d(c)=0]=1$. Thus, we have

$$|\mu - \mathbb{E}[\hat{\mu}]| = (1-\bar{\theta}^*)(1 - \mathbb{E}[\hat{\mu}]).$$

Plugging in (by a similar calculation as for the variance terms),

$$\mathbb{E}[\hat{\mu}] = \frac{1}{\bar{\theta}^*}\sum_{1\leq i\leq I} q_i \int_{c_{\min}}^{\hat{c}_i} p_i(c) f_i(c)dc,$$

we immediately obtain that the worst-case bias (where the worst-case is taken over the data of the non-participants) is

$$|\mu - \mathbb{E}[\hat{\mu}]| = (1-\bar{\theta}^*)\left(1 - \frac{1}{\bar{\theta}^*}\sum_{1\leq i\leq I} q_i \int_{c_{\min}}^{\hat{c}_i} p_i(c) f_i(c)dc\right).$$

So far we have derived the variance and the worst-case bias separately. Notice that both variance and worst-case bias are characterized by the distribution of participants data, i.e., $p_i(c)$, for $c \leq \hat{c}_i$ and all i. Thus, the supremum in $p_i(c)$ (i.e., over the participants' data) of the linear combination of the variance and the worst-case bias gives the worst-case linear combination of variance and bias, i.e., the worst-case bias-variance trade-off, over the data of both participants and non-participants.

4.A.3 Proof of Theorem 4.4.1

Let $\boldsymbol{\theta}^*$ be the desired equilibrium participation rate profile. The proof goes as follows. First, we write the problem of finding the optimal allocation rule as a minimax optimization problem over a discretization of the costs. Second, we interpret this optimization problem as a zero-sum game between the analyst who controls the allocation rule and aims to minimize the bias-variance trade-off, and an adversary who controls the correlation between data and costs and aims to maximize the bias-variance objective. Finally, we convert our solution in the discrete cost case to the original, continuous case.

Reformulating the optimization program. The optimization problem in Definition 4.3.6 involves two constraints, a truthfulness constraint and a budget constraint. Because our payment rule uniquely induces truthfulness and the desired participation rate profile $\boldsymbol{\theta}^*$ by Theorem 4.3.1, we can directly plug in the closed-form expression for the payment into the budget constraint, and replace the truthfulness constraints by a monotonicity constraint on the selection rule A_i, without loss of generality. That is, the expected payment to a single participant is given by

$$\begin{aligned}
&\sum_i q_i \int_{c_{\min}}^{\hat{c}_i} A_i(c) P_i(c) f_i(c) dc \\
=&\sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \left(A_i(c) \left(c - h(c, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)\right) + \left(1 - b(\boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)\right) \int_c^{c_{\max}} A_i(z) dz \right) f_i(c) dc \\
&+ \sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \tau(\boldsymbol{\theta}_i^*, \boldsymbol{\alpha}_i) f_i(c) dc \\
=&\sum_i q_i \int_{c_{\min}}^{\hat{c}_i} A_i(c) \left(c - h(c, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)\right) f_i(c) dc \\
&+ \sum_i q_i \left(1 - b(\boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)\right) \int_{c_{\min}}^{\hat{c}_i} \left(\int_c^{c_{\max}} A_i(z) dz \right) f_i(c) dc \\
&+ \sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \tau(\boldsymbol{\theta}_i^*, \boldsymbol{\alpha}_i) f_i(c) dc.
\end{aligned}$$

First, let us simplify the double-integral term. We remark that

$$\begin{aligned}
\int_{c_{\min}}^{\hat{c}_i} \left(\int_c^{c_{\max}} A_i(z) dz \right) f_i(c) dc &= \int_{c_{\min}}^{\hat{c}_i} \left(\int_{c_{min}}^{c_{\max}} A_i(z) dz \right) f_i(c) dc \\
&- \int_{c_{\min}}^{\hat{c}_i} \left(\int_{c_{min}}^{c} A_i(z) dz \right) f_i(c) dc \\
&= \int_{c_{min}}^{c_{\max}} \left(\int_{c_{\min}}^{\hat{c}_i} f_i(c) dc \right) A_i(z) dz \\
&- \int_{c_{min}}^{\hat{c}_i} \left(\int_z^{\hat{c}_i} f_i(c) dc \right) A_i(z) dz \\
&= F_i(\hat{c}_i) \int_{c_{min}}^{c_{\max}} A_i(z) dz - F_i(\hat{c}_i) \int_{c_{min}}^{\hat{c}_i} A_i(z) dz \\
&+ \int_{c_{min}}^{\hat{c}_i} F_i(z) A_i(z) dz \\
&= F_i(\hat{c}_i) \int_{\hat{c}_i}^{c_{\max}} A_i(c) dc + \int_{c_{\min}}^{\hat{c}_i} F_i(c) A_i(c) dc.
\end{aligned}$$

As $\tau(\boldsymbol{\theta}_i, \boldsymbol{\alpha}_i)$ is independent of c, we also have that

$$\begin{aligned}\int_{c_{\min}}^{\hat{c}_i} \tau(\boldsymbol{\theta}_i^*, \boldsymbol{\alpha}_i) f_i(c) dc =& \ \tau(\boldsymbol{\theta}_i^*, \boldsymbol{\alpha}_i) \int_{c_{\min}}^{\hat{c}_i} f_i(c) dc \\ =& \ F_i(\hat{c}_i) \left(h(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - w(\bar{\theta}^*)\right) \\ & - F_i(\hat{c}_i) \left(\left(1 - b(\boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)\right) \int_{\hat{c}_i}^{c_{\max}} A_i(z) dz\right).\end{aligned}$$

Thus,

$$\begin{aligned}&\sum_i q_i \int_{c_{\min}}^{\hat{c}_i} A_i(c) P_i(c) f_i(c) dc \\ =& \sum_i q_i \int_{c_{\min}}^{\hat{c}_i} A_i(c) \left(c - h(c, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)\right) f_i(c) dc \\ &+ \sum_i q_i \left(1 - b(\boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)\right) \int_{c_{\min}}^{\hat{c}_i} F_i(c) A_i(c) dc \\ &+ \sum_i q_i \left(1 - b(\boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)\right) F_i(\hat{c}_i) \int_{\hat{c}_i}^{c_{\max}} A_i(c) dc \\ &- \sum_i q_i F_i(\hat{c}_i) \left(\left(1 - b(\boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)\right) \int_{\hat{c}_i}^{c_{\max}} A_i(z) dz\right) \\ &+ \sum_i q_i F_i(\hat{c}_i) \left(h(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - w(\bar{\theta}^*)\right) \\ =& \sum_i q_i \int_{c_{\min}}^{\hat{c}_i} A_i(c) \left(c - h(c, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) + \left(1 - b(\boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)\right) \frac{F_i(c)}{f_i(c)}\right) f_i(c) dc \\ &+ \sum_i q_i F_i(\hat{c}_i) \left(h(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - w(\bar{\theta}^*)\right).\end{aligned}$$

Remembering that by definition $F_i(\hat{c}_i) = \bar{\theta}_i^*$, and that the virtual costs are given by

$$\phi_i(c; \boldsymbol{\theta}^*) = c - h(c, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) + (1 - b(\boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i)) \frac{F_i(c)}{f_i(c)},$$

we can rewrite

$$\begin{aligned}&\sum_i q_i \int_{c_{\min}}^{\hat{c}_i} A_i(c) P_i(c) f_i(c) dc \\ =& \sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \phi_i(c; \boldsymbol{\theta}^*) A_i(c) f_i(c) dc + \sum_i q_i \bar{\theta}_i^* (h(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - w(\bar{\theta}^*)).\end{aligned}$$

Therefore, the equivalent optimization program is given by:

$$\begin{aligned}
&\min_{A,\boldsymbol{\theta}^*} \; T(A,\boldsymbol{\theta}^*) \\
&\quad s.t. \\
&\qquad \sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \phi_i(c;\boldsymbol{\theta}^*) A_i(c) f_i(c) dc \\
&\qquad + \sum_i q_i \bar{\theta}_i^* (h(\hat{c}_i,\boldsymbol{\theta}_i^*;\boldsymbol{\alpha}_i) - g(\hat{c}_i,\boldsymbol{\theta}_i^*;\boldsymbol{\alpha}_i) - w(\bar{\theta})) \leq \frac{B}{s}, \\
&\qquad A_i(c) \in [0,1] \text{ is a non-increasing function } \forall i \in [I], \\
&\qquad 0 < \bar{\theta}_i^* \leq 1 \; \forall i \in [I].
\end{aligned}$$

Now, we fix the equilibrium participation profile $\boldsymbol{\theta}^*$. We focus on finding the optimal selection rule given the participation profile. This is given by the following optimization program, plugging back the expression for $T(A,\boldsymbol{\theta}^*)$:

$$\begin{aligned}
\min_{A_i(c)} \max_{p_i(c)} \quad & \frac{\gamma}{s\left(\bar{\theta}^*\right)^2} \left(\sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \frac{p_i(c)}{A_i(c)} f_i(c) dc - \frac{1}{\bar{\theta}^*} \left(\sum_i q_i \int_{c_{\min}}^{\hat{c}_i} p_i(c) f_i(c) dc \right)^2 \right) \\
& + (1-\gamma)(1-\bar{\theta}^*) \left(1 - \frac{1}{\bar{\theta}^*} \sum_i q_i \int_{c_{\min}}^{\hat{c}_i} p_i(c) f_i(c) dc \right) \\
s.t. \quad & \sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \phi_i(c;\boldsymbol{\theta}^*) A_i(c) f_i(c) dc \\
& + \sum_i q_i \bar{\theta}_i (h(\hat{c}_i,\boldsymbol{\theta}_i^*;\boldsymbol{\alpha}_i) - g(\hat{c}_i,\boldsymbol{\theta}_i^*;\boldsymbol{\alpha}_i) - w(\bar{\theta}^*)) \leq \frac{B}{s}, \\
& 0 \leq A_i(c), \leq 1 \; \forall i, c, \\
& 0 \leq p_i(c), \leq 1 \; \forall i, c.
\end{aligned}$$

Note that we do not require the monotonicity constraints on A_i and p_i. We will later show that the solutions A_i^* and p_i to this relaxed optimization problem are indeed monotone, hence relaxing the constraint is without loss of generality.

Solving for the discrete cost case. The above formulation is presented for continuous costs. Before we find the solution to the continuous cost problem, let us first focus on the case of discrete costs and find the corresponding optimization program and solution. We will later show how to transform the discrete solution into an optimal solution to the continuous optimization problem above.

We first introduce the notations in the discrete case. Suppose the cost c in the population is from a discrete and finite set $\{c_1, c_2, ..., c_J\}$ with size J. Recall that there are I groups of agents parameterized by data correlation strength. We write π_{ij} as the probability of an agent belonging to group i and having cost c_j. Naturally we have $\sum_{1\le i\le I, 1\le j\le J} \pi_{ij} = 1$. Recall that agents' decisions have threshold structure, i.e., only agents with costs below the cost threshold choose to participate. We use $t(i)$ to denote the index of threshold cost in group i. That is, if the cost $c \le c_{t(i)}$ for an agent in group i, he would like to participate. Let ϕ_{ij} be the virtual cost of an agent with cost c_j in group i as follows

$$\phi_{ij} = \begin{cases} c_1 - h(c_1, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i), & \text{if } j = 1, \\ c_j - h(c_j, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) + (1 - b(\boldsymbol{\theta}_i; \boldsymbol{\alpha}_i))(c_j - c_{j-1}) \frac{\sum_{t=1}^{j-1} \pi_{it}}{\pi_{ij}}, & \text{if } j > 1. \end{cases}$$

Since $c_1 - h(c_1, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) > 0$, $\phi_{ij} > 0$. Recall that $p_i(c) = Pr[x = 1 | c, \boldsymbol{\alpha}_i]$ in continuous case is the probability of the data being one. We use p_{ij} to denote the probability of the data being one for an agent in group i having cost c_j. We are trying to find the optimal selection probability for discrete cost in each group. Let A_{ij} be the selection probability for agents with cost c_j in group i. The discrete version of the the min-max optimization problem is as follows:

$$\begin{aligned} \min_A \max_p \ & \frac{\gamma}{s\left(\bar{\theta}^*\right)^2} \left(\sum_{1\le i\le I, 1\le j\le t(i)} \pi_{ij} \cdot \frac{p_{ij}}{A_{ij}} - \frac{1}{\bar{\theta}^*} \left(\sum_{1\le i\le I, 1\le j\le t(i)} \pi_{ij} p_{ij} \right)^2 \right) \\ & + (1-\gamma)(1-\bar{\theta}^*) \left(1 - \frac{1}{\bar{\theta}^*} \sum_{1\le i\le I, 1\le j\le t(i)} \pi_{ij} p_{ij} \right) \\ s.t. \ & \sum_{1\le i\le I, 1\le j\le t(i)} \pi_{ij} \phi_{ij} A_{ij} \\ & + \sum_i q_i \bar{\theta}_i (h(c_{t(i)}, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - g(c_{t(i)}, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - w(\bar{\theta}^*)) \le \frac{B}{s}, \\ & 0 \le A_{ij} \le 1, \ \forall i \in [I], \ j \in [t(i)], \\ & 0 \le p_{ij} \le 1, \ \forall i \in [I], \ j \in [t(i)]. \end{aligned} \tag{29}$$

We write $A = [A_{ij}]_{1\le i\le I, 1\le j\le t(i)}$ and $p = [p_{ij}]_{1\le i\le I, 1\le j\le t(i)}$. Here, there is an implicit constraint on the mechanism to make the solution meaningful:

$$\frac{B}{s} - \sum_i q_i \bar{\theta}_i (h(c_{t(i)}, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - g(c_{t(i)}, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - w(\bar{\theta}^*)) > 0.$$

If it does not hold, otherwise, it means $\sum_{i,j} \pi_{ij} \phi_{ij} A_{ij} \le 0$ and $A_{ij} = 0, \forall i, j$ is the solution. Recall that $\phi_{ij} > 0, \forall i, j$. In this case, no data is collected, which is meaningless. The constraint means the budget should be higher than a threshold related to participation rate $\boldsymbol{\theta}^*$ so as to generate positive selection probability. Meanwhile, without loss of generality, we assume

$$\sum_{1\le i\le I, 1\le j\le t(i)} \pi_{ij}\phi_{ij} + \sum_i q_i\bar{\theta}_i(h(c_{t(i)}, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - g(c_{t(i)}, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - w(\bar{\theta}^*)) > \frac{B}{s}$$

to avoid a trivial solution of $A_{ij} = 1, \forall i, j$. If the above inequality does not hold, it is optimal to select all agents with probability one. This assumption means the analyst would not aggressively select all the participants with abundant budget.

We denote the objective function inside the minimax problem as $U(A, p)$ for simplification. Notice that $U(A, p)$ is a convex in A and concave in p. To see this, let $\pi = [\pi_{ij}]_{1\le i\le I, 1\le j\le t(i)}$. We can write $U(A, p)$ as

$$U(A,p) = \frac{\gamma}{s\left(\bar{\theta}^*\right)^2}\left(\langle \pi, p./A\rangle - \frac{1}{\bar{\theta}^*}\langle \pi, p\rangle^2\right) + (1-\gamma)(1-\bar{\theta}^*)\left(1 - \frac{1}{\bar{\theta}^*}\langle \pi, p\rangle\right).$$

With abuse of notation, we use ./ to denote component-wise product (division) operation for vectors, and use $\langle , \rangle$ to denote inner product operation. Then we can obviously figure out that $U(A, p)$ is convex in A and concave in p, similarly for each problem. In turn, the optimization program defines a convex-concave zero-sum game, in which player 1 (the analyst) is choosing A to minimize $U(A, p)$ given p and player 2 (an adversary) is choosing q to maximize $U(A, p)$ given A. The solution of the minimax problem corresponds to the equilibrium (A^*, p^*) of the zero-sum game satisfying the constraint $U(A^*, p^*) = \min_A U(A, p^*) = \max_p U(A^*, p)$. Thus, we can find the equilibrium of the game to derive the optimal allocation rule.

Next, we find the equilibrium by characterizing the best responses of two players.

Lemma 4.A.1. *Given A, p_{ij} for each i, j is the best response of maximizing player if and only if one of the following holds:*

1. $p_{ij} = 1$ *and* $\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{1}{A_{ij}} - \frac{2}{\bar{\theta}^*}\left(\sum_{i,j}\pi_{ij}p_{ij}\right)\right) - \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma) > 0$;
2. $p_{ij} = 0$ *and* $\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{1}{A_{ij}} - \frac{2}{\bar{\theta}^*}\left(\sum_{i,j}\pi_{ij}p_{ij}\right)\right) - \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma) < 0$;
3. $0 \le p_{ij} \le 1$ *and* $\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{1}{A_{ij}} - \frac{2}{\bar{\theta}^*}\left(\sum_{i,j}\pi_{ij}p_{ij}\right)\right) - \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma) = 0$.

Proof. Since the objective function of the maximizer is convex and differentiable, the KKT conditions are necessary and sufficient for optimality, and the optimal solutions are such that there exist dual variables under which the KKT conditions are satisfied. Note that the Lagrangian of the maximization problem is as follows:

$$L(p, \lambda_{ij}^+, \lambda_{ij}^-, \lambda) = U(A, p) + \sum_{i,j} \lambda_{ij}^+ (1 - p_{ij}) + \sum_{i,j} \lambda_{ij}^- p_{ij}.$$

Hence, the KKT conditions yield

$$\frac{\partial L}{\partial p_{ij}} = \frac{\gamma \pi_{ij}}{s\left(\bar{\theta}^*\right)^2} \left(\frac{1}{A_{ij}} - \frac{2}{\bar{\theta}^*} \left(\sum_{i,j} \pi_{ij} p_{ij} \right) \right) - \frac{1}{\bar{\theta}^*}(1 - \bar{\theta}^*)(1 - \gamma) - \lambda_{ij}^+ + \lambda_{ij}^- = 0,$$

$$\lambda_{ij}^+(1 - p_{ij}) = 0, \ \ \lambda_{ij}^- p_{ij} = 0, \ \ \lambda_{ij}^+ \geq 0, \ \ \lambda_{ij}^- \geq 0, \ \ \lambda \geq 0.$$

- If $\frac{\gamma}{s\left(\bar{\theta}^*\right)^2} \left(\frac{1}{A_{ij}} - \frac{2}{\bar{\theta}^*} \left(\sum_{i,j} \pi_{ij} p_{ij} \right) \right) - \frac{1}{\bar{\theta}^*}(1 - \bar{\theta}^*)(1 - \gamma) > 0$, then $\lambda_{ij}^+ > 0$, hence a best response must satisfy $p_{ij} = 1$ by complementary slackness. Further, when taking $\lambda_{ij}^+ > 0$, $\lambda_{ij}^- = 0$, and $p_{ij} = 1$, the KKT conditions hold; hence, $p_{ij} = 1$ is indeed a (the unique) best response.
- If $\frac{\gamma}{s\left(\bar{\theta}^*\right)^2} \left(\frac{1}{A_{ij}} - \frac{2}{\bar{\theta}^*} \left(\sum_{i,j} \pi_{ij} p_{ij} \right) \right) - \frac{1}{\bar{\theta}^*}(1 - \bar{\theta}^*)(1 - \gamma) < 0$, it must be that $\lambda_{ij}^- > 0$, which in turns implies $p_{ij} = 0$. Further, the KKT conditions hold with $\lambda_{ij}^- > 0$, $\lambda_{ij}^+ = 0$, $p_{ij} = 0$, hence $p_{ij} = 0$ is indeed a (the unique) best response.
- Finally, if $\frac{\gamma}{s\left(\bar{\theta}^*\right)^2} \left(\frac{1}{A_{ij}} - \frac{2}{\bar{\theta}^*} \left(\sum_{i,j} \pi_{ij} p_{ij} \right) \right) - \frac{1}{\bar{\theta}^*}(1 - \bar{\theta}^*)(1 - \gamma) = 0$, the KKT conditions hold so long as $0 \leq p_{ij} \leq 1$ with $\lambda_{ij}^- = \lambda_{ij}^+ = 0$. Therefore, any $p_{ij} \in [0, 1]$ with
$$\frac{\gamma}{s\left(\bar{\theta}^*\right)^2} \left(\frac{1}{A_{ij}} - \frac{2}{\bar{\theta}^*} (\sum_{i,j} \pi_{ij} p_{ij}) \right) - \frac{1}{\bar{\theta}^*}(1 - \bar{\theta}^*)(1 - \gamma) = 0,$$
is a best response for the maximizing player.

□

Next, the best response of the minimizing player (the analyst) is given by the following lemma:

Lemma 4.A.2. *Given p,*

- *if* $p_{ij} = 0, \forall i, j$, *any* $A_{ij} \in [0, 1]$ *that satisfies budget constraint is the best response of the minimizing player;*

- *if $p_{ij} \neq 0, \forall i, j$, the best response of the minimizing player is*

$$A^*_{ij} = \min\left\{1, \sqrt{\frac{\gamma p_{ij}}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_{ij}}}\right\}, \tag{30}$$

  where λ^* is such that

$$\sum_{i,j:p_{ij}\neq 0} \pi_{ij}\phi_{ij} \cdot \min\left\{1, \sqrt{\frac{\gamma p_{ij}}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_{ij}}}\right\}$$
$$+ \sum_i q_i \bar{\theta}_i \left(h(c_{t(i)}, \boldsymbol{\theta}^*_i; \boldsymbol{\alpha}_i) - g(c_{t(i)}, \boldsymbol{\theta}^*_i; \boldsymbol{\alpha}_i) - w(\bar{\theta})\right) = B/s.$$

Proof. If $p_{ij} = 0, \forall i, j$, then A does not appear in the objective function. Thus, any $A_{ij} \in [0, 1]$ that satisfies budget constraint is the best response of the minimizing player.

Next, we focus on the case of $p_{ij} \neq 0, \forall i, j$. We drop the constraint that $A_{ij} \geq 0$ for all i, j (we will see that this is without loss of generality, as we will recover a positive solution). The Lagrangian of the minimization problem is as follows:

$$L(A, \lambda, \lambda_{ij}) = U(A, p) + \sum_{i,j} \lambda_{ij}(A_{ij} - 1) +$$
$$\lambda \cdot \left(\sum_{i,j} \pi_{ij}\phi_{ij}A_{ij} + \sum_i q_i\bar{\theta}_i(h(c_{t(i)}, \boldsymbol{\theta}^*_i; \boldsymbol{\alpha}_i) - g(c_{t(i)}, \boldsymbol{\theta}^*_i; \boldsymbol{\alpha}_i) - w(\bar{\theta}^*)) - B/s\right).$$

From the KKT condition, we have the optimal A^*_{ij}, and optimal dual λ^*. λ^*_{ij} must satisfy

$$\frac{\partial L}{\partial A_{ij}} = -\frac{\gamma}{s\left(\bar{\theta}^*\right)^2} \cdot \frac{\pi_{ij}p_{ij}}{A^{*2}_{ij}} + \lambda^*\pi_{ij}\phi_{ij} + \lambda^*_{ij} = 0, \tag{31}$$
$$\lambda^*_{ij}(A^*_{ij} - 1) = 0,$$
$$\lambda^* \cdot \left(\sum_{i,j} \pi_{ij}\phi_{ij}A_{ij} + \sum_i q_i\bar{\theta}_i(h(c_{t(i)}, \boldsymbol{\theta}^*_i; \boldsymbol{\alpha}_i) - g(c_{t(i)}, \boldsymbol{\theta}^*_i; \boldsymbol{\alpha}_i) - w(\bar{\theta}^*)) - B/s\right) = 0.$$

Thus, we have i) $A^*_{ij} = 1$; or ii) $A^*_{ij} < 1$ and $\lambda^*_{ij} = 0$. In the second case, we obtain that $A^*_{ij} = \sqrt{\frac{\gamma p_{ij}}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_{ij}}}$ for some $\lambda^* > 0$, according to (31). A higher value of selection probability (which indicates higher budget) would always reduce variance, and thus, the objective function. Thus, the optimal allocation rule is such that the budget constraint is binding. In conclusion, we have $A^*_{ij} = \min\left\{1, \sqrt{\frac{\gamma p_{ij}}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_{ij}}}\right\}$ where λ^* is such that the budget constraint is binding.

□

Before we present the intersection of both players' best responses, we change some indexes for simplicity. We denote the set of virtual cost of all groups, $\Phi \triangleq \{\phi_{ij} : 1 \le i \le I, 1 \le j \le t(i)\}$. Suppose there are $K \triangleq \sum_{1\le i\le I} t(i)$ number of elements in the set. We sort them in a non-decreasing order, and replace index ij with index k, which indicates the corresponding position in the sorted set, i.e., $\Phi = \{\phi_k : 1 \le k \le K, \phi_1 < \phi_2 < ... < \phi_K\}$. Similarly, we replace index ij of A_{ij}, p_{ij}, π_{ij} with index k, and obtain A_k, p_k, π_k, which are associated with virtual cost ϕ_k. For simplicity, we define l as follows

$$l \triangleq \sum_i q_i \bar{\theta}_i^* (h(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i^*; \boldsymbol{\alpha}_i) - w(\bar{\theta}^*)).$$

We begin with some necessary notations as follows. We define

$$Q(m,z) \triangleq \sum_{k=1}^{m} \pi_k \phi_k + \sqrt{\frac{\phi_m}{z}} \cdot \sum_{k=m+1}^{K} \pi_k \sqrt{\phi_k}, m = 1, ..., K. \tag{32}$$

$$R(m,z) \triangleq 2\gamma \left(\frac{z}{\phi_m} \cdot \sum_{k=1}^{m} \pi_k \phi_k + \sum_{k=m+1}^{K} \pi_k \right) + \left(\bar{\theta}^*\right)^2 (1-\bar{\theta}^*)(1-\gamma)s, m = 1, ..., K. \tag{33}$$

Notice that $R(m,z) > 0$.

Claim 4.A.1. *$Q(m,1)$ is increasing in m, $R(m,1)$ is decreasing in m and thus, $\frac{Q(m,1)}{R(m,1)}$ is increasing in m.*

Proof. We can see that

$$\begin{aligned}
&Q(m+1,1) - Q(m,1) \\
&= \sum_{k=1}^{m+1} \pi_k \phi_k - \sum_{k=1}^{m} \pi_k \phi_k + \sum_{k=m+2}^{K} \pi_k \sqrt{\phi_k \phi_{m+1}} - \sum_{k=m+1}^{K} \pi_k \sqrt{\phi_k \phi_m} \\
&= \pi_{m+1}\phi_{m+1} + \sum_{k=m+2}^{K} \pi_k \sqrt{\phi_k} \left(\sqrt{\phi_{m+1}} - \sqrt{\phi_m} \right) - \pi_{m+1} \sqrt{\phi_m \phi_{m+1}} \\
&= \sum_{k=m+2}^{K} \pi_k \sqrt{\phi_k} \left(\sqrt{\phi_{m+1}} - \sqrt{\phi_m} \right) + \pi_{m+1} \sqrt{\phi_{m+1}} \left(\sqrt{\phi_{m+1}} - \sqrt{\phi_m} \right) > 0.
\end{aligned}$$

The inequality is due to increasing virtual cost, i.e., ϕ_k is increasing in k. Also, we have

$$
\begin{aligned}
R(m+1,1)-R(m,1) &= 2\gamma\left(\sum_{k=1}^{m+1}\pi_k\phi_k\frac{1}{\phi_{m+1}}-\sum_{k=1}^{m}\pi_k\phi_k\frac{1}{\phi_m}+\sum_{k=m+2}^{K}\pi_k-\sum_{k=m+1}^{K}\pi_k\right)\\
&= 2\gamma\left(\sum_{k=1}^{m}\pi_k\phi_k\left(\frac{1}{\phi_{m+1}}-\frac{1}{\phi_m}\right)+\pi_{m+1}\phi_{m+1}\frac{1}{\phi_{m+1}}-\pi_{m+1}\right)\\
&= 2\gamma\sum_{k=1}^{m}\pi_k\phi_k\left(\frac{1}{\phi_{m+1}}-\frac{1}{\phi_m}\right)<0.
\end{aligned}
$$

The inequality is also due to increasing virtual cost. Thus $Q(m,1)$ is increasing in m, and $R(m,1)$ is decreasing in m. Notice that $R(m,1)>0$. As such, we have that $\frac{Q(m,1)}{R(m,1)}$ is increasing in m.

□

Claim 4.A.2. *For $m=1,\dots,K-1$,*

$$
Q(m+1,1)=Q\left(m,\frac{\phi_m}{\phi_{m+1}}\right),\quad R(m+1,1)=R\left(m,\frac{\phi_m}{\phi_{m+1}}\right). \tag{34}
$$

Proof.

$$
\begin{aligned}
Q(m+1,1) &= \sum_{k=1}^{m+1}\pi_k\phi_k+\sqrt{\phi_{m+1}}\cdot\sum_{k=m+2}^{K}\pi_k\sqrt{\phi_k}\\
&= \sum_{k=1}^{m}\pi_k\phi_k+\pi_{m+1}\phi_{m+1}+\sqrt{\phi_{m+1}}\cdot\sum_{k=m+2}^{K}\pi_k\sqrt{\phi_k}\\
&= \sum_{k=1}^{m}\pi_k\phi_k+\sqrt{\phi_{m+1}}\cdot\pi_{m+1}\sqrt{\phi_{m+1}}+\sqrt{\phi_{m+1}}\cdot\sum_{k=m+2}^{K}\pi_k\sqrt{\phi_k}\\
&= \sum_{k=1}^{m}\pi_k\phi_k+\sqrt{\phi_{m+1}}\cdot\sum_{k=m+1}^{K}\pi_k\sqrt{\phi_k}\\
&= \sum_{k=1}^{m}\pi_k\phi_k+\sqrt{\phi_m}\cdot\sqrt{\frac{\phi_{m+1}}{\phi_m}}\cdot\sum_{k=m+1}^{K}\pi_k\sqrt{\phi_k}\\
&= Q\left(m,\frac{\phi_m}{\phi_{m+1}}\right).
\end{aligned}
$$

To prove $R(m+1,1)=R\left(m,\frac{\phi_m}{\phi_{m+1}}\right)$, it suffices to show

$$
\frac{1}{\phi_{m+1}}\cdot\sum_{k=1}^{m+1}\pi_k\phi_k+\sum_{k=m+2}^{K}\pi_k=\frac{\phi_m}{\phi_{m+1}}\cdot\frac{1}{\phi_m}\cdot\sum_{k=1}^{m}\pi_k\phi_k+\sum_{k=m+1}^{K}\pi_k.
$$

To this end, we can check

$$\begin{aligned}\frac{1}{\phi_{m+1}}\cdot\sum_{k=1}^{m+1}\pi_k\phi_k+\sum_{k=m+2}^{K}\pi_k &= \frac{1}{\phi_{m+1}}\cdot\sum_{k=1}^{m}\pi_k\phi_k+\frac{1}{\phi_{m+1}}\cdot\pi_{m+1}\phi_{m+1}+\sum_{k=m+2}^{K}\pi_k\\ &= \frac{1}{\phi_{m+1}}\cdot\sum_{k=1}^{m}\pi_k\phi_k+\pi_{m+1}+\sum_{k=m+2}^{K}\pi_k\\ &= \frac{\phi_m}{\phi_{m+1}}\cdot\frac{1}{\phi_m}\cdot\sum_{k=1}^{m}\pi_k\phi_k+\sum_{k=m+1}^{K}\pi_k.\end{aligned}$$

So we have $R(m+1,1)=R\left(m,\frac{\phi_m}{\phi_{m+1}}\right)$. □

Next, we characterize the intersection of both players' best responses. The minimizing player's strategy corresponds to the solution of the optimization problem in (29). We will show that the minimizing player's strategy has the following form:

$$A_k=\begin{cases}\chi, & \text{if } k\le\hat{k},\\ \frac{1}{\sqrt{\phi_k}}\cdot\frac{B/s-l-\chi\sum_{k=1}^{\hat{k}}\pi_k\phi_k}{\sum_{k=\hat{k}+1}^{K}\pi_k\sqrt{\phi_k}}, & \text{if } k>\hat{k}.\end{cases}\tag{35}$$

Here, the constants χ and $\hat{k}$ are defined as follows:

- If $\frac{B/s-l}{\gamma\bar{\theta}^*}<\frac{Q(1,1)}{R(1,1)}$, then $\chi=0$, $\hat{k}=0$.
- If $\frac{Q(1,1)}{R(1,1)}\le\frac{B/s-l}{\gamma\bar{\theta}^*}<\frac{Q(K,1)}{R(K,1)}$, then $m^*\in\{1,\ldots,K-1\}$ and $z^*\in(0,1]$ be such that $\frac{Q(m^*,z^*)}{R(m^*,z^*)}=\frac{B/s-l}{\gamma\bar{\theta}^*}$ (we prove its existence later).
  - If $\frac{B/s-l}{Q(m^*,z^*)}\le 1$, then $\chi=\frac{B/s-l}{Q(m^*,z^*)}$ and $\hat{k}=m^*$.
  - If $\frac{B/s-l}{Q(m^*,z^*)}>1$, then $\chi=1$ and $\hat{k}=\max\{k:Q(k,1)<B/s-l\}$..
- If $\frac{B/s-l}{\gamma\bar{\theta}^*}\ge\frac{Q(K,1)}{R(K,1)}$, then $\chi=\frac{B/s-l}{\sum_{k=1}^{K}\phi_k}$ and $\hat{k}=K$.

Now we begin to present how to obtain this solution by deriving the intersection of both players' best responses.

- Case 1: $\frac{B/s-l}{\gamma\bar{\theta}^*}<\frac{Q(1,1)}{R(1,1)}$, i.e.,

$$(B/s-l)(2\gamma+\bar{\theta}^*(1-\bar{\theta}^*)(1-\gamma)s)<\gamma\sqrt{\phi_1}\cdot\sum_{k=1}^{K}\pi_k\sqrt{\phi_k}.\tag{36}$$

Notice that $\sum_{k=1}^{K} \pi_k = \bar{\theta}^*$. Then,

$$A_k = \frac{1}{\sqrt{\phi_k}} \cdot \frac{B/s - l}{\sum\limits_{1 \le k \le K} \pi_k \sqrt{\phi_k}}, \quad \forall k, \tag{37}$$

and $p_k = 1, \forall k$ constitute the mutual best response. To see this, A_k is a best response to p_k. Let

$$\lambda^* = \frac{\gamma \left(\sum\limits_{k=1}^{K} \pi_k \sqrt{\phi_k} \right)^2}{s(\bar{\theta}^*)^2 \left(B/s - l\right)^2}.$$

Then, we have

$$\begin{aligned}
&\sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}} \\
&= \sqrt{\frac{\gamma}{s(\bar{\theta}^*)^2} \times \frac{s(\bar{\theta}^*)^2 \left(B/s - l\right)^2}{\gamma \left(\sum\limits_{k=1}^{K} \pi_k \sqrt{\phi_k} \right)^2}} \times \sqrt{\frac{1}{\phi_k}} \\
&= \frac{1}{\sqrt{\phi_k}} \cdot \frac{B/s - l}{\sum\limits_{k=1}^{K} \pi_k \sqrt{\phi_k}} \\
&= A_k.
\end{aligned}$$

Meanwhile,

$$\begin{aligned}
A_k &= \frac{1}{\sqrt{\phi_k}} \cdot \frac{B/s - l}{\sum\limits_{k=1}^{K} \pi_k \sqrt{\phi_k}} \\
&\le \frac{B/s - l}{\sqrt{\phi_1} \sum\limits_{k=1}^{K} \pi_k \sqrt{\phi_k}} \\
&< \frac{\gamma}{2\gamma + \bar{\theta}^*(1 - \bar{\theta}^*)(1 - \gamma)s} \\
&< \frac{1}{2} < 1.
\end{aligned}$$

The budget constraint is binding, as

$$\begin{aligned}\sum_{k=1}^{K}\pi_k\phi_k A_k &= \sum_{k=1}^{K}\pi_k\phi_k\frac{1}{\sqrt{\phi_k}}\cdot\frac{B/s-l}{\sum\limits_{k=1}^{K}\pi_k\sqrt{\phi_k}}\\ &= \frac{B/s-l}{\sum\limits_{k=1}^{K}\pi_k\sqrt{\phi_k}}\cdot\sum_{k=1}^{K}\pi_k\sqrt{\phi_k}\\ &= B/s-l.\end{aligned}$$

Thus, we have $A_k = \min\left\{1, \sqrt{\frac{\gamma p_k}{s(\bar{\theta}^*)^2\lambda^*\phi_k}}\right\}$ for all k. So A is indeed a best response to p, as per Lemma 4.A.2. It is also easy to check that A_k is decreasing in k as virtual cost ϕ_k is increasing.

On the other hand, q is a best response to A in P2. We have for all k, $p_k = 1$, and

$$\begin{aligned}\frac{\gamma}{s}\left(\frac{1}{A_k}-\frac{2}{\bar{\theta}^*}(\sum_{k=1}^{K}\pi_k p_k)\right) &= \frac{\gamma}{s}\left(\frac{\sqrt{\phi_k}\sum\limits_{k=1}^{K}\pi_k\sqrt{\phi_k}}{B/s-l}-2\right)\\ &\geq \frac{\gamma}{s}\left(\frac{\sqrt{\phi_1}\sum\limits_{k=1}^{K}\pi_k\sqrt{\phi_k}}{B/s-l}-2\right)\\ &> \frac{\gamma}{s}\left(2+\frac{\bar{\theta}^*(1-\bar{\theta}^*)(1-\gamma)s}{\gamma}-2\right)\\ &= \bar{\theta}^*(1-\bar{\theta}^*)(1-\gamma).\end{aligned}$$

The equality in the first line follows from $\sum_{k=1}^{K} p_k = \bar{\theta}^*$. The inequality in the second line follows from $\phi_k \geq \phi_1$. The inequality in the third line follows from $\frac{B/s-l}{\gamma\bar{\theta}^*} < \frac{Q(1,1)}{R(1,1)}$. Thus, $\frac{\gamma}{s}\left(\frac{1}{A_k}-\frac{2}{\bar{\theta}^*}(\sum\limits_{k=1}^{K}\pi_k p_k)\right)-\bar{\theta}^*(1-\bar{\theta}^*)(1-\gamma) > 0$. So p is indeed a best response to A, as per Lemma 4.A.1. Meanwhile, we can see that p is indeed non-decreasing.

- Case 2: $\frac{Q(1,1)}{R(1,1)} \leq \frac{B/s-l}{\gamma\bar{\theta}^*} < \frac{Q(K,1)}{R(K,1)}$.

Claim 4.A.3. *There exists* $m^* \in \{1, ..., K-1\}$ *and* $z^* \in (\frac{\phi_{m^*}}{\phi_{m^*+1}}, 1]$ *such that*

$$\frac{Q(m^*, z^*)}{R(m^*, z^*)} = \frac{B/s-l}{\gamma\bar{\theta}^*}. \tag{38}$$

Proof. Recall that $\frac{Q(m,1)}{R(m,1)}$ is increasing in m according to Claim 4.A.1. As $\frac{Q(1,1)}{R(1,1)} \leq \frac{B/s-l}{\gamma\bar{\theta}^*} < \frac{Q(K,1)}{R(K,1)}$, there exists a unique $m^* \in \{1, ..., K-1\}$ such that

$$\frac{Q(m^*,1)}{R(m^*,1)} \leq \frac{B/s-l}{\gamma\bar{\theta}^*} < \frac{Q(m^*+1,1)}{R(m^*+1,1)}.$$

Recall that $Q(m+1,1) = Q\left(m, \frac{\phi_m}{\phi_{m+1}}\right)$ and $R(m+1,1) = R\left(m, \frac{\phi_m}{\phi_{m+1}}\right)$ in Claim 4.A.2. We have

$$\frac{Q(m^*,1)}{R(m^*,1)} \leq \frac{B/s-l}{\gamma\bar{\theta}^*} < \frac{Q(m^*+1,1)}{R(m^*+1,1)} = \frac{Q\left(m^*, \frac{\phi_{m^*}}{\phi_{m^*+1}}\right)}{R\left(m^*, \frac{\phi_{m^*}}{\phi_{m^*+1}}\right)}.$$

Notice that $Q(m,z)$ is continuously decreasing in z and $R(m,z)$ is continuously increasing in z. So we have $\frac{Q(m,z)}{R(m,z)}$ continuously decreasing in z. Thus, there is a unique $z^* \in (\frac{\phi_{m^*}}{\phi_{m^*+1}}, 1]$ such that

$$\frac{Q(m^*,z^*)}{R(m^*,z^*)} = \frac{B/s-l}{\gamma\bar{\theta}^*}.$$

□

There are two subcases depending on the value of $\frac{B/s-l}{Q(m^*,z^*)}$.

– Case 2(a): $\frac{B/s-l}{Q(m^*,z^*)} \leq 1$. Then

$$p_k = \begin{cases} \frac{z^*}{\phi_{m^*}} \cdot \phi_k, & \text{if } k \leq m^*, \\ 1, & \text{if } k > m^*, \end{cases} \tag{39}$$

and

$$A_k = \begin{cases} \chi \triangleq \frac{B/s-l}{Q(m^*,z^*)}, & \text{if } k \leq m^*, \\ \frac{1}{\sqrt{\phi_k}} \cdot \frac{B/s-l-\chi \sum_{k=1}^{m^*} \pi_k \phi_k}{\sum_{k=m^*+1}^{K} \pi_k \sqrt{\phi_k}}, & \text{if } k > m^*, \end{cases} \tag{40}$$

constitute the mutual best response. Indeed, on the one hand, A is a best response to p. Let

$$\lambda^* = \frac{\gamma\left(\sum_{k=1}^{K} \pi_k \sqrt{p_k \phi_k}\right)^2}{s(\bar{\theta}^*)^2 \left(B/s-l\right)^2}.$$

For $k \leq m^*$, we have

$$\sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}} = \sqrt{\frac{\gamma}{s(\bar{\theta}^*)^2} \times \frac{s(\bar{\theta}^*)^2 \left(B/s - l\right)^2}{\gamma \left(\sum_{k=1}^{K} \pi_k \sqrt{p_k \phi_k}\right)^2}} \times \sqrt{\frac{p_k}{\phi_k}}$$

$$= \frac{B/s - l}{\sum_{k=1}^{m^*} \pi_k \sqrt{\frac{z^*}{\phi_{m^*}} \cdot \phi_k \cdot \phi_k} + \sum_{k=m^*+1}^{K} \pi_k \sqrt{\phi_k}} \times \sqrt{\frac{z^*}{\phi_{m^*}}}$$

$$= \frac{B/s - l}{Q(m^*, z^*)} = A_k \leq 1.$$

For $k > m^*$, by a similar derivation, we have

$$\sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}}$$

$$= \frac{B/s - l}{Q(m^*, z^*)} \times \sqrt{\frac{\phi_{m^*}}{z^*}} \times \sqrt{\frac{1}{\phi_k}}$$

$$= \frac{B/s - l}{Q(m^*, z^*)} \times \frac{\sum_{k=1}^{m^*} \pi_k \phi_k + \sum_{k=m^*+1}^{K} \pi_k \sqrt{\frac{\phi_k \phi_m}{z^*}} - \sum_{k=1}^{m^*} \pi_k \phi_k}{\sum_{k=m^*+1}^{K} \pi_k \sqrt{\phi_k}} \times \sqrt{\frac{1}{\phi_k}}$$

$$= \frac{B/s - l}{Q(m^*, z^*)} \times \frac{Q(m^*, z^*) - \sum_{k=1}^{m^*} \pi_k \phi_k}{\sum_{k=m^*+1}^{K} \pi_k \sqrt{\phi_k}} \times \sqrt{\frac{1}{\phi_k}}$$

$$= \frac{B/s - l - \frac{B/s - l}{Q(m^*, z^*)} \cdot \sum_{k=1}^{m^*} \pi_k \phi_k}{\sum_{k=m^*+1}^{K} \pi_k \sqrt{\phi_k}} \times \sqrt{\frac{1}{\phi_k}}$$

$$= \frac{B/s - l - \chi \cdot \sum_{k=1}^{m^*} \pi_k \phi_k}{\sum_{k=m^*+1}^{K} \pi_k \sqrt{\phi_k}} \times \sqrt{\frac{1}{\phi_k}} = A_k.$$

As the virtual cost ϕ_k is increasing, A_k is decreasing for $k > m^*$. Notice that

$A_{m^*} > A_{m^*+1}$. This is because $z^* > \frac{\phi_{m^*}}{\phi_{m^*+1}}$, and naturally

$$\begin{aligned} A_{m^*} &= \sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}} \\ &= \sqrt{\frac{\gamma}{s\left(\bar{\theta}^*\right)^2 \lambda^*}} \times \sqrt{\frac{z^*}{\phi_{m^*}}} \\ &> \sqrt{\frac{\gamma}{s\left(\bar{\theta}^*\right)^2 \lambda^*}} \times \sqrt{\frac{1}{\phi_{m^*+1}}} \\ &= A_{m^*+1}. \end{aligned}$$

From this, we can see that A_k given by (40) is non-increasing, and hence, $A_k \leq 1$ for all k. Thus, we have $A_k = \min\left\{1, \sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}}\right\}$, for all k. The budget constraint is binding, as

$$\begin{aligned} \sum_{k=1}^{K} \pi_k \phi_k A_k &= \sum_{k=1}^{m^*} \pi_k \phi_k \chi + \sum_{k=m^*+1}^{K} \pi_k \phi_k \frac{1}{\sqrt{\phi_k}} \cdot \frac{B/s - l - \chi \sum_{k=1}^{m^*} \pi_k \phi_k}{\sum_{k=m^*+1}^{K} \pi_k \sqrt{\phi_k}} \\ &= \chi \cdot \sum_{k=1}^{m^*} \pi_k \phi_k + \frac{B/s - l - \chi \sum_{k=1}^{m^*} \pi_k \phi_k}{\sum_{k=m^*+1}^{K} \pi_k \sqrt{\phi_k}} \cdot \sum_{k=m^*+1}^{K} \pi_k \sqrt{\phi_k} \\ &= \chi \cdot \sum_{k=1}^{m^*} \pi_k \phi_k + B/s - l - \chi \cdot \sum_{k=1}^{m^*} \pi_k \phi_k = B/s - l. \end{aligned}$$

So A is indeed a best response to p, as per Lemma 4.A.2.

On the other hand, p is a best response to A. Recall that $\frac{Q(m^*,z^*)}{R(m^*,z^*)} = \frac{B/s-l}{\gamma\bar{\theta}^*}$. For

$k \leq m^*$, recall that $0 \leq p_k \leq 1$ and $A_k = \frac{B/s-l}{Q(m^*,z^*)}$. We have

$$\begin{aligned}
&\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{1}{A_k} - \frac{2}{\bar{\theta}^*}(\sum_{k=1}^{K}\pi_k p_k)\right) \\
&= \frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{Q(m^*,z^*)}{B/s-l} - \frac{2}{\bar{\theta}^*}\left(\sum_{k=1}^{m^*}\pi_k\phi_k\frac{z^*}{\phi_m} + \sum_{k=m^*+1}^{K}\pi_k\right)\right) \\
&= \frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{R(m^*,z^*)}{\gamma\bar{\theta}^*} - \frac{2}{\bar{\theta}^*}\left(\sum_{k=1}^{m^*_+}\pi_k\phi_k\frac{z^*}{\phi_m} + \sum_{k=m^*+1}^{K}\pi_k\right)\right) \\
&= \frac{1}{s(\bar{\theta}^*)^3}\left(R(m^*,z^*) - 2\gamma\left(\sum_{k=1}^{m^*}\pi_k\phi_k\frac{z^*}{\phi_m} + \sum_{k=m^*+1}^{K}\pi_k\right)\right) \\
&= \frac{1}{s(\bar{\theta}^*)^3} \times \left(\bar{\theta}^*\right)^2(1-\bar{\theta}^*)(1-\gamma)s \\
&= \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma),
\end{aligned}$$

i.e., $\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{1}{A_k} - \frac{2}{\bar{\theta}^*}\left(\sum_{k=1}^{K}\pi_k p_k\right)\right) - \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma) = 0$. With this equality, we do have a best response for $k \leq m^*$.

For $k > m^*$, we have $p_k = 1$. Remember that $A_{m^*+1} > A_{m^*}$ and that A_k is decreasing for $k > m^*$. We obtain $A_k < A_{m^*}$ for $k > m^*$. Then we have for $k > m^*_+$,

$$\begin{aligned}
&\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{1}{A_k} - \frac{2}{\bar{\theta}^*}(\sum_{k=1}^{K}\pi_k p_k)\right) - \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma) \\
> \ &\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{1}{A_{m^*}} - \frac{2}{\bar{\theta}^*}(\sum_{k=1}^{K}\pi_k p_k)\right) - \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma) \\
= \ &0.
\end{aligned}$$

So p is indeed a best response to A, as per Lemma 4.A.1. We can see that p_k is indeed non-decreasing.

- Case 2(b): $\frac{B/s-l}{Q(m^*,z^*)} > 1$.

Claim 4.A.4. *There exists* $k' \in \{1, ..., m^*\}$ *and* $z' \in (\frac{\phi_{k'}}{\phi k'+1}, 1]$*such that*

$$R(k', z') = \gamma\bar{\theta}^*. \tag{41}$$

Proof. Recall that $R(m^*, z^*) = \frac{Q(m^*,z^*)}{B/s-l} \cdot \gamma\bar{\theta}^*$ and $\frac{B/s-l}{Q(m^*,z^*)} > 1$. We obtain $R(m^*, z^*) < \gamma\bar{\theta}^*$. Recall that $R(m, z)$ is decreasing in m, and increasing in z, and $R(m^*+1,1) = R\left(m^*, \frac{\phi_{m^*}}{\phi_{m^*+1}}\right)$ (Claim 4.A.2). We have

$$R(m^*+1,1) = R\left(m^*, \frac{\phi_{m^*}}{\phi_{m^*+1}}\right) < R(m^*, z^*) < \gamma\bar{\theta}^*.$$

Here, the first inequality holds as $\frac{\phi_{m^*}}{\phi_{m^*+1}} < z^*$, according to Claim 4.A.3. Meanwhile, $R(1,1) = 2\gamma\bar{\theta}^* + \left(\bar{\theta}^*\right)^2(1-\bar{\theta}^*)(1-\gamma)s > \gamma\bar{\theta}^*$. Thus, there exists $k' \le m^*$ such that

$$R(k'+1,1) < \gamma\bar{\theta}^* \le R(k',1).$$

Recall that $R\left(k', \frac{\phi_{k'}}{\phi_{k'+1}}\right) = R(k'+1,1)$. We have

$$R\left(k', \frac{\phi_{k'}}{\phi_{k'+1}}\right) = R(k'+1,1) < \gamma\bar{\theta}^* \le R(k',1).$$

Since $R(m,z)$ is continuously increasing in z, there exists $z' \in (\frac{\phi_{k'}}{\phi_{k'+1}}, 1]$ such that

$$R(k', z') = \gamma\bar{\theta}^*.$$

□

Define

$$k^* = \max\{k : Q(k,1) < B/s - l\}. \tag{42}$$

Then

$$p_k = \begin{cases} \frac{z'}{\phi_{k'}} \cdot \phi_k, & \text{if } k \le k', \\ 1, & \text{if } k > k', \end{cases} \tag{43}$$

and

$$A_k = \begin{cases} \chi \triangleq 1, & \text{if } k \le k^*, \\ \frac{1}{\sqrt{\phi_k}} \cdot \frac{B/s - l - \sum_{k=1}^{k^*} \pi_k \phi_k}{\sum_{k=k^*+1}^{K} \pi_k \sqrt{\phi_k}}, & \text{if } k > k^*, \end{cases} \tag{44}$$

constitute the mutual best response. To see this, we know that A is the best response to p. Recall that $Q(m,z)$ is increasing in m and decreasing in z. As $k' \le m^*$ (Claim 4.A.4) and $z^* \le 1$ (Claim 4.A.3), we have

$$Q(k',1) \le Q(m^*,1) \le Q(m^*, z^*) < B/s - l,$$

i.e., $B/s - l < Q(k',1)$. By the definition of $k^* = \max\{k : Q(k,1) < B/s - l\}$, we have $k^* \ge k'$.

Meanwhile, we have $Q(k^*+1,1) \geq B/s - l$, i.e.,

$$\begin{aligned}
&\sum_{k=1}^{k^*+1} \pi_k \phi_k + \sqrt{\phi_{k^*+1}} \sum_{k=k^*+2}^{K} \pi_k \sqrt{\phi_k} \\
=&\sum_{k=1}^{k^*} \pi_k \phi_k + \sqrt{\phi_{k^*+1}} \cdot \pi_{k^*+1} \sqrt{\phi_{k^*+1}} + \sqrt{\phi_{k^*+1}} \cdot \sum_{k=k^*+2}^{K} \pi_k \sqrt{\phi_k} \\
=&\sum_{k=1}^{k^*} \pi_k \phi_k + \sqrt{\phi_{k^*+1}} \cdot \sum_{k=k^*+1}^{K} \pi_k \sqrt{\phi_k} \\
\geq& B/s - l.
\end{aligned}$$

Thus,

$$\sum_{k=1}^{k^*} \pi_k \phi_k + \sqrt{\phi_{k^*+1}} \cdot \sum_{k=k^*+1}^{K} \pi_k \sqrt{\phi_k} \geq B/s - l,$$

which is actually

$$\frac{1}{\sqrt{\phi_{k^*+1}}} \cdot \frac{B/s - l - \sum_{k=1}^{k^*} \pi_k \phi_k}{\sum_{k=k^*+1}^{K} \pi_k \sqrt{\phi_k}} = A_{k^*+1} \leq 1.$$

Let

$$\lambda^* = \frac{\gamma \left(\sum_{k=k^*+1}^{K} \pi_k \sqrt{\phi_k} \right)^2}{s(\bar{\theta}^*)^2 \left(B/s - l - \sum_{k=1}^{k^*} \pi_k \phi_k \right)^2}.$$

For $k > k^*$, as $k^* \geq k'$, we have $p_k = 1$ and

$$\begin{aligned}
\sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}} &= \sqrt{\frac{\gamma}{s(\bar{\theta}^*)^2} \times \frac{s(\bar{\theta}^*)^2 \left(B/s - l - \sum_{k=1}^{k^*} \pi_k \phi_k \right)^2}{\gamma \left(\sum_{k=k^*+1}^{K} \pi_k \sqrt{\phi_k} \right)^2}} \times \sqrt{\frac{1}{\phi_k}} \\
&= \frac{1}{\sqrt{\phi_k}} \cdot \frac{B - l - \sum_{k=1}^{k^*} \pi_k \phi_k}{\sum_{k=k^*+1}^{K} \pi_k \sqrt{\phi_k}} \\
&= A_k.
\end{aligned}$$

Due to the increasing virtual costs in k, we have $A_k \leq A_{k^*+1} \leq 1$, for all $k \geq k^*+1$. So we have $\min\left\{1, \sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}}\right\} = A_k$ for all $k > k^*$.

Next, in the regime $k \leq k^*$, we consider two possibilities of k^*: $k^* > k'$ and $k^* = k'$. Recall that $k^* \geq k'$.

∗ Firstly, we focus on $k^* > k'$. For $k \in [k'+1, k^*]$, we have $p_k = 1$ and

$$\sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}} = \sqrt{\frac{\gamma}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}} \geq \sqrt{\frac{\gamma}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_{k^*}}} > 1.$$

The last inequality holds due to $Q(k^*, 1) < B/s - l$, i.e.,

$$\sum_{k=1}^{k^*} \pi_k \phi_k + \sum_{k=k^*+1}^{K} \pi_k \sqrt{\phi_k \phi_{k^*}} < B/s - l,$$

which leads to (by plugging the value of λ^*)

$$\begin{aligned}
\sqrt{\frac{\gamma}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_{k^*}}} &= \sqrt{\frac{\gamma}{s(\bar{\theta}^*)^2} \times \frac{s(\bar{\theta}^*)^2 \left(B/s - l - \sum_{k=1}^{k^*} \pi_k \phi_k\right)^2}{\gamma \left(\sum_{k=k^*+1}^{K} \pi_k \sqrt{\phi_k}\right)^2}} \cdot \frac{1}{\sqrt{\phi_{k^*}}} \\
&= \frac{B/s - l - \sum_{k=1}^{k^*} \pi_k \phi_k}{\sum_{k=k^*+1}^{K} \pi_k \sqrt{\phi_k}} \cdot \frac{1}{\sqrt{\phi_{k^*}}} \\
&> 1.
\end{aligned}$$

Thus, $\min\left\{1, \sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}}\right\} = 1 = A_k$.

For $k \leq k'$, since $z' \geq \frac{\phi_{k'}}{\phi_{k'+1}}$ (Claim 4.A.4) and $\phi_{k^*} \geq \phi_{k'}$, we have

$$\begin{aligned}
\sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}} &= \sqrt{\frac{\gamma}{s\left(\bar{\theta}^*\right)^2 \lambda^*}} \times \sqrt{\frac{z'}{\phi_{k'}}} \\
&\geq \sqrt{\frac{\gamma}{s\left(\bar{\theta}^*\right)^2 \lambda^*}} \times \sqrt{\frac{1}{\phi_{k'+1}}} \\
&\geq \sqrt{\frac{\gamma}{s\left(\bar{\theta}^*\right)^2 \lambda^*}} \times \sqrt{\frac{1}{\phi_{k^*}}} \\
&> 1.
\end{aligned}$$

Thus, we have $\min\left\{1, \sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}}\right\} = 1 = A_k$.

In summary, we have $\min\left\{1, \sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2\lambda^*\phi_k}}\right\} = A_k$ for all $k \leq k^*$ when $k^* > k'$.

∗ Secondly, we focus on $k^* = k'$. On the one hand, we have $k' \leq m^*$ according to Claim 4.A.4, and therefore, $k^* \leq m^*$. On the other hand, $Q(m^*, 1) \leq Q(m^*, z^*) < B/s - l$, so we have that $k^* \geq m^*$ by definition of $k^* = \max\{k : Q(k, 1) < B/s - l\}$. So, it must be that $k' = k^* = m^*$.
Recall in the proof of Claim 4.A.4 that $R(m^*, z^*) < \gamma\bar{\theta}^* = R(k', z') = R(m^*, z')$, i.e., $R(m^*, z^*) < R(m^*, z')$. And since $R(m, z)$ is increasing in z, we have $z^* < z'$. Since $Q(m, z)$ is decreasing in z, it must be that

$$Q(m^*, z') < Q(m^*, z^*) < B/s - l.$$

That is

$$Q(m^*, z') = \sum_{k=1}^{m^*} \pi_k \phi_k + \sqrt{\frac{\phi_{m^*}}{z'}} \sum_{k=m^*+1}^{K} \pi_k \sqrt{\phi_k} < B/s - l.$$

For $k \leq k^* = k'$, we have $p_k = \frac{z'}{\phi_{k'}} \cdot \phi_k$, and

$$\begin{aligned}
\sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}} &= \sqrt{\frac{\gamma}{s\left(\bar{\theta}^*\right)^2 \lambda^*}} \times \sqrt{\frac{z'}{\phi_{k'}}} \\
&= \sqrt{\frac{z'}{\phi_{m^*}}} \cdot \frac{B - l - \sum_{k=1}^{m^*} \pi_k \phi_k}{\sum_{k=m^*+1}^{K} \pi_k \sqrt{\phi_k}} \\
&> 1.
\end{aligned}$$

Thus, we have $\min\left\{1, \sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2\lambda^*\phi_k}}\right\} = 1 = A_k$.
In summary, we have $\min\left\{1, \sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2\lambda^*\phi_k}}\right\} = A_k$ for all $k \leq k^*$ when $k^* = k'$.

In conclusion, we have $A_k = \min\left\{1, \sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2\lambda^*\phi_k}}\right\}$, for all k, and A_k is non-increasing in k. The budget constraint is binding through similar calculation, so A is indeed a best response to p.

On the other hand, p is best response to A. For $k \leq k^*$, we have $0 < p_k \leq 1$ and $A_k = 1$. From $R(k', z') = \gamma\bar{\theta}^*$, we obtain

$$R(k',z') = 2\gamma\left(\sum_{k=1}^{k'}\pi_k\phi_k\frac{z'}{\phi_{k'}} + \sum_{k=k'+1}^{K}\pi_k\right) + \left(\bar{\theta}^*\right)^2(1-\bar{\theta}^*)(1-\gamma)s = \gamma\bar{\theta}^*,$$

i.e.,

$$2\gamma\left(\sum_{k=1}^{K}\pi_k p_k\right) + \left(\bar{\theta}^*\right)^2(1-\bar{\theta}^*)(1-\gamma)s = \gamma\bar{\theta}^*.$$

So

$$\begin{aligned}&\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{1}{A_k} - \frac{2}{\bar{\theta}^*}(\sum_{k=1}^{K}\pi_k p_k)\right) - \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma)\\ =&\frac{\gamma}{s(\bar{\theta}^*)^2}\left(1 - \frac{2}{\bar{\theta}^*}(\sum_{k=1}^{K}\pi_k p_k)\right) - \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma)\\ =&0.\end{aligned}$$

With the above equality, any $p_k \in [0,1]$ is a best response according to Lemma 4.A.1. For $k > k^*$, we have $p_k = 1$, and since $A_{k^*} < 1$, we have

$$\begin{aligned}&\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{1}{A_k} - \frac{2}{\bar{\theta}^*}(\sum_{k=1}^{K}\pi_k p_k)\right) - \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma)\\ >&\frac{\gamma}{s(\bar{\theta}^*)^2}\left(1 - \frac{2}{\bar{\theta}^*}(\sum_{k=1}^{K}\pi_k p_k)\right) - \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma)\\ =&0.\end{aligned}$$

So p is indeed a best response to A, as per Lemma 4.A.1. We can see that p_k is indeed non-decreasing.

- Case 3: $\frac{B/s-l}{\gamma\bar{\theta}^*} \geq \frac{Q(K,1)}{R(K,1)}$. Recall that $Q(m,z)$ is continuously decreasing in z and $R(m,z)$ is continuously increasing in z. Thus, $\frac{Q(m,z)}{R(m,z)}$ is decreasing in z. Notice that when $z = 0$, we have

$$\begin{aligned}\frac{Q(K,0)}{R(K,0)} &= \frac{\sum_{k=1}^{K}\pi_k\phi_k}{2\gamma\left(\frac{0}{\phi_m}\cdot\sum_{k=1}^{K}\pi_k\phi_k\right) + \left(\bar{\theta}^*\right)^2(1-\bar{\theta}^*)(1-\gamma)s}\\ &= \frac{\sum_{k=1}^{K}\pi_k\phi_k}{\left(\bar{\theta}^*\right)^2(1-\bar{\theta}^*)(1-\gamma)s}.\end{aligned}$$

If $\frac{\sum_{k=1}^{K} \pi_k \phi_k}{\left(\bar{\theta}^*\right)^2 (1-\bar{\theta}^*)(1-\gamma)s} > \frac{B/s-l}{\gamma\bar{\theta}^*}$, let $\tilde{z} \in (0,1]$ be such that $\frac{Q(K,\tilde{z})}{R(K,\tilde{z})} = \frac{B/s-l}{\gamma\bar{\theta}^*}$. Such a $\tilde{z}$ exists because $\frac{Q(K,1)}{R(K,1)} \leq \frac{B/s-l}{\gamma\bar{\theta}^*} < \frac{Q(K,0)}{R(K,0)}$. Then

$$p_k = \tilde{z} \cdot \frac{\phi_k}{\phi_K}, \forall k, \tag{45}$$

and

$$A_k = \frac{B/s - l}{\sum_{k=1}^{K} \pi_k \phi_k}, \forall k \tag{46}$$

constitute the mutual best response. To see this, we know that A is a best response to p. Let

$$\lambda^* = \frac{\gamma \left(\sum_{k=1}^{K} \pi_k \sqrt{p_k \phi_k} \right)^2}{s(\bar{\theta}^*)^2 \left(B/s - l\right)^2}.$$

Then

$$\begin{aligned}
\sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}} &= \sqrt{\frac{\gamma}{s(\bar{\theta}^*)^2} \times \frac{s(\bar{\theta}^*)^2 \left(B/s-l\right)^2}{\gamma \left(\sum_{k=1}^{K} \pi_k \sqrt{p_k \phi_k} \right)^2}} \times \sqrt{\frac{p_k}{\phi_k}} \\
&= \frac{B/s - l}{\sum_{k=1}^{K} \pi_k \sqrt{\frac{\tilde{z}}{\phi_K} \cdot \phi_k \cdot \phi_k}} \times \sqrt{\frac{\tilde{z}}{\phi_K}} \\
&= \frac{B/s - l}{\sum_{k=1}^{K} \pi_k \phi_k} = A_k.
\end{aligned}$$

As we assume $\sum_{k=1}^{K} \pi_k \phi_k > B/s - l$ without loss of generality to avoid the trivial solution of any selection probability being one, we have that $\frac{B/s-l}{\sum_{k=1}^{K} \pi_k \phi_k} < 1$. Thus, we get that $\min\left\{1, \sqrt{\frac{\gamma p_k}{s\left(\bar{\theta}^*\right)^2 \lambda^* \phi_k}}\right\} = A_k$, for all k. The budget constraint is binding as

$$\sum_{k=1}^{K} \pi_k \phi_k A_k = \frac{B/s - l}{\sum_{k=1}^{K} \pi_k \phi_k} \cdot \sum_{k=1}^{K} \pi_k \phi_k = B/s - l.$$

So A is indeed a best response to p, as per Lemma 4.A.2.

On the other hand, p is a best response to A. Notice that

$$A_k = \frac{B/s - l}{\sum\limits_{k=1}^{K} \pi_k \phi_k} = \frac{B/s - l}{Q(K, \tilde{z})}.$$

From $\frac{Q(K,\tilde{z})}{R(K,\tilde{z})} = \frac{B/s-l}{\gamma\bar{\theta}^*}$, we have

$$\begin{aligned}
\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{1}{A_k} - \frac{2}{\bar{\theta}^*}(\sum_{k=1}^{K} \pi_k p_k)\right) &= \frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{Q(K,\tilde{z})}{B/s - l} - \frac{2}{\bar{\theta}^*}\left(\sum_{k=1}^{K} \pi_k \tilde{z}\frac{\phi_k}{\phi_K}\right)\right) \\
&= \frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{R(K,\tilde{z})}{\gamma\bar{\theta}^*} - \frac{2}{\bar{\theta}^*}\left(\sum_{k=1}^{K} \pi_k \tilde{z}\frac{\phi_k}{\phi_K}\right)\right) \\
&= \frac{1}{s(\bar{\theta}^*)^3}\left(R(K,\tilde{z}) - 2\gamma\left(\sum_{k=1}^{K} \pi_k \tilde{z}\frac{\phi_k}{\phi_m}\right)\right) \\
&= \frac{1}{s(\bar{\theta}^*)^3} \times \left(\bar{\theta}^*\right)^2 (1 - \bar{\theta}^*)(1-\gamma)s \\
&= \frac{1}{\bar{\theta}^*}(1 - \bar{\theta}^*)(1-\gamma),
\end{aligned}$$

i.e.,$\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{1}{A_k} - \frac{2}{\bar{\theta}^*}\left(\sum\limits_{k=1}^{K} \pi_k p_k\right)\right) - \frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma) = 0$. So $pk = \tilde{z} \cdot \frac{\phi_k}{\phi_K}$, for all k, is a best response to A, as per Lemma 4.A.1. We can see that p_k is non-decreasing.

If $\frac{\sum\limits_{k=1}^{K} \pi_k \phi_k}{\left(\bar{\theta}^*\right)^2(1-\bar{\theta}^*)(1-\gamma)s} \leq \frac{B/s-l}{\gamma\bar{\theta}^*}$, then $p_k = 0$ for all k and

$$A_k = \frac{B/s - l}{\sum\limits_{k=1}^{K} \pi_k \phi_k}, \forall k, \tag{47}$$

constitute one mutual best response. To see this, on one hand, A is one best response to p. This is because when $p_k = 0$ for all k, any $A_k \in [0, 1]$ that satisfies budget constraint is a best response, according to Lemma 4.A.2. $A_k < 1$ indeed satisfies budget constraint. Actually, there are other formats of A that satisfy budget constraint being best response. We present the constant solution, which is non-increasing, for simplicity.

On the other hand, p is best response to A. We have

$$\frac{\gamma}{s(\bar{\theta}^*)^2}\cdot\frac{1}{A_k}-\frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma)$$
$$=\frac{\gamma}{s(\bar{\theta}^*)^2}\left(\frac{\sum_{k=1}^{K}\pi_k\phi_k}{B/s-l}\right)-\frac{1}{\bar{\theta}^*}(1-\bar{\theta}^*)(1-\gamma)\leq 0.$$

So $p_k = 0$ is a best response to A, according to Lemma 4.A.1.

From discrete to continuous costs. So far, we present the intersections of both players' best responses, i.e., an equilibrium of the min-max game. A_k of the equilibrium corresponds to the solution of the minimax optimization problem. We then convert the solution under discrete cost to the solution under continuous cost. The continuous cost can be considered as a special case of discrete cost by setting K to be infinity.

We begin with some key notations under continuous case. We denote the distribution of virtual cost in group i as ω_i. We can obtain it based on the connection between cost and virtual cost. Recall that the probability of group i is q_i. Thus, we have the distribution of virtual cost in all groups as $\sum_i q_i\omega_i(\phi) \triangleq \omega(\phi)$. As non-participants whose costs are higher than the cost threshold would not get payments, we only need to focus on participants and their virtual costs. Let $\phi_{i\,\min}$ and $\phi_{i\,\max}$ be the maximum and minimum value of virtual costs of participants in group i, respectively. Let $\phi_{\min}$ and $\phi_{\max}$ be the maximum and minimum value of virtual costs of participants, respectively. Naturally, $\phi_{\min} = \min_i \phi_{i\,\min}$ and $\phi_{\max} = \max_i \phi_{i\,\max}$.

We divide the interval $[\phi_{\min}, \phi_{\max}]$ into K numbers of sub-intervals with equal length $(\phi_{\max} - \phi_{\min})/K$. When K is very large, we can approximate the continuous distribution by K discrete probabilities

$$\pi_k = \int_{\phi_{\min}\frac{K-k+1}{K}+\phi_{\max}\frac{k-1}{K}}^{\phi_{\min}\frac{K-k}{K}+\phi_{\max}\frac{k}{K}} \omega(\phi)d\phi, \quad k = 1, ..., K,$$

which are integrals of $\omega(\phi)$ in sub-intervals. Consider the functions $Q(m,z)$ and $R(m,z)$ defined in (32) and (33). We define their continuous versions by replacing summations with integrals as follows:

$$Q_c(x) = \int_{\phi_{\min}}^{x} \phi\omega(\phi)d\phi + \sqrt{x}\int_{x}^{\phi_{\max}} \sqrt{\phi}\omega(\phi)d\phi$$

$$R_c(x) = 2\gamma \left(\frac{1}{x} \int_{\phi_{\min}}^{x} \phi\omega(\phi) d\phi + \int_{x}^{\phi_{\max}} \omega(\phi) d\phi \right) + \left(\bar{\theta}^*\right)^2 (1 - \bar{\theta}^*)(1-\gamma)s.$$

We can see that $Q\left(\frac{x-\phi_{\min}}{\phi_{\max}-\phi_{\min}}, 1\right) \to Q_c(x)$ and $R\left(\frac{x-\phi_{\min}}{\phi_{\max}-\phi_{\min}}, 1\right) \to R_c(x)$ as $K \to \infty$. As $\phi_m \to \phi_{m+1}, m = 1, .., K$ in the limit $K \to \infty$, we have $Q(m, z) \to Q(m+1, z)$ and $R(m, z) \to R(m+1, z)$. Since $Q(m+1, 1) = Q\left(m, \frac{\phi_m}{\phi_{m+1}}\right), R(m+1, 1) = R\left(m, \frac{\phi_m}{\phi_{m+1}}\right)$, according to Claim 4.A.2, we have $Q\left(m, \frac{\phi_m}{\phi_{m+1}}\right) \to Q(m, 1)$ and $R\left(m, \frac{\phi_m}{\phi_{m+1}}\right) \to R(m, 1)$. Thus, it suffices to consider $z = 1$ and leverage the limit $Q\left(\frac{x-\phi_{\min}}{\phi_{\max}-\phi_{\min}}, 1\right) \to Q_c(x)$ and $R\left(\frac{x-\phi_{\min}}{\phi_{\max}-\phi_{\min}}, 1\right) \to R_c(x)$ when characterizing the connection between $Q(m, z)$, $R(m, z)$ and $Q_c(x)$, $R_c(x)$.

Based on the connection between discrete Q and R and continuous Q_c and R_c, we adapt the solutions of discrete case to that of continuous case.

- Case 1: $\frac{B/s-l}{\gamma\bar{\theta}^*} < \frac{Q_c(\phi_{\min})}{R_c(\phi_{\min})}$, i.e., $(B/s - l)(2\gamma + \bar{\theta}^*(1 - \bar{\theta}^*)(1-\gamma)s) < \gamma\sqrt{\phi_{\min}} \cdot \int_{\phi_{\min}}^{\phi_{\max}} \sqrt{\phi}\omega(\phi) d\phi$. We have the optimal allocation rule for virtual cost ϕ

$$A(\phi) = \frac{1}{\sqrt{\phi}} \cdot \frac{B/s - l}{\int_{\phi_{\min}}^{\phi_{\max}} \sqrt{\phi}\omega(\phi) d\phi}.$$

  We can equivalently present the solution in terms of cost c:

$$A_i(c) = \frac{1}{\sqrt{\phi_i(c)}} \cdot \frac{B/s - l}{\sum\limits_i q_i \int_{c_{\min}}^{\hat{c}_i} \sqrt{\phi_i(c)} f_i(c) dc}.$$

- Case 2: $\frac{Q_c(\phi_{\min})}{R_c(\phi_{\min})} \le \frac{B/s-l}{\gamma\bar{\theta}^*} < \frac{Q_c(\phi_{\max})}{R_c(\phi_{\max})}$. Let $\phi' \in [\phi_{\min}, \phi_{\max})$ such that $\frac{Q_c(\phi')}{R_c(\phi')} = \frac{B/s-l}{\gamma\bar{\theta}^*}$.

  - Case 2(a): $\frac{B/s-l}{Q_c(\phi')} \le 1$. We have

$$A(\phi) = \begin{cases} \chi \triangleq \frac{B/s-l}{Q_c(\phi')}, & \text{if } \phi \le \phi', \\ \frac{1}{\sqrt{\phi}} \cdot \frac{B/s-l-\chi \int_{\phi_{\min}}^{\hat{\phi}} \phi\omega(\phi) d\phi}{\int_{\phi'}^{\phi_{\max}} \phi\omega(\phi) d\phi}, & \text{if } \phi < \phi'. \end{cases}$$

    The solution in terms of cost c is

$$A_i(c) = \begin{cases} \chi \triangleq \frac{B/s-l}{Q_c(\phi')}, & \text{if } \phi_i(c) \le \phi', \\ \frac{1}{\sqrt{\phi_i(c)}} \cdot \frac{B/s-l-\chi \sum\limits_i q_i \int_{\phi_{i\min}}^{\hat{\phi}} \phi_i(c)\omega_i(\phi_i) d\phi_i}{\sum\limits_i q_i \int_{\phi'}^{\phi_{i\max}} \sqrt{\phi_i(c)}\omega_i(\phi_i) d\phi_i}, & \text{if } \phi_i(c) > \phi'. \end{cases}$$

- Case 2(b): $\frac{B/s-l}{Q_c(\phi')} > 1$. Let ϕ^* be such that $Q_c(\phi^*) = B/s - l$. Then we have

$$A(\phi) = \begin{cases} 1, & \text{if } \phi \leq \phi^*, \\ \frac{1}{\sqrt{\phi}} \cdot \frac{B/s-l-\int_{\phi_{\min}}^{\phi^*} \phi\omega(\phi)d\phi}{\int_{\phi^*}^{\phi_{\max}} \phi\omega(\phi)d\phi}, & \text{if } \phi < \phi^*. \end{cases}$$

The solution in terms of cost c is

$$A_i(c) = \begin{cases} 1, & \text{if } \phi_i(c) \leq \phi^*, \\ \frac{1}{\sqrt{\phi_i(c)}} \cdot \frac{B/s-l-\sum\limits_i q_i \int_{\phi_{i\min}}^{\hat{\phi}} \phi_i(c)\omega_i(\phi_i)d\phi_i}{\sum\limits_i q_i \int_{\hat{\phi}}^{\phi_{i\max}} \sqrt{\phi_i(c)}\omega_i(\phi_i)d\phi_i}, & \text{if } \phi_i(c) > \phi^*. \end{cases}$$

- Case 3: $\frac{B/s-l}{\gamma\bar{\theta}^*} \geq \frac{Q_c(\phi_{\max})}{R_c(\phi_{\max})}$. We have

$$A(\phi) = \chi = \frac{B/s-l}{\int_{\phi_{\min}}^{\phi_{\max}} \phi\omega(\phi)d\phi}.$$

The solution in terms of cost c is

$$A_i(c) = \chi = \frac{B/s-l}{\sum\limits_i q_i \int_{\phi_{i\min}}^{\phi_{i\max}} \phi_i(c)\omega_i(\phi_i)d\phi_i}.$$

In summary, we can present the solution of continuous case for cost $c \leq \hat{c}_i$ in group i as follows:

$$A_i(c) = \begin{cases} \chi, & \text{if } \phi_i(c) \leq \hat{\phi}, \\ \frac{1}{\sqrt{\phi_i(c)}} \cdot \frac{B/s-l-\chi\sum\limits_i q_i \int_{\phi_{i\min}}^{\hat{\phi}} \phi_i(c)\omega_i(\phi_i)d\phi_i}{\sum\limits_i q_i \int_{\hat{\phi}}^{\phi_{i\max}} \sqrt{\phi_i(c)}\omega_i(\phi_i)d\phi_i}, & \text{if } \phi_i(c) > \hat{\phi}. \end{cases} \quad (48)$$

Here, the constants χ and $\hat{\phi}$ are defined as follows:

- If $\frac{B/s-l}{\gamma\bar{\theta}^*} < \frac{Q_c(\phi_{\min})}{R_c(\phi_{\min})}$, then $\chi = 0$ and $\hat{\phi} < \phi_{i\min}$ for all i.
- If $\frac{Q_c(\phi_{\min})}{R_c(\phi_{\min})} \leq \frac{B/s-l}{\gamma\bar{\theta}^*} < \frac{Q_c(\phi_{\max})}{R_c(\phi_{\max})}$, then $\chi = \min\left\{1, \frac{B/s-l}{Q_c(\phi')}\right\}$ where ϕ' satisfies $\frac{Q_c(\phi')}{R_c(\phi')} = \frac{B/s-l}{\gamma\bar{\theta}^*}$, and $\hat{\phi}$ satisfies $\frac{Q_c(\hat{\phi})}{\max\{1, R_c(\hat{\phi})/\gamma\bar{\theta}^*\}} = B/s - l$.
- If $\frac{B/s-l}{\gamma\bar{\theta}^*} \geq \frac{Q_c(\phi_{\max})}{R_c(\phi_{\max})}$, then $\chi = \frac{B/s-l}{\sum\limits_i q_i \int_{\phi_{i\min}}^{\phi_{i\max}} \phi_i(c)\omega_i(\phi_i)d\phi_i}$ and $\hat{\phi} = \phi_{\max}$.

$\square$

4.A.4 Proof of Corollary 4.4.1

Corollary 4.4.1 is a special case of Theorem 4.4.1. The proof is in Case 1 in the proof of Theorem 4.4.1. □

4.A.5 Proof of Proposition 4.4.1

Under condition 1) of Proposition 4.4.1, the solution falls into Case 1 in the proof of Theorem 4.4.1. Plugging the solution of Case 1 in the proof of Theorem 4.4.1 to the objective function, we have the following optimization problem over participation profile $\boldsymbol{\theta}^*$:

$$\min_{\boldsymbol{\theta}^*} \ T^*(\boldsymbol{\theta}^*) = \frac{\gamma}{s}\left(U(\boldsymbol{\theta}^*) - \frac{1}{\bar{\theta}^*}\right),$$

where

$$U(\boldsymbol{\theta}^*) \triangleq \frac{1}{\left(\bar{\theta}^*\right)^2} \cdot \frac{\left(\sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \sqrt{\phi_i(c;\boldsymbol{\theta}^*)} f_i(c) dc\right)^2}{B/s - l(\boldsymbol{\theta}^*)}.$$

Note that $\bar{\theta}^* = \sum_i q_i \bar{\theta}_i^*$ by the definition of average participation rate. Thus, $\frac{\partial \bar{\theta}^*}{\partial \bar{\theta}_i^*} = q_i$. Define

$$r(\boldsymbol{\theta}^*) \triangleq \left(\sum_i q_i \int_{c_{\min}}^{\hat{c}_i} \sqrt{\phi_i(c;\boldsymbol{\theta}^*)} f_i(c) dc\right)^2,$$

and recall that

$$l(\boldsymbol{\theta}^*) \triangleq \sum_j q_j \bar{\theta}_i^* (h(\hat{c}_j, \boldsymbol{\theta}_j^*; \boldsymbol{\alpha}_j) - g(\hat{c}_j, \boldsymbol{\theta}_j^*; \boldsymbol{\alpha}_j) - w(\bar{\theta}^*)),$$

and

$$\Delta_i \triangleq h(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i) - g(\hat{c}_i, \boldsymbol{\theta}_i; \boldsymbol{\alpha}_i).$$

We can write $U(\boldsymbol{\theta}^*)$ as follows:

$$U(\boldsymbol{\theta}^*) = \frac{1}{\left(\bar{\theta}^*\right)^2} \cdot \frac{r(\boldsymbol{\theta}^*)}{B/s - l(\boldsymbol{\theta}^*)} = \frac{1}{\left(\bar{\theta}^*\right)^2} \cdot \frac{r(\boldsymbol{\theta}^*)}{B/s - \sum_i q_i \bar{\theta}_i^* (\Delta_i - w(\bar{\theta}^*))}.$$

Before presenting the derivative of the objective function with respect to group i's participation ratio $\bar{\theta}_i^*$, i.e., $\frac{\partial T^*(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*}$, we give the derivative of $l(\boldsymbol{\theta}^*)$ with respect to $\bar{\theta}_i^*$, which appears in $\frac{\partial T^*(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*}$. Note that $\frac{\partial l(\boldsymbol{\theta})}{\partial \bar{\theta}_i^*} = \frac{\partial \sum_j q_j \bar{\theta}_j^* (\Delta_j - w(\bar{\theta}^*))}{\partial \bar{\theta}_j^*}$. That is,

$$
\begin{aligned}
& \frac{\partial \sum_j q_j \bar{\theta}_j^* (\Delta_j - w(\bar{\theta}^*))}{\partial \bar{\theta}_i^*} \\
= \; & \frac{\partial q_i \bar{\theta}_i^* (\Delta_i - w(\bar{\theta}^*))}{\partial \bar{\theta}_i^*} + \frac{\partial \sum_{j \neq i} q_j \bar{\theta}_j^* (\Delta_j - w(\bar{\theta}^*))}{\partial \bar{\theta}_i^*} \\
= \; & q_i (\Delta_i - w(\bar{\theta}^*)) + q_i \bar{\theta}_i^* \left(\frac{\partial \Delta_i}{\partial \bar{\theta}_i^*} - q_i w'(\bar{\theta}^*) \right) + \sum_{j \neq i} q_j \bar{\theta}_j^* \left(\frac{\partial \Delta_j}{\partial \bar{\theta}_i^*} - q_i w'(\bar{\theta}^*) \right) \\
= \; & q_i (\Delta_i - w(\bar{\theta}^*)) + q_i \bar{\theta}_i^* \frac{\partial \Delta_i}{\partial \bar{\theta}_i^*} + \sum_{j \neq i} q_j \bar{\theta}_j^* \frac{\partial \Delta_j}{\partial \bar{\theta}_i^*} - q_i w'(\bar{\theta}^*) \sum_j q_j \bar{\theta}_j^* \\
= \; & q_i (\Delta_i - w(\bar{\theta}^*)) + q_i \bar{\theta}_i^* \frac{\partial \Delta_i}{\partial \bar{\theta}_i^*} + \sum_{j \neq i} q_j \bar{\theta}_j^* \frac{\partial \Delta_j}{\partial \bar{\theta}_i^*} - q_i w'(\bar{\theta}^*) \bar{\theta}^* \quad \text{using } \sum_j q_j \bar{\theta}_j^* = \bar{\theta}^* \\
= \; & q_i \left(\bar{\theta}_i^* \frac{\partial \Delta_i}{\partial \bar{\theta}_i^*} + \Delta_i - w(\bar{\theta}^*) - \bar{\theta}^* w'(\bar{\theta}^*) \right) + \sum_{j \neq i} q_j \bar{\theta}_j^* \frac{\partial \Delta_j}{\partial \bar{\theta}_i^*}.
\end{aligned}
$$

The derivative of the objective function with respect to group i's participation ratio $\bar{\theta}_i^*$ is

$$
\frac{\partial T^*(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*} = \frac{\gamma}{s} \frac{\partial U(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*} + \frac{\gamma q_i}{s \left(\bar{\theta}^*\right)^2},
$$

where

$$
\begin{aligned}
\frac{\partial U(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*} = & \frac{\frac{\partial r(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*}}{(\bar{\theta}^*)^2 (B/s - l(\boldsymbol{\theta}^*))} - \frac{2 q_i r(\boldsymbol{\theta}^*)}{(\bar{\theta}^*)^3 (B/s - l(\boldsymbol{\theta}^*))} \\
& + \frac{q_i r(\boldsymbol{\theta}^*) (\bar{\theta}_i^* \frac{\partial \Delta_i}{\partial \bar{\theta}_i^*} + \Delta_i - w(\bar{\theta}^*) - \bar{\theta}^* w'(\bar{\theta}^*))}{(\bar{\theta}^*)^2 (B/s - l(\boldsymbol{\theta}^*))^2} \\
& + \frac{r(\boldsymbol{\theta}^*) \sum_{j \neq i} q_j \bar{\theta}_j^* \frac{\partial \Delta_j}{\partial \bar{\theta}_i^*}}{(\bar{\theta}^*)^2 (B/s - l(\boldsymbol{\theta}^*))^2}.
\end{aligned}
$$

We wish to characterize the conditions under which $\frac{\partial T^*(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*} \leq 0, \forall i$, i.e.,

$$
\frac{\partial T^*(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*} = \frac{\gamma}{s} \frac{\partial U(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*} + \frac{\gamma q_i}{s \left(\bar{\theta}^*\right)^2} \leq 0.
$$

This is equivalent to

$$
\frac{\partial T^*(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*} = \frac{\partial U(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*} + \frac{q_i}{\left(\bar{\theta}^*\right)^2} \leq 0,
$$

by removing γ/s, which is greater than zero. Plugging in $\frac{\partial U(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*}$, we have

$$\frac{\frac{\partial r(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*}}{(\bar{\theta}^*)^2(B/s - l(\boldsymbol{\theta}^*))} - \frac{2q_i r(\boldsymbol{\theta}^*)}{(\bar{\theta}^*)^3(B/s - l(\boldsymbol{\theta}^*))}$$
$$+ \frac{q_i r(\boldsymbol{\theta}^*)(\bar{\theta}_i^* \frac{\partial \Delta_i}{\partial \bar{\theta}_i^*} + \Delta_i - w(\bar{\theta}^*) - \bar{\theta}^* w'(\bar{\theta}^*))}{(\bar{\theta}^*)^2(B/s - l(\boldsymbol{\theta}^*))^2}$$
$$+ \frac{r(\boldsymbol{\theta}^*) \sum_{j \neq i} q_j \bar{\theta}_j^* \frac{\partial \Delta_j}{\partial \bar{\theta}_i^*}}{(\bar{\theta}^*)^2(B/s - l(\boldsymbol{\theta}^*))^2} + \frac{q_i}{(\bar{\theta}^*)^2} \leq 0.$$

Multiplying $(\bar{\theta}^*)^2(B/s - l(\boldsymbol{\theta}^*))^2$, which is greater than zero (recall that $(B/s - l(\boldsymbol{\theta}^*) > 0$ implicitly. Otherwise, there is no positive solution of allocation rule) at both sides, we have

$$\begin{aligned}
&\frac{\partial r(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*}(B/s - l(\boldsymbol{\theta}^*)) - \frac{2q_i r(\boldsymbol{\theta}^*)(B/s - l(\boldsymbol{\theta}^*))}{\bar{\theta}^*} + q_i r(\boldsymbol{\theta}^*)(\bar{\theta}_i^* \frac{\partial \Delta_i}{\partial \bar{\theta}_i^*} \\
&+ \Delta_i - w(\bar{\theta}^*) - \bar{\theta}^* w'(\bar{\theta}^*)) \\
&+ r(\boldsymbol{\theta}^*) \sum_{j \neq i} q_j \bar{\theta}_j^* \frac{\partial \Delta_j}{\partial \bar{\theta}_i^*} + q_i (B/s - l(\boldsymbol{\theta}^*))^2 \\
&\leq 0.
\end{aligned} \tag{49}$$

To ensure $\frac{\partial T^*(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*} \leq 0, \forall i$, we require the inequality in (49) to be true for all i. Define

$$\begin{aligned}
D_i(\boldsymbol{\theta}^*, B, \gamma, s, \mathbf{q}) \triangleq\ & \frac{(\bar{\theta}_i^* \Delta_i'(\bar{\theta}_i^*) + \Delta_i(\bar{\theta}_i^*) - w(\bar{\theta}^*))}{\bar{\theta}^*} + \frac{\frac{\partial r(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*}(B/s - l(\boldsymbol{\theta}^*))}{\bar{\theta}^* q_i r(\boldsymbol{\theta}^*)} \\
& - \frac{2(B/s - l(\boldsymbol{\theta}^*))}{(\bar{\theta}^*)^2} + \frac{(B/s - l(\boldsymbol{\theta}))^2}{\bar{\theta}^* r(\boldsymbol{\theta}^*)} + \frac{\sum_{j \neq i} q_j \bar{\theta}_j^* \frac{\partial \Delta_j}{\partial \bar{\theta}_i^*}}{q_i \bar{\theta}^*}.
\end{aligned} \tag{50}$$

If $w'(\bar{\theta}^*) \geq D_i(\boldsymbol{\theta}^*, , B, \gamma, s, \mathbf{q})$, for all $\boldsymbol{\theta}$ in which $\bar{\theta}_i^* \geq \bar{\theta}_{\min} > 0$, the inequality in (49) holds, i.e., we can obtain $\frac{\partial T^*(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*} \leq 0, \forall i$. Thus, the objective function is decreasing in the group participation ratio, and the optimal participation ratio for group i is one, for all groups i. □

4.A.6 Proof of Proposition 4.4.2

The idea of the proof is similar to the proof of Proposition 4.4.1. We can similarly achieve (49). Define

$$\begin{aligned}\delta_i(\boldsymbol{\theta}^*, B, \gamma, s, \mathbf{q}) \triangleq\ & \frac{w(\bar{\theta}^*) + \bar{\theta}^* w'(\bar{\theta}^*) - \Delta_i(\bar{\theta}_i^*)}{\bar{\theta}_i^*} - \frac{\frac{\partial r(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*}(B/s - l(\boldsymbol{\theta}^*))}{\bar{\theta}_i^* q_i r(\boldsymbol{\theta}^*)} \\ & + \frac{2(B/s - l(\boldsymbol{\theta}^*))}{\bar{\theta}_i^* \bar{\theta}^*} - \frac{(B/s - l(\boldsymbol{\theta}))^2}{\bar{\theta}_i^* r(\boldsymbol{\theta}^*)} - \frac{\sum_{j \neq i} q_j \bar{\theta}_j^* \frac{\partial \Delta_j}{\partial \bar{\theta}_i^*}}{q_i \bar{\theta}_i^*}.\end{aligned} \tag{51}$$

If $\Delta_i'(\bar{\theta}_i^*) \leq \delta_i(\boldsymbol{\theta}, B, \gamma, s, \mathbf{q})$, for all $\boldsymbol{\theta}$ in which $\bar{\theta}_i^* \geq \bar{\theta}_{\min} > 0$, the inequality in (49) holds, i.e., we can obtain $\frac{\partial T^*(\boldsymbol{\theta}^*)}{\partial \bar{\theta}_i^*} \leq 0, \forall i$. Thus, the objective function is decreasing in the group participation rate, and the optimal participation rate for group i is one, for all groups i. □

Chapter 5

OPTIMAL PRICING IN MARKETS WITH NON-CONVEX COSTS

5.1 Introduction

While there has been a long history of studying markets under convexity assumptions (such as convexity of cost functions, preferences, etc.) in economic theory, *non-convexities* are ubiquitous in most real-world markets. Non-convexities in cost functions arise due to start-up or shut-down costs, indivisibilities, avoidable costs, or simply economies of scale.

It has been widely noted in the literature that in the presence of non-convexities, there may be no linear prices (constant per quantity) that support a competitive market equilibrium [e.g., 127, 128], and it was suggested as early as 1980s that in these markets, one needs to consider using *price functions*, as opposed to the conventional prices [129]. Following the work of [130, 131], there have been many pricing schemes proposed in the literature. In particular, during the past decade, motivated by the deregulation of the electricity markets in the US and around the world, the problem of pricing in non-convex markets has attracted renewed interest from researchers, and there has been considerable work on this problem [132]. These pricing schemes are deployed in practice, and the operation and efficiency of our electricity markets relies on them [133].

Formally, the non-convex pricing problem is that, given an inelastic demand for a commodity from a number of consumers, a market operator seeks to satisfy the demand by purchasing the required amount from a group of competing suppliers with non-convex cost functions. The operator knows the suppliers' cost functions, and it announces a price/payment function for each supplier, which determines the payment to that supplier for producing different quantities. Each supplier then makes an individual decision about how much to produce in order to maximize its own profit. The key design question is how to devise the price functions in order to ensure certain economic properties for the market. We should remark that this problem is quite different from mechanism design, since the cost functions of the suppliers are known to the market operator, and the players can influence the market only by choosing their production level. However, as we shall see, the design of the price functions in these markets is challenging.

An important early approach to the pricing problem was the work of [17], sometimes referred to as integer programming (IP) pricing, which considered the class of non-convexities that arise from the start-up costs of the suppliers (with linear marginal costs). The paper proposes a clever pricing rule, based on solving a mixed-integer linear program and forcing the integral variables to their optimal values as a constraint. The scheme is economically efficient and has a nice dual interpretation. Modified versions of IP pricing have been proposed by [134, 135] and others. Another approach, proposed for the more general class of non-convex cost functions that are in the form of a start-up plus a convex (rather than linear) cost, is the minimum-uplift (MU) pricing [18], and its closely related refinement of [136], known as convex hull (CH) pricing. These schemes provide discriminatory uplifts to different suppliers to incentivize production, and the uplifts are minimal in a specific sense [19]. The possibility of having both positive and negative uplifts was also considered by [137, 138]. Other pricing schemes include the semi-Lagrangian relaxation (SLR) approach of [139] and the primal-dual (PD) approach of [140]. These schemes seek to find uniform linear prices that are revenue-adequate (but not supporting of the equilibrium). A survey on all the above pricing schemes was recently written by [132]. However, the overall desired properties, as well as the properties that each of the schemes satisfy, were not examined there. We formalize the desired properties considered in the literature in Section 5.2, and discuss the properties of the existing schemes in Section 5.5. Table 5.1 summarizes the common schemes and their properties.

Despite the large body of work on the pricing problem, the existing schemes have several shortcomings. For example, most of the existing schemes mentioned above are proposed for specific classes of non-convex cost functions, and cannot handle more general non-convexities. Furthermore, even the ones that are applicable for general cost functions fail to satisfy some of the key desired properties of the market, such as economic efficiency or supporting a competitive equilibrium. In addition, none of the existing schemes is accompanied by a computationally tractable algorithm for general non-convexities, and they typically rely on off-the-shelf heuristic solvers for mixed-integer programs that are known to be NP-hard.

In this paper, we propose a pricing scheme for markets with general non-convex costs that designs arbitrary parametric price functions and addresses all the issues mentioned above. Our approach seeks to find the optimal schedule (allocation) and the optimal pricing rule simultaneously, which generally allows for finding *economically more efficient* solutions. The ability to use arbitrarily specified parametric price functions (e.g., piece-wise linear, quadratic, etc.) enables our approach to design price functions that are *less discriminatory*, while still *supporting a competitive equilibrium*. Furthermore, our pricing scheme is accompanied by a *computationally efficient* (polynomial-time) approximation algorithm which allows one to find the approximately-optimal schedule and prices for general non-convex cost functions. Lastly, we extend the proposed pricing rule to *networked* markets, which, to the best of our knowledge, are not considered in any of the existing work.

Specifically, this paper makes the following contributions.

1. We propose a framework for pricing in markets with *general* non-convex costs, using *general* price functions (Section 5.3.1). Our scheme seeks to find the optimal price functions and allocations simultaneously, while imposing the equilibrium conditions as constraints. For this reason, our approach is generally economically more efficient than the existing methods, while satisfying the equilibrium conditions. Moreover, the ability to use general price forms allows one to obtain more uniform prices (smaller "uplifts").

2. We supplement our pricing scheme with a computationally efficient (polynomial-time) approximation algorithm for finding the allocations and prices (Section 5.3.2).

3. We extend our framework to *networked* markets, and also propose an approximation algorithm that can compute the solution efficiently for acyclic networks, a common scenario in electric distribution networks (Section 5.4).

4. We survey the common pricing schemes proposed in the literature for markets with non-convex costs and provide a compact summary of their properties (Section 5.5).

5. We evaluate the proposed method through extensive numerical examples, and show how it compares with the existing methods (Section 5.6).

5.2 Market Description and Pricing Objectives

While our goal in this paper is to design an economically and computationally efficient pricing scheme for non-convex *networked* markets, we begin with the problem of designing one for a *single* market, which is difficult in its own right. We return to the case of networked markets in Section 5.4. When the cost functions are non-convex, even this seemingly simple problem has proven to be challenging, and a wide variety of pricing schemes have been proposed for it in the literature. In the following, we describe the market model and survey the desired market properties.

5.2.1 Market Model

We consider a single market consisting of n competing suppliers (often referred to as firms or generators). The market is run by a market operator that seeks to satisfy a deterministic and inelastic demand d for a commodity in a single period. Each supplier i has a cost function $c_i(q_i)$ for producing quantity q_i, which may be non-convex.

The suppliers' cost functions are known by the operator, and the operator uses them to determine the prices. In particular, the operator announces price/payment functions $p_i(q_i)$, which determine the payment to supplier i when producing q_i. Note that, in general, the price functions can be different for different suppliers, but it is often desired to have close-to-uniform prices.

Upon the announcement of the price functions, each supplier i makes an individual decision, based on the price function $p_i(\cdot)$ and the cost function $c_i(\cdot)$, about how much to produce (and whether to participate in the market), in order to maximize its own profit, i.e., $p_i(q_i) - c_i(q_i)$. The suppliers are then paid for their production according to the payment function, and the demand (consumers) is charged for the total payment.

This model is classical, and has been studied in a wide variety of contexts, initially under the assumption of convex cost functions for production and linear pricing functions, but more recently under non-convex cost functions. Non-convex cost functions are particularly important in the context of electricity markets. As a result, there is a large literature focusing on non-convex pricing in electricity markets, see [132] for a recent survey. Often this literature assumes specific forms of non-convexities (e.g., startup/fixed costs), and specific forms of payment functions (e.g., linear plus uplift). The results from this literature have guided the design and operation of electricity markets across the world today.

5.2.2 Pricing Objectives

The key design question in the market described above is how to devise the payment functions. The goal of the operator is to (1) find the optimal quantities q_i^*, and (2) design the payment functions $p_i(\cdot)$ that ensure that the suppliers produce the optimal quantities q_i^*.

There is a huge design space for such payment functions, and there is a large literature evaluating proposals in the context of non-convex cost functions, e.g., [17–19, 132, 134, 136, 139–141].

From this literature, various desirable properties, which pricing rules attempt to satisfy, have emerged. The following is a summary of the most sought-after properties in this literature. Note that no existing rules satisfy all of these properties for general non-convex markets.

1. ***Market Clearing*** (a.k.a. ***Load Balancing***): The total supply is equal to the demand, i.e., $\sum_{i=1}^{n} q_i^* = d$.
2. ***Economic Efficiency***
   a) ***Minimal Production Cost*** *(Suppliers' Total Cost)*: The total production cost of the suppliers, i.e., $\sum_{i=1}^{n} c_i(q_i^*)$, is minimal.
   b) ***Minimal Payment*** *(Total Paid Cost)*: The total cost that is paid to the suppliers for the commodity, i.e., $\sum_{i=1}^{n} p_i(q_i^*)$, is minimal.
3. ***Incentivizing***
   a) ***Revenue Adequacy***: For every supplier, the net profit at the optimum is nonnegative, i.e., $p_i(q_i^*) - c_i(q_i^*) \geq 0$, for $i = 1, \ldots, n$.
   b) ***Support a Competitive Equilibrium***: The optimum production quantity for each supplier is a maximizer of its individual profit, i.e., $q_i^* \in \arg\max_{q_i} p_i(q_i) - c_i(q_i)$, or equivalently $p_i(q_i^*) - c_i(q_i^*) \geq \max_{q_i \neq q_i^*} p_i(q_i) - c_i(q_i)$, for $i = 1, \ldots, n$.
4. ***Simplicity and Uniformity***: The price functions are simple and interpretable (ideally linear: $p_i(q_i) = \lambda_i q_i$) and non-discriminatory (ideally uniform across suppliers: $p_i(q_i) = p(q_i)$).
5. ***Computational Tractability***: The optimal quantities and price functions can be computed/approximated in time that is polynomial in n.

A few remarks about these properties are warranted. Property 1 ensures that the demand is met. Property 2 is somewhat more elaborate and concerns the economic efficiency of the scheme, in terms of total expenditure. Even though in certain cases (e.g., in IP pricing of [17] for startup-plus-linear costs), the suppliers' total cost $\sum_{i=1}^{n} c_i(q_i)$ and the total paid cost $\sum_{i=1}^{n} p_i(q_i)$ match and are both minimal, there is an inevitable gap between the two in general. Ultimately, the quantity which determines the cost of satisfying the demand is the total payment to the suppliers $\sum_{i=1}^{n} p_i(q_i)$, and therefore Property 2b is arguably more crucial than Property 2a. However, ostensibly, because the price functions are not directly available while computing the optimal quantities, many pricing schemes have resorted to minimizing the total suppliers' cost $\sum_{i=1}^{n} c_i(q_i)$ as a surrogate for the paid cost. In this paper, we advocate a direct approach for minimizing the total payment.

Property 3 incentivizes the suppliers to follow the dispatch and produce the socially-optimal quantities. More specifically, Property 3a ensures that the suppliers do not lose by producing q_i^*, and further, Property 3b assures that it is in each supplier's best interest to follow the dispatch. Property 3b is generally a stronger condition than Property 3a, and if $p_i(0) = c_i(0) = 0\ \forall i$, then (3b) implies (3a).

Property 4 concerns having price forms that are "close to linear" (simple) and "close to uniform" (non-discriminatory), in some sense. One common approach to this is to use uniform linear prices plus a generator-dependent "uplift," i.e., $p_i(q_i) = \lambda q_i + u_i \mathbb{1}_{q_i = q_i^*}$, and try to minimize the uplifts u_i. As Property 4 is subjective by its nature, we allow arbitrary parametrized price functions in our scheme. However, we also examine our scheme when applied to the popular minimal-uplift approach. Note that Property 4 also rules out the use of "dictatorial" prices, in which the operator pays the cost (plus an ϵ) only at the desired amount, and pays nothing for any other amount produced.

The final property, Computational Tractability, is particularly challenging to address in the context of non-convex markets. Nearly all standard approaches work by computing the optimal production quantities and then deriving the prices from these quantities in some way. Under general non-convex cost functions, this first step is already computationally intractable. Thus, it is important to consider relaxations (approximations) of other properties if the goal is to enforce computational tractability. To that end, we consider approximate versions of the Incentivizing and Economic Efficiency conditions, which we discuss in Section 5.3.2. We propose an algorithm that satisfies these approximate versions, while being computationally tractable.

5.3 Proposed Scheme: Equilibrium-Constrained Pricing

Most existing schemes in the literature (see Section 5.5 for a detailed summary) are proposed for specific classes of non-convexities, and are not applicable for more general non-convex costs. Furthermore, even the ones that are applicable for more general cost functions either already lack some of the key properties (such as economic efficiency) or they lose those properties for more general costs. Additionally, the existing schemes are not accompanied by a computationally tractable algorithm for general non-convexities, and they typically rely on off-the-shelf heuristic solvers for mixed-integer programs that are NP-hard. This serves to emphasize that no existing pricing scheme satisfies the desired properties described in Section 5.2.2.

The main contribution of this paper is the introduction of a new pricing scheme, *Equilibrium-Constrained (EC) pricing*, which is applicable to general non-convex costs, allows using general parametric price functions, and satisfies all the desired properties outlined before, as long as the price class is general enough. The name of this scheme stems from the fact that we directly impose all the equilibrium conditions as constraints in the optimization problem for finding the best allocations, as opposed to adjusting the prices later to make the allocations an equilibrium. The optimization problem is, of course, non-convex, and non-convex problems are intractable in general. However, we also present a tractable approximation algorithm for approximately solving the proposed optimization.

We present the formulation of the optimization at the core of Equilibrium-Constrained pricing in Section 5.3.1, and then develop an efficient algorithm for solving the optimization problem approximately in Section 5.3.2.

5.3.1 Pricing Formulation

In this section, we propose a systematic approach for determining a pricing rule under generic non-convex costs that minimizes payments and satisfies the properties outlined in Section 5.2.2, while allowing flexibility in the choice of the form of price functions.

Specifically, consider a class of desired price functions, denoted by $\mathcal{P}$, which can be an arbitrary class such as linear, linear plus uplift, piece-wise linear, etc. This choice can be due to interpretability/uniformity reasons or other practical considerations. The core of Equilibrium-Constrained pricing is an optimization problem for finding the best price functions in $\mathcal{P}$ and the best allocations, at the same time. The operator is buying the commodity from the suppliers, on behalf of the consumers, and therefore its objective is to minimize the total cost incurred (total payment), subject to the equilibrium constraints. The optimization problem can be expressed as follows.

Equilibrium-Constrained (EC) Pricing:

$$p^* = \min_{\substack{q_1,\dots,q_n \\ p_1,\dots,p_n \in \mathcal{P}}} \quad \sum_{i=1}^{n} p_i(q_i) \tag{5.1a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \tag{5.1b}$$

$$p_i(q_i) - c_i(q_i) \geq 0, \quad i = 1, \dots, n \tag{5.1c}$$

$$p_i(q_i) - c_i(q_i) \geq \max_{q_i' \neq q_i} \; p_i(q_i') - c_i(q_i'), \quad i = 1, \dots, n. \tag{5.1d}$$

Constraints (5.1b), (5.1c), and (5.1d) are the Market Clearing, Revenue Adequacy, and Competitive Equilibrium conditions, respectively. Constraint (5.1d) can also be equivalently expressed as

$$p_i(q_i) - c_i(q_i) \geq p_i(q_i') - c_i(q_i') \quad \forall q_i' \neq q_i, \quad i = 1, \dots, n. \tag{5.2}$$

The key difference between EC pricing and the existing methods for pricing in non-convex markets is that it directly minimizes the total paid cost and seeks to find both the optimal allocations q_i^* and the optimal price functions $p_i^*(\cdot)$ simultaneously. The scheme enforces the desired economic properties as constraints, while allowing the use of any class of price functions, rather than imposing a fixed form for the price.

Since this scheme minimizes the total payments, and does not impose any explicit constraint on the total production cost, it would be natural to ask what happens to the latter quantity as we minimize the former. The minimum total production cost is defined as $c^* = \sum_{i=1}^{n} c_i(q_i^0)$, where

$$(q_1^0, \dots, q_n^0) = \arg\min_{q_1,\dots,q_n} \quad \sum_{i=1}^{n} c_i(q_i) \tag{5.3a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \tag{5.3b}$$

is the "minimal production cost" solution.

Remark 5.3.1. *It is easy to see, by relaxing the last constraint* (5.1d)*, and using constraint* (5.1c)*, that the optimal value of the optimization problem* (5.1) *is bounded* *below by the minimum total production cost. Mathematically, we have*

$$\begin{aligned} p^* \geq \quad & \min_{\substack{q_1,\ldots,q_n \\ p_1,\ldots,p_n}} \sum_{i=1}^{n} p_i(q_i) & \geq \quad & \min_{q_1,\ldots,q_n} \sum_{i=1}^{n} c_i(q_i) & = c^* \\ & \text{s.t.} \quad \sum_{i=1}^{n} q_i = d & & \text{s.t.} \quad \sum_{i=1}^{n} q_i = d & \\ & \qquad p_i(q_i) \geq c_i(q_i), \quad i = 1, \ldots, n. & & & \end{aligned}$$

In other words, the total production cost is always upper-bounded by the total payment. Therefore, minimizing the total payment puts a cap on the total production cost as well, while the opposite is not true in general (minimizing the total production cost can result in very high payments, which can be seen in, e.g., the case studies in Figs. 5.4a and 5.5a).

Remark 5.3.2. *We have imposed nearly all the desired properties as constraints in the optimization problem* (5.1)*, and it might not be clear whether this optimization problem has a solution at all. Indeed, there always exists a class of price functions for which problem* (5.1) *has a solution, and further the bound mentioned in Remark 5.3.1 is achieved.*

A naive choice of price function, often referred to as dictatorial pricing, is enough to prove this claim. In fact, one can check that for any price function of the form

$$p_i(q_i) \begin{cases} = c_i(q_i) & \text{for } q_i = q_i^0 \\ \leq c_i(q_i) & \text{for } q_i \neq q_i^0 \end{cases}$$

problem (5.1) *has an optimal solution $q^* = q^0$, and achieves the bound $p^* = c^*$.*

While Remark 5.3.2 asserts the existence of an optimal price function in general, the problem may not have a solution for certain specific classes of price functions. The key point is that problem (5.1) always allows using more sophisticated price forms (e.g., piece-wise linear) for which it will have a solution; and for any given choice of price form, it finds the best one, along with the optimal quantities.

Remark 5.3.3. *While in most scenarios the operator is buying the commodity from the suppliers on behalf of the consumers, and it makes sense to minimize the total payments $\sum_{i=1}^{n} p_i(q_i)$, in general one may seek to balance between the consumers' and the suppliers' costs. In other words, one can take the objective to* *be a linear combination of the consumers' cost* $\sum_{i=1}^n p_i(q_i)$ *and the suppliers' net cost (negative profit)* $\sum_{i=1}^n (c_i(q_i) - p_i(q_i))$*. Without loss of generality, the weighted sum can be normalized to an affine (i.e., convex) combination* $(1-\theta)\sum_{i=1}^n p_i(q_i) + \theta \sum_{i=1}^n (c_i(q_i) - p_i(q_i))$ *with parameter* θ*. The optimization can be expressed as follows.*

$$p^*_\theta = \min_{\substack{q_1,\ldots,q_n \\ p_1,\ldots,p_n \in \mathcal{P}}} \quad (1-2\theta)\sum_{i=1}^n p_i(q_i) + \theta \sum_{i=1}^n c_i(q_i) \tag{5.4a}$$

$$\text{s.t.} \quad \sum_{i=1}^n q_i = d \tag{5.4b}$$

$$p_i(q_i) - c_i(q_i) \geq 0, \quad i = 1, \ldots, n \tag{5.4c}$$

$$p_i(q_i) - c_i(q_i) \geq \max_{q_i' \neq q_i} \; p_i(q_i') - c_i(q_i'), \quad i = 1, \ldots, n. \tag{5.4d}$$

For the cases when the total payment $p^* = \sum_{i=1}^n p_i(q_i^*)$ *from the optimization problem* (5.1) *matches the lower bound* $c^* = \sum_{i=1}^n c_i(q_i^*)$ *(such as in the linear+uplift example of Section 5.3.1), the solution from* (5.4) *is the same as that of* (5.1)*, and the prices will be insensitive to parameter* θ*.*

It is worth mentioning that our algorithm proposed in Section 5.3.2 for solving (5.1) *is also capable of handling the weighted problem* (5.4)*. However, for the sake of simplicity, we focus on the case of* $\theta = 0$*.*

To be more explicit about the class of price functions, we consider a general parametric form for $\mathcal{P}$, specified by $p_i(q_i) := p(q_i; \alpha, \beta_i)$ with two types of parameters $\alpha \in \mathbb{R}^{l_1}$, and $\beta_i \in \mathbb{R}^{l_2}$ for $i = 1, \ldots, n$, where parameter α is shared among all the suppliers, and it constitutes the uniform component of the price, while parameter β_i is specific to supplier i. The parameters are in general constrained to be in some bounded sets $\mathcal{A} \subseteq \mathbb{R}^{l_1}$ and $\mathcal{B} \subseteq \mathbb{R}^{l_2}$, i.e., $\alpha \in \mathcal{A}$, and $\beta_i \in \mathcal{B}$ for all $i = 1, \ldots, n$. This parametric form is general enough that it encompasses all the assumed price forms in the literature. In particular, the linear-plus-uplift form ($p_i(q_i) = \lambda q_i + u_i \mathbb{1}_{q_i = \hat{q}_i}$) is a special case of this form, where the shared parameter is the uniform price λ, and the individual parameters are the amount and location of the uplifts $u_i, \hat{q}_i$. Using the general parametric form, the optimization problem (5.1) can be re-expressed as follows.

Parameterized Equilibrium-Constrained (EC) Pricing:

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ \alpha\in\mathcal{A} \\ \beta_1,\ldots,\beta_n\in\mathcal{B}}} \sum_{i=1}^{n} p(q_i;\alpha,\beta_i) \tag{5.5a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \tag{5.5b}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) \geq 0, \quad i = 1,\ldots,n \tag{5.5c}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) \geq \max_{q_i'\neq q_i} p(q_i';\alpha,\beta_i) - c_i(q_i'), \quad i = 1,\ldots,n. \tag{5.5d}$$

To show a concrete application of this general pricing scheme, we apply our framework to the popular class of linear-plus-uplift price functions, which has been a standard form considered in the electricity markets literature, e.g., [18, 136], and minimize the uplifts. We derive closed-form solutions for the optimal quantities and prices (for general cost functions). In this case, the total payment matches the total cost, which is the lowest theoretically possible. In contrast, the convex hull (CH) and minimum-uplift (MU) pricing schemes, which are the most closely related schemes and use the same type of price functions fail to achieve this bound and typically exhibit a large gap. The integer programming (IP) pricing, on the other hand, is capable of achieving the bound, but only for startup+linear cost functions, and not for more general cost functions such as startup+convex. (See Section 5.5 for more details on the existing schemes and their comparison with EC.)

Linear+Uplift Pricing

As mentioned earlier, using a linear uniform price plus an uplift term is a common choice of class of price functions, in practice. For this class, we have $p(q_i;\lambda,u_i,\hat{q}_i) = \lambda q_i + u_i \mathbb{1}_{q_i=\hat{q}_i}$, where $\lambda, u_1,\ldots,u_n \geq 0$. Without loss of generality, we can assume $\hat{q}_i^* = q_i^*$, i.e., the optimal location of uplift coincides with the desired production level, which is intuitive (See the e-companion for proof). The optimization problem

(5.5) can then be reduced to

$$p^*_{\text{uplift}} = \min_{\substack{q_1,\ldots,q_n \\ \lambda\geq 0 \\ u_1,\ldots,u_n\geq 0}} \quad \sum_{i=1}^{n}(\lambda q_i + u_i) \tag{5.6a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \tag{5.6b}$$

$$\lambda q_i + u_i - c_i(q_i) \geq 0, \quad i = 1,\ldots,n \tag{5.6c}$$

$$\lambda q_i + u_i - c_i(q_i) \geq \max_{q_i'\neq q_i} \lambda q_i' - c_i(q_i'), \quad i = 1,\ldots,n. \tag{5.6d}$$

Remark 5.3.4. *From Remark 5.3.1, we know that $p^*_{uplift} \geq c^*$. On the other hand, plugging the feasible point $\left(q_i = q_i^0\ \forall i,\ \lambda = 0,\ u_i = c_i(q_i^0)\ \forall i\right)$ into* (5.6) *results in $p^*_{uplift} \leq c^*$. Therefore $p^*_{uplift} = c^*$.*

Problem (5.6) has potentially many solutions, and the solution $q_i = q_i^0\ \forall i,\ \lambda = 0,\ u_i = c_i(q_i^0)\ \forall i$ corresponds to the naive pay-as-bid scheme, which is equivalent to having no uniform price and paying each supplier for its own cost. To obtain price functions that are close to uniform, it is desirable to pick a solution for which the uplifts are minimum (in ℓ_1 sense, for example). That is equivalent to adding a layer on top of the optimization problem (5.6) to pick the minimal-uplift solution among all the solutions, i.e.,

$$\min_{\boldsymbol{q},\lambda,\boldsymbol{u}} \quad \sum_{i=1}^{n} u_i \tag{5.7a}$$

$$\text{s.t.} \quad (\boldsymbol{q},\lambda,\boldsymbol{u}) \in \operatorname*{arg\,min}_{\boldsymbol{q},\lambda,\boldsymbol{u}} \text{(5.6a)} \tag{5.7b}$$

$$\text{s.t. (5.6b), (5.6c), (5.6d)} \tag{5.7c}$$

where $\boldsymbol{q}$ and $\boldsymbol{u}$ denote $(q_1,\ldots,q_n)$ and $(u_1,\ldots,u_n)$, respectively.

Let us define Λ as the set of all λ's for which the linear price λq lies below all the cost functions, i.e.,

$$\Lambda = \{\lambda \geq 0 \mid \lambda q \leq c_i(q),\ \forall q, \forall i\}\,. \tag{5.8}$$

Figure 5.1 illustrates this set for an example with three non-convex costs.

The solutions to problems (5.6) and (5.7) can be found in closed-form, and the following summarizes the results.

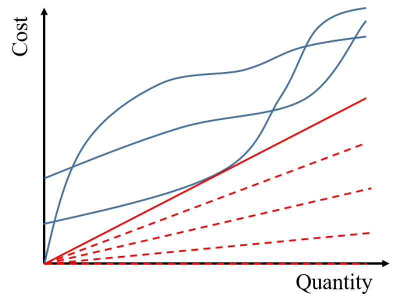


Figure 5.1: An illustration of the set Λ for an example with 3 non-convex cost functions. The three blue curves are the cost functions. The (dashed and solid) red lines lie below all the cost functions and their slopes are in Λ. The (slope of the) solid red line corresponds to the largest element of Λ.

Proposition 5.3.1. *The set of optimal solutions of problem* (5.6) *is given by*

$$\begin{cases} q_i^* = q_i^0, \ \forall i \\ \lambda^* \in \Lambda \\ u_i^* = c_i(q_i^*) - \lambda^* q_i^*, \ \forall i. \end{cases}$$

Proposition 5.3.2. *Problem* (5.7) *has a unique optimal solution as*

$$\begin{cases} q_i^* = q_i^0, \ \forall i \\ \lambda^* = \max \ \Lambda \\ u_i^* = c_i(q_i^*) - \lambda^* q_i^*, \ \forall i. \end{cases}$$

See the e-companion for proofs.

Note that there were two potential alternatives to the two-stage optimization in (5.7) for picking a minimum-uplift solution. One may have attempted to enforce uniformity as a constraint. However, the problem with this is that imposing, e.g., box constraints on u requires knowledge of reasonable upper-bounds on the uplifts, which may not be available; and on the other hand, insisting on exact uniformity makes the problem infeasible in most non-convex cases. The other alternative is to minimize a combination of the two objectives in (5.6) and (5.7). In this case, the weighted objective becomes $\sum_{i=1}^{n}(\lambda q_i + \gamma u_i)$ for some appropriate constant γ, and it is not hard to show that the solution will be the same as that of the proposed two-stage optimization.

5.3.2 An Efficient Approximation Algorithm

The optimization problem (5.5) defines a pricing rule that satisfies the desired properties in any non-convex market. For specific classes of cost functions, similar to the existing approaches, one may be able to solve this optimization problem using off-the-shelf solvers. For generic non-convex cost functions, however, there is no existing algorithm that can solve the optimization problem (5.5) to optimality. Furthermore, even finding an approximate solution, e.g., by discretizating the variables, requires a brute-force search, which quickly becomes intractable. In this section, we design a computationally efficient algorithm for solving the problem (5.5) approximately, based on decomposing it into smaller pieces, which works for general non-convex cost functions. This approximation algorithm can also be used to provide tractable calculations of some of the other non-convex pricing rules such as IP pricing.

Before going through the details of the algorithm, let us define the notion of an approximate solution to (5.5), which we consider. One could define an approximate solution as a value that is close enough, in a certain sense, to the optimal solution $(q_1^*, \ldots, q_n^*, \alpha^*, \beta_1^*, \ldots, \beta_n^*)$. However, no matter how close is that approximation to the optimal solution, that per se does not guarantee anything about the properties that the scheme will satisfy. Instead, we define an approximate solution to (5.5) as a set of quantities $q_1, \ldots, q_n$ and price parameters $\alpha, \beta_1, \ldots, \beta_n$ for which the Market Clearing condition holds exactly, the Revenue Adequacy and Competitive Equilibrium conditions are relaxed by an ϵ, and the total payment is at most $n\epsilon$ away from the optimal. More formally, it is defined as follows.

Definition 5.3.1. $(q_1, \ldots, q_n, \alpha, \beta_1, \ldots, \beta_n)$ *is called an* ϵ-approximate solution *to* (5.5) *if it satisfies*

$$\sum_{i=1}^{n} q_i = d, \qquad \textit{(Market Clearing)}$$

$$p(q_i; \alpha, \beta_i) - c_i(q_i) + \epsilon \geq 0, \quad i = 1, \ldots, n, \qquad \textit{(ϵ-Revenue Adequacy)}$$

$$p(q_i; \alpha, \beta_i) - c_i(q_i) + \epsilon \geq p(q_i'; \alpha, \beta_i) - c_i(q_i'), \quad \forall q_i' \neq q_i, \quad i = 1, \ldots, n, \qquad \textit{(ϵ-Competitive Equilibrium)}$$

and

$$\sum_{i=1}^{n} p(q_i; \alpha, \beta_i) \leq p^* + n\epsilon. \qquad \textit{(ϵ-Economic Efficiency)}$$

Given this notion of an approximate solution, we can move towards designing the algorithm. The optimization problem (5.5) looks highly coupled, at first, since the constraints share a lot of common variables. However, one can see that, for a fixed value of α, the objective becomes additively separable in (q_i, β_i). Furthermore (again for fixed α), constraints (5.5c),(5.5d) involve only the i-th variables (q_i, β_i) for each i. Although the Market Clearing condition still couples the variables together, this observation allows us to reformulate (5.5) as

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ \alpha\in\mathcal{A}}} \quad \sum_{i=1}^{n} g_i(q_i;\alpha) \tag{5.9a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d, \tag{5.9b}$$

where

$$g_i(q;\alpha) = \min_{\beta_i\in\mathcal{B}} \quad p(q;\alpha,\beta_i) \tag{5.10a}$$

$$\text{s.t.} \quad p(q;\alpha,\beta_i) - c_i(q) \geq 0, \tag{5.10b}$$

$$p(q;\alpha,\beta_i) - c_i(q) \geq p(q';\alpha,\beta_i) - c_i(q'), \quad \forall q' \neq q, \tag{5.10c}$$

for all $i = 1, \ldots, n$.

Therefore, for any fixed value of α and q_i, the optimization over β_i can be done individually, as in (5.10). What remains to address, however, is the coupling of the variables as a result of the Market Clearing constraint. One naive approach would be to simply try all possible choices of $(q_1, \ldots, q_n)$, and pick the one that has the minimum objective value. This is very inefficient. Instead, we take a dynamic programming approach, and group pairs of variables together, defining a new variable as their *parent*. We then group the parents together, and continue this process until we reach the *root*, i.e., where there is only one node. During this procedure, at each new node i, we need to solve the following (small) problem

$$\begin{aligned} g_i(q;\alpha) = \min_{q_j,q_k} \quad & g_j(q_j;\alpha) + g_k(q_k;\alpha) \\ \text{s.t.} \quad & q_j + q_k = q, \end{aligned} \tag{5.11}$$

for every q, where j and k are the *children* of i. At the root of the tree, we will be able to compute $g_{\text{root}}(d;\alpha)$. Figure 5.2 shows an example of the created binary tree for this procedure for $n = 8$. This procedure can be repeated for different values of α, and the optimal value p^* can be computed as $\min_\alpha g_{\text{root}}(d;\alpha)$.

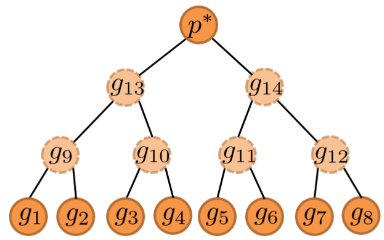


Figure 5.2: An example of the binary tree defined by Algorithm 5.4 for $n = 8$. The faded circles correspond to the added dummy nodes.

The problem with recursion (5.11) is that it requires an infinite-dimensional computation at every step, since the values of $g_i(q;\alpha)$ need to be computed for every q. To get around this issue, we note that the variables q_i live in the bounded set $[0, d]$, and hence can be discretized to lie in a finite set Q, such that every possible q_i is at most $\delta(\epsilon)$ away from some point in Q. Similarly, if the α and β_i's are continuous variables, we can discretize the bounded sets $\mathcal{A}$ and $\mathcal{B}$ into some finite sets $\mathcal{A}'$ and $\mathcal{B}'$, such that every point in $\mathcal{A}$ (or $\mathcal{B}$) is at most $\delta(\epsilon)$ away, in infinity-norm sense, from some point in $\mathcal{A}'$ (or $\mathcal{B}'$). See the e-companion for details.

For finding an ϵ-approximate solution, (5.10) is relaxed to

$$g_i(q;\alpha) = \min_{\beta_i \in \mathcal{B}'} \quad p(q;\alpha,\beta_i) \tag{5.12a}$$

$$\text{s.t.} \quad p(q;\alpha,\beta_i) - c_i(q) + \epsilon \geq 0, \tag{5.12b}$$

$$p(q;\alpha,\beta_i) - c_i(q) + \epsilon \geq p(q';\alpha,\beta_i) - c_i(q'), \quad \forall q' \neq q, \tag{5.12c}$$

for all $i = 1, \ldots, n$, and (5.11) remains the same, except the variables (q_j, q_k) take values in Q, i.e.,

$$\begin{aligned} g_i(q;\alpha) = \min_{q_j, q_k \in Q} \quad & g_j(q_j;\alpha) + g_k(q_k;\alpha) \\ \text{s.t.} \quad & q_j + q_k = q, \end{aligned} \tag{5.13}$$

for all $i > n$. We denote the optimizer of (5.12) by $b_i(q;\alpha)$, and the optimizer of (5.13), which is a pair of quantities (q_j, q_k), by $x_i(q;\alpha)$. The full procedure is summarized in psuedocode in Algorithm 5.4.

While not immediately clear, the proposed approximation algorithm can be shown to run in time that is polynomial in both n and $1/\epsilon$ (in fact, linear in n). Further, the solution it provides is ϵ-accurate under a mild smoothness assumption on the cost and price functions, which holds true for almost any function considered in

Algorithm 5.4 Find an ϵ-approximate solution to the optimal pricing problem (5.5).

Input: $n, c_1(\cdot), \ldots, c_n(\cdot), p(\cdot;\cdot), \epsilon$
for α in $\mathcal{A}'$ **do**
    $S = 1 : n$
    for [**do**for the leaves]i in S
        compute $g_i(q;\alpha)$ **for all** q in Q, using (5.12)
    end for
    while [**do**while not reached the root]$|S| > 2$
        $S_{\text{new}} = S(\text{end}) + 1 \ : \ S(\text{end}) + \lceil \frac{|S|}{2} \rceil$
        for [**do**for the intermediate nodes]i in S_{new}
            $[j, k]$ = indices of children of i
            if $k = \emptyset$ **then**
                $g_i(\cdot;\alpha) = g_j(\cdot;\alpha)$
            else[it has two children]
                compute $g_i(q;\alpha)$ **for all** q in Q, using (5.13)
            end if
        end for
        $S = S_{\text{new}}$
    end while
    $[j, k] = S$
    compute $g_{\text{root}}(d;\alpha)$, using (5.13)   # at the root
end for
$\alpha^* = \underset{\alpha \in \mathcal{A}'}{\arg\min} \quad g_{\text{root}}(d;\alpha)$
$q^*_{\text{root}} = d$
for $i = \text{root} : -1 : n + 1$ **do**
    $[q^*_j, q^*_k] = x_i(q^*_i;\alpha^*)$, where $[j, k]$ = indices of children of i
end for
for $i = n : -1 : 1$ **do**
    $\beta^*_i = b_i(q^*_i;\alpha^*)$
end for
return $(q^*_1, \ldots, q^*_n, \alpha^*, \beta^*_1, \ldots, \beta^*_n)$

the literature. These two results are summarized in the following theorem, which is proven in the e-companion.

Theorem 5.3.1. *Consider $c_i(\cdot)$ and $p(\cdot;\cdot)$ that have at most a finite number of discontinuities and are Lipschitz on each continuous piece of their domain. Algorithm 5.4 finds an ϵ-approximate solution to the optimal pricing problem* (5.5) *with running time $O\left(n(1/\epsilon)^{l_1+l_2+2}\right)$, where n is the number of suppliers, and l_1 and l_2 are the number of shared and individual parameters in the price, respectively.*

It is worth emphasizing that while there are $l_1 + nl_2$ variables in the price functions in total, parameters l_1 and l_2 do not scale with n, and are typically very small constants. For example, for the so-called linear-plus-uplift price functions $l_1 = l_2 = 1$. Therefore, the algorithm is very efficient.

We should also remark that if one requires the total payment in Definition 5.3.1 to be at most ϵ (rather than $n\epsilon$) away from the optimal p^*, the running time of our algorithm will still be polynomial in both n and $1/\epsilon$, i.e., $O\left(n^3(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$. See the e-companion for details.

5.4 Equilibrium-Constrained Pricing for Networked Markets

We now consider the more general problem of finding an efficient pricing scheme in a networked market. The networked market we consider has n suppliers, located at the nodes (vertices) $V = \{1, \ldots, n\}$ of a network, and connected through lines (edges) E, where, without loss of generality, the edges are defined to be from the smaller node to the larger node (i.e., $\forall(i, j) \in E,\ i < j$). The i-th supplier has a cost function $c_i(q_i)$ for producing quantity q_i, which may be non-convex, as before, and there is an inelastic demand d_i at each node i. The lines connecting the nodes can possibly have certain capacities for the flows they can carry. We denote the flow of any line $e = (i, j)$, from i to j, by f_e, and its limits (capacity) by $\underline{f_e}$ and $\overline{f_e}$ (the flow from j to i is $-f_e$).

Note that if there are multiple suppliers co-located in a market, we can simply assign them each their own vertex, and connect them through paths with infinite capacities. In other words, a node with multiple suppliers can be simply replaced with a "line graph" composed of those suppliers and infinite-capacity edges.

5.4.1 Pricing Formulation

A key benefit of EC pricing is the ease of generalization to the networked setting. There are no current pricing rules that can be readily applied to the networked case. In this setting, our Equilibrium-Constrained pricing can be formulated as the following optimization problem.

Networked Equilibrium-Constrained (EC) Pricing:

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ \{f_e\}_{e\in E} \\ p_1,\ldots,p_n\in\mathcal{P}}} \quad \sum_{i=1}^{n} p_i(q_i) \tag{5.14a}$$

$$\text{s.t.} \quad q_i - d_i = \sum_{\substack{j \\ (i,j)\in E}} f_{(i,j)} - \sum_{\substack{j \\ (j,i)\in E}} f_{(j,i)}, \quad i = 1,\ldots,n \tag{5.14b}$$

$$\underline{f_e} \le f_e \le \overline{f_e}, \quad e \in E \tag{5.14c}$$

$$p_i(q_i) - c_i(q_i) \ge 0, \quad i = 1,\ldots,n \tag{5.14d}$$

$$p_i(q_i) - c_i(q_i) \ge \max_{q_i'\neq q_i} p_i(q_i') - c_i(q_i'), \quad i = 1,\ldots,n. \tag{5.14e}$$

The objective is the total payment, as discussed before, and the optimization is over quantities q_i, line flows f_e, and price functions $p_i \in \mathcal{P}$. Constraint (5.14b) is the Market Clearing condition (or Flow Conservation) for each individual node, i.e., the net production at each node should be equal to its outgoing flow. Constraint (5.14c) enforces the line limits (Capacity Constraints). Constraints (5.14d) and (5.14e) are Revenue Adequacy and Competitive Equilibrium, respectively, as before. The key difference between the networked setting and the single-market one is that here the Market Clearing condition is spread across the network, and we have to solve the problem for the flows as well.

Remark 5.4.1. *When the capacity constraints* (5.14c) *are relaxed ($\underline{f_e} = -\infty, \overline{f_e} = \infty$, $\forall e \in E$), the networked problem reduces to the single-market one. In this case, the solution to the optimization problem* (5.14) *reduces to that of* (5.1)*. That is because the only constraint involving the flows would be* (5.14b)*, and we can always find flows that satisfy it, as long as $\sum_{i=1}^{n} q_i - \sum_{i=1}^{n} d_i = 0$, which is the conventional Market Clearing condition.*

Assuming a parametric form $p_i(q_i) \coloneqq p(q_i; \alpha, \beta_i)$ for $\mathcal{P}$, with shared parameters α and individual parameters β_i as before, the proposed pricing can be expressed as follows.

Parameterized Networked Equilibrium-Constrained (EC) Pricing:

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ \{f_e\}_{e\in E} \\ \alpha\in\mathcal{A} \\ \beta_1,\ldots,\beta_n\in\mathcal{B}}} \quad \sum_{i=1}^{n} p(q_i;\alpha,\beta_i) \tag{5.15a}$$

$$\text{s.t.} \quad q_i - d_i = \sum_{\substack{j \\ (i,j)\in E}} f_{(i,j)} - \sum_{\substack{j \\ (j,i)\in E}} f_{(j,i)}, \quad i = 1,\ldots,n \tag{5.15b}$$

$$\underline{f_e} \le f_e \le \overline{f_e}, \quad e \in E \tag{5.15c}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) \ge 0, \quad i = 1,\ldots,n \tag{5.15d}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) \ge \max_{q_i'\neq q_i} p(q_i';\alpha,\beta_i) - c_i(q_i'), \quad i = 1,\ldots,n \tag{5.15e}$$

5.4.2 An Efficient Approximation Algorithm

For certain classes of non-convexities, the optimization problem (5.15) can still be solved using off-the-shelf solvers, similar to those used in the other methods for the no-network case. However, those algorithms cannot handle more general classes of non-convexities. In this section, we develop a computationally efficient approximation algorithm for general non-convex costs, for a special class of networks.

A special yet important class of networks are *acyclic* networks, which are a typical topology in many markets, including *electricity distribution networks*. Acyclic networks have a *tree* topology (they do not have cycles), which allows us to devise an efficient algorithm for them. In the remainder of this section, we limit our attention to these networks. The main ideas extend directly to more general networks, as long as there are not "too many cycles" in the network in some sense (i.e., bounded tree-width networks). We have focused on the acyclic case due to space constraints.

Without loss of generality, let us denote the first node as the *root* of the tree, and nodes with only one neighbor as the *leaves*. Every node (except the root) has a unique *parent*, defined as the first node on the unique path connecting it to the root node. The set of nodes that have a given node i as their parent is said to be node i's *children*. It can be shown that any tree with arbitrary degree can be transformed into a *binary tree*, i.e., a tree where each node has a unique parent and at most 2 children, with $O(n)$ nodes (see the e-companion). Thus, we can focus on binary trees.

For a node i, let $\text{ch}_1(i), \text{ch}_2(i)$ denote its children ($\text{ch}_1(i) = \emptyset$ and/or $\text{ch}_2(i) = \emptyset$ when i has less than two children). The problem can then be written as

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ f_1,\ldots,f_n \\ \alpha\in\mathcal{A} \\ \beta_1,\ldots,\beta_n\in\mathcal{B}}} \quad \sum_{i=1}^{n} p(q_i;\alpha,\beta_i) \tag{5.16a}$$

$$\text{s.t.} \quad q_i - d_i = f_{\mathrm{ch}_1(i)} + f_{\mathrm{ch}_2(i)} - f_i, \quad i = 1,\ldots,n \tag{5.16b}$$

$$\underline{f_i} \le f_i \le \overline{f_i}, \quad i = 1,\ldots,n \tag{5.16c}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) \ge 0, \quad i = 1,\ldots,n \tag{5.16d}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) \ge \max_{q_i'\ne q_i} \ p(q_i';\alpha,\beta_i) - c_i(q_i'), \quad i = 1,\ldots,n \tag{5.16e}$$

where f_i represents the incoming flow to each node i from its parent, and $\underline{f_{\mathrm{root}}} = \overline{f_{\mathrm{root}}} = 0$.

Similarly as in the single-market case, we define an ϵ-approximate solution to this problem.

Definition 5.4.1. $(q_1,\ldots,q_n,f_1,\ldots,f_n,\alpha,\beta_1,\ldots,\beta_n)$ *is called an* ϵ-approximate solution *to* (5.16) *if it satisfies*

$$|q_i - d_i - f_{\mathrm{ch}_1(i)} - f_{\mathrm{ch}_2(i)} + f_i| \le \epsilon, \quad i = 1,\ldots,n, \qquad \textit{(ϵ-Load Balancing)}$$

$$\underline{f_i} \le f_i \le \overline{f_i}, \quad i = 1,\ldots,n, \qquad \textit{(Flow Limit)}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) + \epsilon \ge 0, \quad i = 1,\ldots,n, \qquad \textit{(ϵ-Revenue Adequacy)}$$

$$p(q_i;\alpha,\beta_i) - c_i(q_i) + \epsilon \ge p(q_i';\alpha,\beta_i) - c_i(q_i'), \quad \forall q_i' \ne q_i, \quad i = 1,\ldots,n, \qquad \textit{(ϵ-Competitive Equilibrium)}$$

$$\sum_{i=1}^{n} p(q_i;\alpha,\beta_i) \le p^* + n\epsilon. \qquad \textit{(ϵ-Economic Efficiency)}$$

The main difference from the definition in the single-market case is that the Market Clearing condition has been replaced with ϵ-Load Balancing and exact Flow Limit conditions here.

Note that the minimization over the variables β_i in problem (5.16) can be done "internally," and the problem can be re-expressed as

$$p^* = \min_{\substack{q_1,\ldots,q_n \\ f_1,\ldots,f_n \\ \alpha \in A}} \quad \sum_{i=1}^{n} g_i(q_i;\alpha) \tag{5.17a}$$

$$\text{s.t.} \quad q_i - d_i = f_{\text{ch}_1(i)} + f_{\text{ch}_2(i)} - f_i, \quad i = 1,\ldots,n \tag{5.17b}$$

$$\underline{f_i} \le f_i \le \overline{f_i}, \quad i = 1,\ldots,n \tag{5.17c}$$

where

$$g_i(q;\alpha) = \min_{\beta_i \in \mathcal{B}} \quad p(q;\alpha,\beta_i) \tag{5.18a}$$

$$\text{s.t.} \quad p(q;\alpha,\beta_i) - c_i(q) \ge 0, \tag{5.18b}$$

$$p(q;\alpha,\beta_i) - c_i(q) \ge p(q';\alpha,\beta_i) - c_i(q'), \quad \forall q' \ne q, \tag{5.18c}$$

for all $i = 1,\ldots,n$.

The key insight is that the tree structure of the constraints (5.17b) allows us to write the optimization problem in a recursive form as follows.

$$p^* = \min_{\alpha} \; h_{\text{root}}(0;\alpha) \tag{5.19}$$

where

$$h_i(f_i;\alpha) = \min_{q_i, f_{\text{ch}_1(i)}, f_{\text{ch}_2(i)}} \quad g_i(q_i;\alpha) + h_{\text{ch}_1(i)}(f_{\text{ch}_1(i)};\alpha) + h_{\text{ch}_2(i)}(f_{\text{ch}_2(i)};\alpha) \tag{5.20a}$$

$$\text{s.t.} \quad q_i - d_i = f_{\text{ch}_1(i)} + f_{\text{ch}_2(i)} - f_i \tag{5.20b}$$

$$\underline{f_{\text{ch}_1(i)}} \le f_{\text{ch}_1(i)} \le \overline{f_{\text{ch}_1(i)}} \tag{5.20c}$$

$$\underline{f_{\text{ch}_2(i)}} \le f_{\text{ch}_2(i)} \le \overline{f_{\text{ch}_2(i)}} \tag{5.20d}$$

for all $i = 1,\ldots,n$.

Now, this recursive form is amenable to dynamic programming. However, since the variables are continuous, each step still requires an infinite-dimensional search. In order to tackle this issue, we can discretize the variables and solve the following approximate version.

$$h_i(f_i;\alpha) = \min_{\substack{q_i \in Q_i \\ f_{\text{ch}_1(i)} \in F_{\text{ch}_1(i)} \\ f_{\text{ch}_2(i)} \in F_{\text{ch}_2(i)}}} \quad g_i(q_i;\alpha) + h_{\text{ch}_1(i)}(f_{\text{ch}_1(i)};\alpha) + h_{\text{ch}_2(i)}(f_{\text{ch}_2(i)};\alpha) \tag{5.21a}$$

$$\text{s.t.} \quad |q_i - d_i - f_{\text{ch}_1(i)} - f_{\text{ch}_2(i)} + f_i| \le \epsilon \tag{5.21b}$$

Algorithm 5.5 Find an ϵ-approximate solution to the optimal networked pricing problem (5.16).

Input: G=(V,E), $c_1(\cdot), \ldots, c_n(\cdot), p(\cdot;\cdot), \epsilon$
for α in $\mathcal{A}'$ **do**
  for all nodes i **do**
    compute $g_i(q_i;\alpha)$ **for all** q_i in Q_i, using (5.22)
  end for
  for all nodes $i \neq$root (in bottom-up order) **do**
    compute $h_i(f;\alpha)$ **for all** f in F_i, using (5.21)
  end for
  compute $h_{\text{root}}(0;\alpha)$, using (5.21)
end for
$\alpha^* = \arg\min_{\alpha\in\mathcal{A}'} \; h_{\text{root}}(0;\alpha)$
$f^*_{\text{root}} = 0$
for all nodes i (in top-down order) **do**
  $[q_i^*, f^*_{\text{ch}_1(i)}, f^*_{\text{ch}_2(i)}] = y_i(f_i^*;\alpha^*)$
  $\beta_i^* = b_i(q_i^*;\alpha^*)$
end for
return $(q_1^*, \ldots, q_n^*, f_1^*, \ldots, f_n^*, \alpha^*, \beta_1^*, \ldots, \beta_n^*)$

for all $i = 1, \ldots, n$, where $Q_1, \ldots, Q_n$ and $F_1, \ldots, F_n$ are properly-defined discrete sets (see the e-companion for details). We denote the optimizer (triple) of (5.21) by $y_i(f_i;\alpha)$.

$$g_i(q;\alpha) = \min_{\beta_i\in\mathcal{B}'} \quad p(q;\alpha,\beta_i) \tag{5.22a}$$

$$\text{s.t.} \quad p(q;\alpha,\beta_i) - c_i(q) + \epsilon \geq 0, \tag{5.22b}$$

$$p(q;\alpha,\beta_i) - c_i(q) + \epsilon \geq p(q';\alpha,\beta_i) - c_i(q'), \quad \forall q' \neq q, \tag{5.22c}$$

for all $i = 1, \ldots, n$. The optimizer of (5.22) is denoted by $b_i(q;\alpha)$.

The steps of the procedure are summarized in psuedocode in Algorithm 5.5, and the following result summarizes the theoretical guarantee of the algorithm.

Theorem 5.4.1. *Consider $c_i(\cdot)$ and $p(\cdot;\cdot)$ that have at most a finite number of discontinuities and are Lipschitz on each continuous piece of their domain. Algorithm 5.5 finds an ϵ-approximate solution to the optimal networked pricing problem* (5.16)*, with running time $O\left(n(1/\epsilon)^{l_1+\max\{l_2,1\}+2}\right)$, where n is the number of suppliers, and l_1 and l_2 are the number of shared and individual parameters in the price, respectively.*

It is worth mentioning that the network algorithm developed in this section suggests another way of solving the no-network case as well, by replacing the single market with a line graph with infinite capacities. This algorithm will, in turn, have time complexity $O\left(n(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$, which is the same as that of the one developed in Section 5.3.2.

5.5 Existing Pricing Schemes

In this section, we review the existing pricing schemes in the literature and summarize their properties. No prior pricing rule for general non-convex markets satisfies all the properties discussed in Section 5.2.2. However, it is possible to achieve all the properties in the case when the cost functions are convex via a classical approach: *shadow pricing*. We first briefly illustrate how shadow pricing works for the convex case, and then survey some prominent approaches in the literature that seek to extend the properties of shadow pricing to the non-convex case, contrasting them with the EC scheme.

5.5.1 Pricing in Convex Markets

When the cost functions $c_i(\cdot)$ are convex, a simple and uniform pricing rule, often referred to as *shadow pricing* or *marginal-cost pricing* [142, 143], can achieve all the above-mentioned properties. The pricing scheme works as follows. The operator first solves the convex program

$$\min_{q_1,\dots,q_n} \quad \sum_{i=1}^{n} c_i(q_i) \tag{5.23a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \qquad (\lambda) \tag{5.23b}$$

where λ is the dual variable corresponding to the load-balance constraint. Let $q_1^*, \dots, q_n^*$ and λ^* denote an optimal primal-dual pair of this problem (if there are multiple dual solutions, take λ^* to be the smallest). A payment function of the form

$$p_i(q_i) = \lambda^* q_i \quad i = 1, \dots, n \tag{5.24}$$

satisfies all the properties outlined in Section 5.2.2, and it is relatively straightforward to see that.

For simplicity assume that $c_i(\cdot)$ are differentiable. The optimal solution of (5.23)

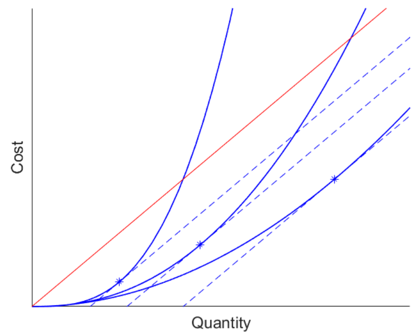


Figure 5.3: An illustration of shadow pricing for the case of 3 convex cost functions. The points indicated by $*$ show the optimal quantities. The 3 functions have the same derivative at their optimal quantities, and the tangent line lies below the function (because of convexity). The red (solid) line that passes through the origin is the uniform price function, which is parallel to the three lines.

satisfies the following (KKT) conditions (which does not require convexity):

$$\begin{cases} \sum_{i=1}^{n} q_i^* = d \\ \frac{dc_i}{dq_i}(q_i^*) = \lambda^*, \quad i = 1, \ldots, n. \end{cases}$$

Next, note that supplier i's profit-maximization problem is

$$\max_{q_i} \quad \lambda^* q_i - c_i(q_i).$$

Since $c_i(\cdot)$ is convex, the objective is concave, and any point at which the derivative is zero, is a global maximizer. In particular, the derivative at q_i^* is zero because of the KKT conditions, and therefore that is a solution to the supplier i's profit-maximization problem. As a result, the scheme supports a competitive equilibrium that clears the market and minimizes the production cost, while using a price form that is simple and uniform. Figure 5.3 illustrates the optimal quantities and the price function for an example with three suppliers.

Note that the total payment of this scheme is $\sum_{i=1}^{n} p_i(q_i^*) = \lambda^* d$, which can be generally higher than $\sum_{i=1}^{n} c_i(q_i^*)$. One can always opt for a *non-uniform* affine price function as $p_i(q_i) = \lambda^* q_i + b_i$, with $b_i = c_i(q_i^*) - \lambda^* q_i^*$, which has lower payments, and makes $\sum_{i=1}^{n} p_i(q_i^*)$ exactly equal to $\sum_{i=1}^{n} c_i(q_i^*)$. However, if one requires a *uniform and linear* price function, it can be shown that $p_i(q_i) = \lambda^* q_i$ has the *lowest total payment* among all such functions.

Table 5.1: A summary of common pricing schemes and their properties. IP: Integer Programming. MU: Minimum Uplift. CH: Convex Hull. SLR: Semi-Lagrangean Relaxation. PD: Primal-Dual. EC: Equilibrium-Constrained.

Scheme \ Property	Price form $p_i(q_i) =$	Proposed for $c_i(q_i) =$	Market Clearing	Revenue Adequate	Supports Competitive Equilibrium	Economically Efficient
Shadow Pricing	λq_i	Convex	✓	✓	✓	✓
IP	$\lambda q_i + u_i \mathbb{1}_{q_i>0}$	Startup+linear	✓	✓	✓	✓
MU/CH	$\lambda q_i + u_i \mathbb{1}_{q_i=q_i^*}$	Startup+convex	✓	✓	✓	×
SLR	λq_i	Startup+linear	✓	✓	×	×
PD	λq_i	Startup+linear	✓	✓	×	×
EC (proposed)	User-specified	General	✓	✓	✓	✓

The results in this table assume solving the formulation for each scheme exactly. However, in practice, these schemes rely on numerical solvers for their problems, and if the problem is non-convex, there is no guarantee of maintaining these properties in general. In particular, the IP scheme requires a non-convex solver. The MU/CH, SLR, and PD schemes, for the cost functions that they are proposed for (i.e., startup+convex or startup+linear), require only convex solvers and therefore satisfy the checked properties exactly. The EC scheme is accompanied by an efficient algorithm for solving the non-convex problem for general cost functions, which satisfies the exact Market Clearing property and the ϵ-approximate versions of the other three properties (see Section 5.3.2).

5.5.2 Pricing in Non-Convex Markets

If the cost functions are non-convex, the approach of shadow pricing, described above, fails. This is because the net profit of each supplier is no longer a concave function, and its stationary points do not necessarily correspond to the maximum. In other words, there may not be a subderivative at q_i^* supporting the cost function $c_i(\cdot)$.

There have been several schemes proposed in the literature that attempt to address this issue and design pricing rules that satisfy the properties discussed above in the context of non-convex cost functions. We review the most promising ones here. Some of the schemes maintain a uniform pricing rule with additional discriminatory side-payments called "uplifts" for incentivizing the suppliers to follow the dispatch, while others raise the uniform price so that it is revenue-adequate. A summary of the pricing schemes, along with their properties, is provided in Table 5.1.

Integer Programming (IP)

A pricing scheme was proposed for non-convex cost functions that are in the form of a fixed (start-up) cost plus a linear marginal cost, sometimes referred to as "IP pricing" [17]. This scheme uses uniform marginal pricing for the commodity and discriminatory pricing for the integral activity of the suppliers. It is based on (i) formulating an optimization similar to (5.23), as a mixed integer linear program (MILP) and solving it for optimal allocations, (ii) reformulating the original MILP as an LP by replacing the integral constraints with forcing commitment choices equal to their optimal values, and (iii) solving the LP problem and using the dual variable λ of Market Clearing constraint as the uniform price and the dual variables $\{u_i^*\}$ of the forced equality constraints as discriminatory uplifts: $p_i(q_i) = \lambda^* q_i + u_i^* \mathbb{1}\{q_i > 0\}$.

IP pricing uses a uniform price plus a discriminatory uplift to clear the market efficiently such that every supplier's net profit is zero. As a result, both total payments and total production costs are minimized at the same time. It is shown that the optimal solutions generated by IP pricing are optimal to the decentralized profit maximization problems for every supplier and thus they support a competitive equilibrium [17]. However, IP pricing assumes knowledge of the optimal solutions to the unit commitment problem and thus is not intended as a practical approach to find the optimal allocation. [18] point out that uniform price generated under IP pricing can be volatile (i.e., a small change in demand could lead to a big change in the uniform price) and uplifts could be generally very large.

Minimum Uplift (MU) / Convex Hull (CH)

To avoid the unwanted properties of IP pricing (i.e., volatility and instability), a pricing scheme, proposed in [18] for the (non-convex) class of startup-plus-convex cost functions, offers minimum uplifts that incentivize each supplier to follow the dispatch rather than maximize their own profits in the absence of uplifts. The scheme is based on solving the mixed-integer program minimizing the total production cost and minimizing total uplifts. Given a fixed uniform price λ, each supplier chooses between following the dispatch to receive the uplifts or not. The uplifts can be viewed as the extra potential profit that the suppliers can make by self-scheduling and maximizing their own profit. Researchers refined the MU pricing and proposed the concept of Convex Hull pricing , which is based on (i) replacing the non-convex cost of the original program with its convex hull to formulate a new LP, (ii) solving the new LP and using the dual variable of Market Clearing constraint as the marginal price and deriving the lost opportunity cost (LOC) as the minimum uplifts to incentivize suppliers' compliance. The final payment $p_i(q_i, z_i)$ as a function of quantity q_i and commitment choice z_i is in the form of a uniform price λ^* and a discriminatory uplift u_i^* as $p_i(q_i) = \lambda^* q_i + u_i^* \mathbb{1}\left\{q_i = q_i^*\right\}$.

Even though MU/CH pricing minimizes total uplifts, the generated marginal price might end up being high, and the payments can be much higher than those of the other schemes. In general, the total payments under this scheme might end up being much higher than the total production costs, which defeats the purpose of minimizing the costs. Even for the class of startup-plus-linear cost functions, where IP pricing is optimal (the total payment is equal to the total production cost, and they are both minimal), MU pricing is not economically efficient, as it fails to minimize the payments.

On the computational side, although the work of [141] proposes a polynomially-solvable primal formulation for the Lagrangian dual problem by explicitly describing the convex hull for piecewise linear or quadratic cost functions, describing the convex hull of cost functions could be very challenging in general and thus makes the problem computationally intractable.

As an aside, MU and CH would not be equivalent if the Market Clearing constraint was an inequality. In that case, the side-payments in CH would be typically larger than those in MU, due to Product Revenue Shortfall [19].

Semi-Lagrangean Relaxation (SLR)

The work of [139] introduced a semi-Lagrangean relaxation approach to find a uniform price that is revenue-adequate at the same solution for quantity and commitment choices as the original optimization problem, for cost functions of startup-plus-linear form. The scheme is based on formulating and solving the SLR of mixed-integer program (MIP) by semi-relaxing the Market Clearing constraint with standard Lagrange multiplier λ. The solution under SLR satisfies the constraints in the original MIP and makes the duality gap between MILP and SLR zero. Though the payment function $p_i(q_i) = \lambda^* q_i$ under SLR pricing is high enough to avoid negative profits for suppliers, it incentivizes the suppliers to deviate and operate at full capacity and total payments usually end up being much higher than total costs of production.

Primal-Dual (PD)

Another revenue-adequate pricing scheme, proposed by [140], exploits a primal-dual approach to derive a uniform price to guarantee that dispatched suppliers are willing to remain in the market (revenue adequacy). The scheme works for cost functions with the form of start-up cost plus linear cost, and the prices have shown not to deviate much from that of [17]. The approach is based on (i) relaxing the integral constraint of the original MILP to formulate a primal LP problem, (ii) deriving the dual LP problem of the primal LP problem, (iii) formulating a new LP problem that seeks to minimize the duality gap between the primal and dual problems subject to both primal and dual constraints and (iv) adding back the integral constraints as well as nonlinear constraints to ensure that no supplier incurs loss and solving the new problem for optimal solutions q_i^*, z_i^* and λ^*.

Though this scheme may be implemented using standard branch-and-cut solvers, it is computationally intractable in general. The prices $p_i(q_i) = \lambda^* q_i$ and profits produced under PD do not significantly deviate from dual prices if integral constraints are relaxed and thus are always bounded. However, as a revenue-adequate pricing scheme, PD fails to form a competitive equilibrium as suppliers are incentivized to operate at full capacity. In general, total payments are much higher than total production costs.

5.6 Experimental Results

In this section, we compare and contrast EC pricing with the existing approaches using numerical experiments on common case studies. Specifically, we compare the payments and uplifts generated from different pricing schemes, including IP, CH, SLR, PD, and EC. Among all these schemes, only EC allows flexibility of the payment form. As a result, we further divide EC into one with a payment function in the form of linear marginal price plus uplifts and another pricing with a payment form of piecewise linear marginal prices plus uplifts. In practice, specific limits on the number of sections and the maximum slope among all sections can be used to further restrict EC. For convenience, we name these variations of EC in terms of number of piecewise sections of its payment form, e.g., EC2 refers to EC with a payment function in the form of 2 piecewise sections plus uplifts.

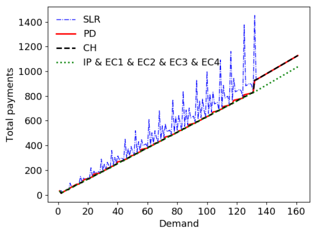
First, we apply all these pricing schemes to a *single* market example from [18], which is a modification of Scarf's example developed in [131]. Second, we adapt cost functions in the modified Scarf's example to be quadratic plus startup cost in order to further explore how these schemes generalize to different cost functions. Finally, we consider a further generalization to a simple 2-node networked market.

5.6.1 Case 1: Linear Plus Startup Cost

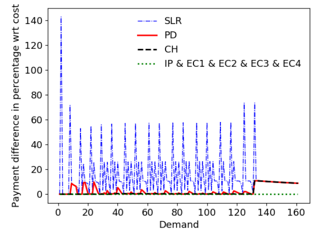
Table 5.2: A summary of the production characteristics in the modified Scarf's example.

Type	Smokestack	High Tech	Med Tech
Capacity	16	7	6
Minimum output	0	0	2
Startup cost	53	30	0
Marginal cost	3	2	7
Quantity	6	5	5

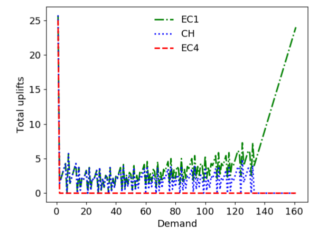
We consider a modified Scarf's example, as proposed in [18]. The parameters are listed in Table 5.2. We assume that demand is inelastic with a maximum capacity of 161 units. We restrict the payment function of EC1, EC2, EC3 and EC4 to, respectively, have one, two, three, and four sections and impose that the marginal price of any section cannot exceed the maximum marginal price for any supplier operating at full capacity. Figure 5.4a shows total payments for different demand levels while Figure 5.4c shows the corresponding uplifts of the pricing schemes that apply, i.e., CH, EC1, EC2, EC3, and EC4. Payments of two revenue-adequate pricing schemes, including SLR and PD, are higher than total costs in general. IP, EC1, EC2, EC3, and EC4 achieve the minimum payments equal to total costs. CH achieves the minimum payments at low demand levels and its total payments surpass total costs as demand gets high. As for uplifts, EC4 achieves the smallest among the five pricing schemes. Total uplifts of CH and EC1 are close to each other at a low demand level and that of EC1 increases significantly when demand approaches capacity. This is not surprising as total payments of CH go above total costs at a high demand, making it possible for relatively smaller total uplifts. It is worth noting that startup prices and marginal prices for IP are volatile and unstable. Figure 5.4d and 5.4e demonstrate that the more complex we allow payment functions of EC family, the smaller total uplifts we can achieve, which means more uniform prices are across suppliers. In practice, there is apparently a trade-off between complexity and uniformity of payment functions among the EC family, and this will be a design choice for the independent system operator (ISO). Overall, EC4 outperforms other pricing schemes in terms of total payments and total uplifts.



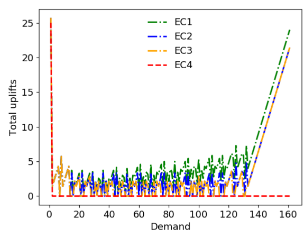
(a) Total payments as a function of demand



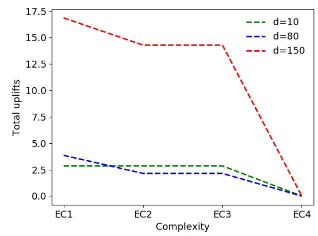
(b) Payment difference in percentage w.r.t cost as a function of demand



(c) Total uplifts as a function of demand



(d) Total uplifts as a function of demand



(e) Total uplifts for pricing schemes at different demand levels

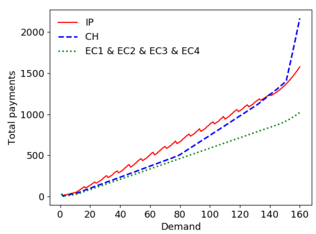
Figure 5.4: An example with cost functions of the form of linear plus startup cost

5.6.2 Case 2: Quadratic Plus Startup Cost

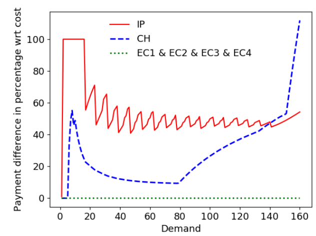
Table 5.3: A summary of the new cost functions in the modified Scarf's example.

Type	Smokestack	High Tech	Med Tech
Cost function	$\frac{3}{16}q^2 + 53 * \mathbb{1}\{q > 0\}$	$\frac{2}{7}q^2 + 30 * \mathbb{1}\{q > 0\}$	$\frac{7}{6}q^2$

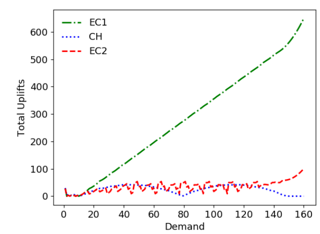
To further explore how these pricing schemes generalize to different cost functions, we modify the cost functions of the example above. Table 5.3 describes the new cost functions for each supplier. Since it is not clear how to generalize SLR and PD, we focus on a comparison among IP, CH, EC1, EC2, EC3, and EC4. We restrict the payment function of EC1, EC2, EC3, and EC4 to, respectively, have one, two, three, and four sections with the marginal price of any section bounded by the maximum of marginal price for any supplier operating at full capacity. As can be seen in Figure 5.5a, EC1 EC2, EC3, and EC4 achieve the possible minimum total payments equal to total costs. Total payments of IP and CH are both above total costs and the gap between total payments and costs grows as demand increases. Observe that the demand here ranges from 1 to 160 because marginal price of CH increases dramatically at the capacity level and the plot over the interval $(1, 160)$ would be a flat line if the whole range were covered. Figure 5.5c shows that total uplifts of EC1



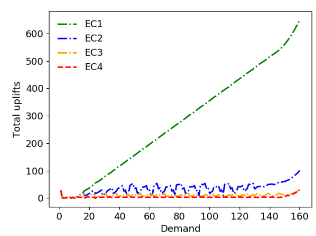
(a) Total payments as a function of demand



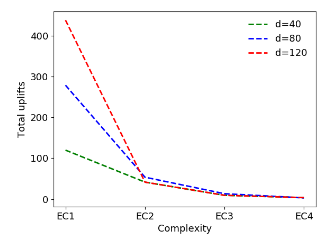
(b) Payment difference in percentage w.r.t cost as a function of demand



(c) Total uplifts as a function of demand



(d) Total uplifts as a function of demand



(e) Total uplifts for pricing schemes at different demand levels

Figure 5.5: An example with cost functions of the form of quadratic plus startup cost.

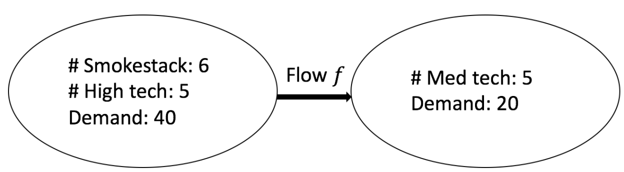
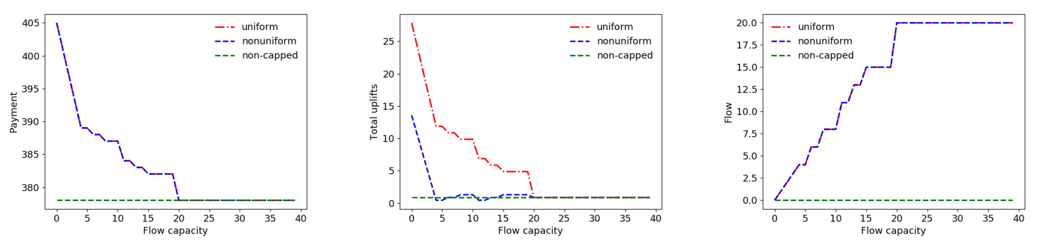


Figure 5.6: A schematic drawing for two connected markets with a constraint on flow capacity.

are much larger than that of CH and EC2. At a low demand level, uplifts of EC1 and EC2 are close to each other. As demand increases, uplifts of EC2 are a little larger than those of CH, in order to maintain a smaller overall payment. There is a trade-off between minimizing total payments and minimizing total costs. Allowing the flexibility of payment function form enables EC2 to perform better than either CH or EC1 in terms of total payments and uplifts. Figure 5.5d and 5.5e show a relationship between complexity of payment function form and magnitude of total uplifts among the EC family pricing schemes. As in the case of cost function being start-up plus linear cost, it is not surprising to see that more complex payment functions tend to allow smaller total uplifts, i.e., more uniform prices across suppliers.

5.6.3 A Networked Market with Capacity Constraints

One advantage EC has over all the other pricing schemes is its generality. Specifically, EC can be applied to networked markets. In this section, we divide a single market with a fixed total demand 60 as described earlier into one market with only med tech suppliers and the other one with the smokestack and high tech suppliers. The cost functions of the suppliers are the same as defined earlier, i.e., linear plus startup cost. As pictured in Figure 5.6, these two markets are connected via a flow capacity constraint. We consider two different cases of non-uniform marginal pricing and uniform marginal pricing for these two markets. Figure 5.7 shows how total payments, total uplifts, and flow between these two connected markets vary as flow capacity increases for nonuniform and uniform marginal pricing settings. The results show that the total payments and total uplifts decrease as more flow is allowed between these two markets until it reaches the demand of one market, which means one market alone meets the total demand. Allowing non-uniform pricing does not further reduce total payments as total payments are minimal and equal the total costs. However, it helps reduce total uplifts, as we can see in Figure 5.7b.



(a) Total payments as a function of flow capacity

(b) Total uplifts as a function of flow capacity

(c) Flow between two markets as a function of flow capacity

Figure 5.7: An example of two connected markets with a constraint on the flow capacity.

5.7 Concluding Remarks

We study the problem of pricing in single and networked markets with non-convex costs. Our key contribution is the proposal of a novel scheme, Equilibrium-Constrained (EC) pricing, which optimizes for the allocations and the price parameters at the same time, while imposing the equilibrium conditions as constraints. Our pricing framework is general in the sense that: (i) it can be used for pricing general non-convex cost functions, (ii) it allows for using general price classes, (iii) can be computed in polynomial-time regardless of the source of the non-convexities, and (iv) it extends easily to networked markets.

This paper opens up a variety of important directions for future work. First, as this framework enables one to use general price classes, it would be interesting to apply it to specific classes of price functions (e.g., quadratic plus uplift, piece-wise, etc.) and characterize the solution theoretically and/or numerically. One can then investigate the potential trade-offs between the complexity of the class and the economic efficiency or the uniformity of the price. Second, since electricity markets are an important application of the pricing problem studied here, it would be interesting to evaluate the proposed scheme in practical settings for electricity markets. Our preliminary exploration shows that we can achieve more efficient (lower total payments) and less discriminatory (lower uplifts) prices with, for instance, piece-wise linear functions. More evaluations in large-scale, practical settings should be carried out in order to evaluate the potential of deployment.

Another important direction to pursue is the extension of our results to networked markets with more general network structures. Our algorithm currently applies to networks with bounded tree-width; however, beyond such networks, new ideas are needed. Finally, our proposed pricing scheme has broader implications for non-convex optimization problems as well. In the convex setting, dual prices are crucial for the development of distributed optimization algorithms, but such approaches have not been possible in non-convex settings due to the lack of pricing rules with the desirable properties laid out in Section 5.2.2. It is now possible to explore whether EC prices can be used as the basis for distributed optimization algorithms in the non-convex setting.

5.A Appendix

5.A.1 Supplement to Section 5.3.1

In this section, we formally prove the reduction of the optimization problem for the class of linear-plus-uplift functions to (5.6), and then show Propositions 5.3.1 and 5.3.2.

Reduction

Here we show that for the class of linear-plus-uplift price functions $p(q_i;\lambda,u_i,\hat{q}_i) = \lambda q_i + u_i \mathbb{1}_{q_i=\hat{q}_i}$, one can assume $\hat{q}_i^* = q_i^*$ without loss of generality, and therefore the optimization problem (5.5) reduces to (5.6) for this class. The optimization problem (5.5) for price function $p(q_i;\lambda,u_i,\hat{q}_i) = \lambda q_i + u_i \mathbb{1}_{q_i=\hat{q}_i}$, $\lambda, u_1, \ldots, u_n \geq 0$, is as follows

$$p^*_{\text{uplift}} = \min_{\substack{q_1,\ldots,q_n \\ \lambda\geq 0 \\ u_1,\ldots,u_n\geq 0 \\ \hat{q}_1,\ldots,\hat{q}_n}} \quad \sum_{i=1}^{n}(\lambda q_i + u_i \mathbb{1}_{q_i=\hat{q}_i}) \tag{25a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d \tag{25b}$$

$$\lambda q_i + u_i \mathbb{1}_{q_i=\hat{q}_i} - c_i(q_i) \geq 0, \quad i = 1, \ldots, n \tag{25c}$$

$$\lambda q_i + u_i \mathbb{1}_{q_i=\hat{q}_i} - c_i(q_i) \geq \max_{q_i'\neq q_i} \lambda q_i' + u_i \mathbb{1}_{q_i'=\hat{q}_i} - c_i(q_i'), \quad i = 1, \ldots, n. \tag{25d}$$

The following lemma shows that this optimization problem can be reduced to (5.6), and the optimal uplifts of (5.6) are no larger than those of (25).

Lemma 5.A.1. *Given any solution* $(\boldsymbol{q^*}, \lambda^*, \boldsymbol{u^*}, \boldsymbol{\hat{q}^*})$ *to the optimization problem* (25)*,* $(\boldsymbol{q^*}, \lambda^*, \underline{\boldsymbol{u}}, \boldsymbol{q^*})$ *is also a solution, where*

$$\underline{u}_i = \begin{cases} u_i^*, & \text{if } \hat{q}_i^* = q_i^* \\ 0, & \text{o.w.} \end{cases}.$$

Proof. Proof of Lemma 5.A.1. Let us first show the feasibility of $(\boldsymbol{q^*}, \lambda^*, \underline{\boldsymbol{u}}, \boldsymbol{q^*})$. For any i such that $\hat{q}_i^* \neq q_i^*$, we have that

$$\lambda^* q_i^* - c_i(q_i^*) \geq 0$$
$$\lambda^* q_i^* - c_i(q_i^*) \geq \max_{q_i'\neq q_i^*} \lambda^* q_i' + u_i^* \mathbb{1}_{q_i'=\hat{q}_i^*} - c_i(q_i') \geq \max_{q_i'\neq q_i^*} \lambda^* q_i' - c_i(q_i'),$$

which implies

$$\lambda^* q_i^* + \underline{u}_i^* \mathbb{1}_{q_i^*=q_i^*} - c_i(q_i^*) \geq 0$$
$$\lambda^* q_i^* + \underline{u}_i^* \mathbb{1}_{q_i^*=q_i^*} - c_i(q_i^*) \geq \max_{q_i'\neq q_i^*} \lambda^* q_i' + \underline{u}_i^* \mathbb{1}_{q_i'=\hat{q}_i^*} - c_i(q_i'),$$

because $\underline{u}_i^* = 0$. Therefore $(\boldsymbol{q^*}, \lambda^*, \underline{\boldsymbol{u}}, \boldsymbol{q^*})$ is feasible.

The objective value of $(\boldsymbol{q^*}, \lambda^*, \underline{\boldsymbol{u}}, \boldsymbol{q^*})$ is

$$\begin{aligned} \sum_{i=1}^{n}(\lambda^* q_i^* + \underline{u}_i) &= \sum_{i:\hat{q}_i^*=q_i^*}(\lambda^* q_i^* + u_i^*) + \sum_{i:\hat{q}_i^*\neq q_i^*} \lambda^* q_i^* \\ &= \sum_{i=1}^{n}(\lambda^* q_i^* + u_i^* \mathbb{1}_{q_i^*=\hat{q}_i^*}), \end{aligned}$$

which is the same as that of $(\boldsymbol{q^*}, \lambda^*, \boldsymbol{u^*}, \boldsymbol{\hat{q}^*})$, and is therefore optimal. □

Based on this lemma, the optimization problem (25) can be reduced to (5.6).

Closed-Form Solutions

Proof. Proof of Proposition 5.3.1. In the optimization problem (5.6), the order of variables in the minimizations does not matter, and further, for every fixed $q_1, \ldots, q_n$ and λ, the minimization over each u_i can be done separately. Therefore this program can be massaged into the following form

$$p^*_{\text{uplift}} = \min_{q_1,\ldots,q_n} \quad \left(\min_{\lambda \geq 0} \sum_{i=1}^{n} g_i(q_i; \lambda) \right) \tag{26a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d, \tag{26b}$$

where

$$g_i(q_i; \lambda) = \min_{u_i \geq 0} \quad \lambda q_i + u_i \tag{27a}$$

$$\text{s.t.} \quad \lambda q_i + u_i - c_i(q_i) \geq 0, \tag{27b}$$

$$\lambda q_i + u_i - c_i(q_i) \geq \max_{q_i' \neq q_i} \lambda q_i' - c_i(q_i'), \tag{27c}$$

for all $i = 1, \ldots, n$. Constraints (27b) and (27c) can be expressed as

$$\lambda q_i + u_i \geq c_i(q_i),$$

$$\lambda q_i + u_i \geq c_i(q_i) + \max_{q_i' \neq q_i} \lambda q_i' - c_i(q_i').$$

It follows that

$$g_i(q_i; \lambda) = \lambda q_i + u_i^* = c_i(q_i) + \max \left\{ 0, \max_{q_i' \neq q_i} \lambda q_i' - c_i(q_i') \right\},$$

which is, of course, a function of λ and q_i. Therefore we have

$$\min_{\lambda \geq 0} \sum_{i=1}^{n} g_i(q_i; \lambda) = \sum_{i=1}^{n} c_i(q_i)$$

and the minimizers λ^* are all values λ for which $\max_{q_i' \neq q_i} \lambda q_i' - c_i(q_i') \leq 0$, which are exactly the elements of $\Lambda = \{\lambda \geq 0 \mid \lambda q \leq c_i(q), \forall q, \forall i\}$ (Figure 5.1 provides a pictorial description of these values). Finally we have the last minimization, which is

$$\min_{q_1,\ldots,q_n} \quad \sum_{i=1}^{n} c_i(q_i) \tag{28a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} q_i = d, \tag{28b}$$

and therefore this minimization problem has $q_i^* = q_i^0\ \forall i$ as its optimizer. We also have $u_i^* = c_i(q_i^*) - \lambda^* q_i^*,\ \forall i$. □

Proof. Proof of Proposition 5.3.2. The steps of the proof are exactly the same as in the previous one, except that the additional minimizer picks the λ with the smallest total uplift $\sum_{i=1}^n u_i(\lambda)$, which corresponds to the largest element of Λ. □

5.A.2 Supplement to Section 5.3.2

In this section, we prove Theorem 5.3.1 in two parts. First, we show that there exist finite sets $Q, \mathcal{A}', \mathcal{B}'$ for which Algorithm 5.4 finds an ϵ-approximate solution, and we quantify the sizes of these sets as a function of ϵ. In the second part, we analyze the running time of Algorithm 5.4.

ϵ-Accuracy

Let us first state a simple but useful lemma.

Lemma 5.A.2 (δ-discretization)**.** *Given a set $C \subseteq [\underline{L_1}, \overline{L_1}] \times \cdots \times [\underline{L_k}, \overline{L_k}]$, for any $\delta > 0$, there exists a finite set C' such that*

$$\forall z \in C,\ \exists z' \in C' \text{ s.t. } \|z - z'\|_\infty \le \delta,$$

and further C' contains at most V/δ^k points, where $V = \prod_{i=1}^k (\overline{L_i} - \underline{L_i})$ is a constant (the volume of the box). C' is said to be a δ-discretization of C.

Let $Q, \mathcal{A}'$, and $\mathcal{B}'$ denote some δ-discretizations of sets $[0, d]$, $\mathcal{A}$ and $\mathcal{B}$, respectively. In other words, for every $q \in [0, d]$, $\alpha \in \mathcal{A}$, and $\beta \in \mathcal{B}$, there exist $q' \in Q$, $\alpha' \in \mathcal{A}'$, and $\beta' \in \mathcal{B}'$, such that $|q - q'| \le \delta$, $\|\alpha - \alpha'\|_\infty \le \delta$, and $\|\beta - \beta'\|_\infty \le \delta$. We can combine all these inequalities as

$$\|(q, \alpha, \beta) - (q', \alpha', \beta')\|_\infty \le \delta.$$

On the other hand, given that the cost function $c_i(\cdot)$ for each i is Lipschitz on each continuous piece of its domain, there exists a positive constant K_i such that $|c_i(q) - c_i(q')| \le K_i |q - q'|$, which implies

$$|c_i(q) - c_i(q')| \le K_i \delta. \tag{29}$$

Similarly, Lipschitz continuity of $p(\cdot;\cdot)$ implies the existence of a positive constant K such that $|p(q, \alpha, \beta) - p(q', \alpha', \beta')| \le K \|(q, \alpha, \beta) - (q', \alpha', \beta')\|_\infty$, which yields

$$|p(q, \alpha, \beta) - p(q', \alpha', \beta')| \le K\delta. \tag{30}$$

Using Eqs. (29),(30), we can see that, for any solution $q_1^*, \ldots, q_n^*, \alpha^*, \beta_1^*, \ldots, \beta_n^*$ to optimization (5.5), there exists a point $q_1, \ldots, q_n, \alpha, \beta_1, \ldots, \beta_n$ with $q_1, \ldots, q_n \in Q$, $\alpha \in \mathcal{A}'$ and $\beta \in \mathcal{B}'$, for which constraints (5.5c) and (5.5d) are violated at most by $(K + K_i)\delta$ and $(2K + 2K_i)\delta$, respectively, and the objective is larger than p^* at most by $nK\delta$. As a result, this point will be an ϵ-approximate solution if

$$(K + K_i)\delta \leq \epsilon \quad \forall i, \tag{31}$$

$$2(K + K_i)\delta \leq \epsilon \quad \forall i, \tag{32}$$

$$nK\delta \leq n\epsilon. \tag{33}$$

These constraints altogether enforce an upper bound on the value of δ as

$$\delta \leq C\epsilon,$$

for some constant C. Therefore if we pick

$$\delta = \frac{d}{\lceil \frac{d}{C\epsilon} \rceil}, \tag{34}$$

our algorithm is guaranteed to encounter an ϵ-approximate solution while enumerating the points, and $Q = \{0, \delta, 2\delta, \ldots, d\}$ is a valid δ-discretization for $[0, d]$, which has $N_q = \left\lceil \frac{d}{C\epsilon} \right\rceil + 1 = O\left(\frac{1}{\epsilon}\right)$ points. The nice thing about this particular choice of δ is that now d can be written as a sum of n elements in Q (because all the elements, including d, are multiples of δ), which allows us to satisfy the Market Clearing condition exactly. Based on Lemma (5.A.2), $\mathcal{A}'$ and $\mathcal{B}'$ contain $N_\alpha = O\left(\frac{1}{\delta^{l_1}}\right) = O\left(\frac{1}{\epsilon^{l_1}}\right)$ and $N_\beta = O\left(\frac{1}{\delta^{l_2}}\right) = O\left(\frac{1}{\epsilon^{l_2}}\right)$ points.

Finally, if there are any discontinuities in the cost or price functions, we can simply add them to our discrete sets Q, $\mathcal{A}'$, and $\mathcal{B}'$, and since there are at most a finite number of them, the sizes of the sets remain in the same order, i.e., $N_q = O\left(\frac{1}{\epsilon}\right)$, $N_\alpha = O\left(\frac{1}{\epsilon^{l_1}}\right)$, and $N_\beta = O\left(\frac{1}{\epsilon^{l_2}}\right)$. Next, we calculate the time complexity of Algorithm 5.4 running on these discrete sets.

Run-Time Analysis

In this section, we show that Algorithm 5.4 has a time complexity of $O\left(n(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$. For every fixed α, we have the following computations

1. The leaves: We need to compute $g_i(q;\alpha)$ for every i and every $q \in Q$. Computing each $g_i(q;\alpha)$ (i.e., for fixed i, q, α) takes $O(N_\beta N_q)$. The reason for that is we have to search over all $\beta_i \in B'$, and for each one, there are $N_q + 1$ constraints to check. More explicitly, we need to (a) check $O(N_\beta N_q)$ constraints, (b) compute N_β objectives, and (c) find the minimum among those N_β values. All these steps together take $O(N_\beta N_q)$, and repeating for every i and q makes it $O(nN_\beta N_q^2)$.

2. The intermediate nodes: In each new level, there are at most half as many (+1) nodes as in the previous level. For each node i in this level, we need to compute $g_i(q;\alpha)$ for every $q \in Q$. For every fixed q, there are $O(N_q)$ possible pairs of (q_j, q_k) that add up to q, and therefore we need to (a) sum $O(N_q)$ pairs of objective values, and (b) find the minimum among them, which take $O(N_q)$. Hence, the computation for each node takes $O(N_q^2)$. There are $O(\frac{n}{2} + \frac{n}{4} + \cdots + 2) = O(n)$ intermediate nodes in total, and therefore the total complexity of this part is $O(nN_q^2)$.

3. The root: Finally at the root, we need to compute $g_{\text{root}}(d;\alpha)$. There are N_q possible pairs of (q_j, q_k) that add up to d. Therefore, we need to compute N_q sums, and find the minimum among the resulting N_q values, which takes $O(N_q)$.

Putting the pieces together, the computation for all values of α takes

$$N_\alpha \times \left(O(nN_\beta N_q^2) + O(nN_q^2) + O(N_q)\right),$$

which in turn is $O(nN_\alpha N_\beta N_q^2)$. Finally, finding the minimum among the N_α values simply takes $O(N_\alpha)$.

The backward procedure, which finds the quantities q_i and the parameters β_i, takes just $O(n)$, since it is just a substitution for every node. As a result, the total running time is $O(nN_\alpha N_\beta N_q^2)$, which, based on the first part (Section 5.A.2), is $O\left(n(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$.

Remark on the ϵ-Approximation

As mentioned at the end of Section 5.3.2, if one requires the total payment in Definition 5.3.1 to be at most ϵ (rather than $n\epsilon$) away from the optimal p^*, the running time of our algorithm will still be polynomial in both n and $1/\epsilon$, i.e.,

$O\left(n^3(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$. To see that, notice in this case (31) and (32) remain the same, and (33) changes to $nK\delta \leq \epsilon$. Therefore, the upper bound enforced by the constraints will be $\delta \leq \frac{C\epsilon}{n}$, for some constant C. In this case, our choice of δ would be $\delta = \frac{d}{\lceil \frac{dn}{C\epsilon} \rceil}$, and hence $N_q = O\left(\frac{n}{\epsilon}\right)$. N_α and N_β remain the same as before. The running time is $O(nN_\alpha N_\beta N_q^2)$, as computed previously, which in this case would be $O\left(n^3(\frac{1}{\epsilon})^{l_1+l_2+2}\right)$.

5.A.3 Supplement to Section 5.4

In this section, we first show the transformation of the problem on a tree to one on a binary tree, and then prove Theorem 5.4.1.

Transformation into Binary Tree

Lemma 5.A.3. *Given any tree with n nodes (suppliers), there exists a binary tree with additional nodes which has the same solution $(q_i^*, \ldots, q_n^*, \alpha^*, \beta_1, \ldots, \beta_n)$ for those nodes as the original network. The binary tree has $O(n)$ nodes.*

Proof. Take any node i that has $k_i > 2$ children. For any two children, introduce a dummy parent node. For any two dummy parent nodes, introduce a new level of dummy parent nodes. Continue this process until there are 2 or less nodes in the uppermost layer, and then connect them to node i (See Fig. 5.8). The capacities of the lines immediately connected to the children are the same as those in the original graph. The capacities of the new lines are infinite.

The total number of introduced dummy nodes by this procedure is

$$O(\frac{k_i}{2} + \frac{k_i}{4} + \cdots + 2) = O(k_i).$$

Since there are $1 + k_1 + k_2 + \cdots + k_n = n$ nodes in total in the original tree, the number of introduced additional nodes is $O(k_1 + \cdots + k_n) = O(n)$. Therefore the total number of nodes in the new (binary) tree is $O(n)$. □

Proof of Theorem 5.4.1

Most of the proof is similar to the one presented in Section 5.A.2. For this reason, we only highlight the main points. The proof consists of ϵ-accuracy and run-time, as before.

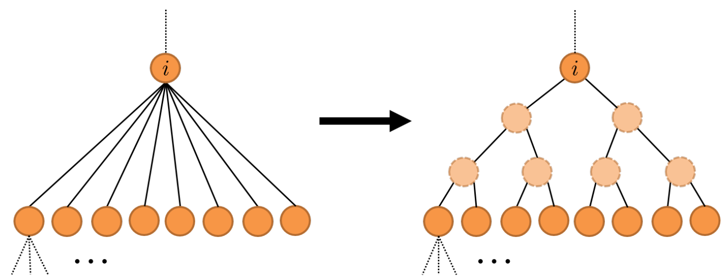


Figure 5.8: The transformation of an arbitrary-degree tree to a binary tree.

ϵ-Accuracy

Let $Q_1, \ldots, Q_n, F_1, \ldots, F_n, \mathcal{A}', \mathcal{B}'$ denote some δ-discretizations of sets $[0, d_1 + \overline{f_{\mathrm{ch}_1(1)}} + \overline{f_{\mathrm{ch}_2(1)}} - \underline{f_1}], \ldots, [0, d_n + \overline{f_{\mathrm{ch}_1(\mathrm{n})}} + \overline{f_{\mathrm{ch}_2(\mathrm{n})}} - \underline{f_n}]$, $[\underline{f_1}, \overline{f_1}], \ldots, [\underline{f_n}, \overline{f_n}]$, $\mathcal{A}$, $\mathcal{B}$, respectively. Note that if any line capacities are infinite, the intervals can be replaced by $[0, \sum_{i=1}^{n} d_i]$ instead. Similarly as in Section 5.A.2, the constraints enforce an upper bound on the value of δ as $\delta \leq C\epsilon$, for some constant C. Based on Lemma (5.A.2), the sizes of the sets will be $N_{q_i} = O\left(\frac{1}{\epsilon}\right)$ $\forall i$, $N_{f_i} = O\left(\frac{1}{\epsilon}\right)$ $\forall i$, $N_\alpha = O\left(\frac{1}{\epsilon^{l_1}}\right)$, and $N_\beta = O\left(\frac{1}{\epsilon^{l_2}}\right)$.

Run-Time Analysis

For every fixed α, the run-time of the required computations is as follows.

1. The time complexity of computing $g_i(q_i; \alpha)$ for each node i and each fixed value of q_i is $O(N_\beta N_{q_i})$. Therefore, computing it for all nodes and all values takes $O(nN_\beta N_q^2)$.

2. Computing $h_i(f_i; \alpha)$ for each node i and each fixed value of f_i takes $O(N_f^2)$, because there are $O(N_f) \times O(N_f)$ pairs of values for $(f_{\mathrm{ch}_1(i)}, f_{\mathrm{ch}_2(i)})$ (q_i is automatically determined as the closest point in Q_i to $d_i + f_{\mathrm{ch}_1(i)} + f_{\mathrm{ch}_2(i)} - f_i$). Therefore, its overall computation for all nodes and all values takes $O(nN_f^3)$.

As a result, the overall computation takes $N_\alpha \times \left(O\left(nN_\beta N_q^2\right) + O\left(nN_f^3\right)\right)$, which is $O\left(n(\frac{1}{\epsilon})^{l_1+l_2+2}\right) + O\left(n(\frac{1}{\epsilon})^{l_1+3}\right)$, or equivalently $O\left(n(\frac{1}{\epsilon})^{l_1+\max\{l_2,1\}+2}\right)$.

BIBLIOGRAPHY

[1] D. Reinsel, J. Rydning, and J. F. Gantz, *Worldwide global datasphere forecast, 2021–2025: The world keeps creating more data — now, what do we do with it all?* Mar. 2021. [Online]. Available: `https://www.idc.com/getdoc.jsp?containerId=US46410421`.

[2] S. Sengupta, S. Basak, P. Saikia, S. Paul, V. Tsalavoutis, F. Atiah, V. Ravi, and A. Peters, "A review of deep learning with special emphasis on architectures, applications and recent trends," *Knowledge-Based Systems*, vol. 194, p. 105 596, 2020.

[3] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," *arXiv preprint arXiv:2007.05558*, 2020.

[4] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "TensorFlow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[5] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," 2017.

[6] D. Chappell, "Introducing Azure machine learning," *A guide for technical professionals, sponsored by Microsoft Corporation*, 2015.

[7] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1709–1720, 2017.

[8] S. Sridharan, K. Vaidyanathan, D. Kalamkar, D. Das, M. E. Smorkalov, M. Shiryaev, D. Mudigere, N. Mellempudi, S. Avancha, B. Kaul, *et al.*, "On scale-out deep learning training for Cloud and HPC," *arXiv preprint arXiv:1801.08030*, 2018.

[9] F. A. Chudak and D. B. Shmoys, "Approximation algorithms for precedence-constrained scheduling problems on parallel machines that run at different speeds," *Journal of Algorithms*, vol. 30, no. 2, pp. 323–343, 1999.

[10] S. Li, "Scheduling to minimize total weighted completion time via time-indexed linear programming relaxations," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, Oct. 2017, pp. 283–294. DOI: `10.1109/FOCS.2017.34`.

[11] S. Davies, J. Kulkarni, T. Rothvoss, J. Tarnawski, and Y. Zhang, "Scheduling with communication delays via LP hierarchies and clustering," *arXiv preprint arXiv:2004.09682*, 2020.

[12] B. Maiti, R. Rajaraman, D. Stalfa, Z. Svitkina, and A. Vijayaraghavan, "Scheduling precedence-constrained jobs on related machines with communication delay," *arXiv preprint arXiv:2004.10776*, 2020.

[13] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," *arXiv preprint arXiv:1906.02243*, 2019.

[14] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," *arXiv preprint arXiv:1907.10597*, 2019.

[15] D. Amodei and D. Hernandez, *AI and compute*, May 2018. [Online]. Available: `https://openai.com/blog/ai-and-compute/`.

[16] A. Wierman, Z. Liu, I. Liu, and H. Mohsenian-Rad, "Opportunities and challenges for data center demand response," in *International Green Computing Conference*, IEEE, 2014, pp. 1–10.

[17] 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, 2005.

[18] W. W. Hogan and B. J. Ring, "On minimum-uplift pricing for electricity markets," *Electricity Policy Group*, 2003.

[19] D. A. Schiro, T. Zheng, F. Zhao, and E. Litvinov, "Convex hull pricing in electricity markets: Formulation, analysis, and implementation challenges," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 4068–4075, 2016.

[20] Y. Su, X. Ren, S. Vardi, A. Wierman, and Y. He, "Communication-aware scheduling of precedence-constrained tasks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 47, no. 2, pp. 21–23, 2019. DOI: `10.1145/3374888.3374897`,

[21] Y. Su, X. Ren, S. Vardi, and A. Wierman, "Communication-aware scheduling of precedence-constrained tasks on related machines," *Under Review*, 2020,

[22] Y. Su, J. Yu, V. Anand, and A. Wierman, "Learning-augmented energy-aware scheduling of precedence-constrained tasks," *Under Review*, 2021,

[23] G. Liao*, Y. Su*, J. Ziani, A. Wierman, and J. Huang, "The privacy paradox and optimal bias-variance trade-offs in data acquisition," in *Proceedings of the 2021 ACM Conference on Economics and Computation*, 2021,

[24] N. Azizan, Y. Su, K. Dvijotham, and A. Wierman, "Optimal pricing in markets with nonconvex costs," *Operations Research*, vol. 68, no. 2, pp. 480–496, 2020. DOI: `10.1287/opre.2019.1900`,

[25] E. G. Coffman and J. L. Bruno, *Computer and job-shop scheduling theory*. John Wiley & Sons, 1976.

[26] R. L. Graham, "Bounds on multiprocessing timing anomalies," *SIAM Journal on Applied Mathematics*, vol. 17, no. 2, pp. 416–429, 1969.

[27] J.-J. Hwang, Y.-C. Chow, F. D. Anger, and C.-Y. Lee, "Scheduling precedence graphs in systems with interprocessor communication times," *SIAM Journal on Computing*, vol. 18, no. 2, pp. 244–257, 1989.

[28] G. Ananthanarayanan, M. C.-C. Hung, X. Ren, I. Stoica, A. Wierman, and M. Yu, "GRASS: Trimming stragglers in approximation analytics," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, Seattle, WA: USENIX Association, 2014, pp. 289–302, ISBN: 978-1-931971-09-6. [Online]. Available: `https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/ananthanarayanan`.

[29] X. Ren, G. Ananthanarayanan, A. Wierman, and M. Yu, "Hopper: Decentralized speculation-aware cluster scheduling at scale," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM '15, London, United Kingdom: ACM, 2015, pp. 379–392, ISBN: 978-1-4503-3542-3. DOI: `10.1145/2785956.2787481`. [Online]. Available: `http://doi.acm.org/10.1145/2785956.2787481`.

[30] M.-Y. Wu and D. D. Gajski, "Hypertool: A programming aid for message-passing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 1, no. 3, pp. 330–343, 1990.

[31] Y. Xu, K. Li, L. He, L. Zhang, and K. Li, "A hybrid chemical reaction optimization scheme for task scheduling on heterogeneous computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3208–3222, 2015.

[32] T. Yang and A. Gerasoulis, "DSC: Scheduling parallel tasks on an unbounded number of processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 9, pp. 951–967, 1994.

[33] H. Topcuoglu, S. Hariri, and M.-y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260–274, 2002.

[34] F. Wu, Q. Wu, and Y. Tan, "Workflow scheduling in cloud: A survey," *The Journal of Supercomputing*, vol. 71, no. 9, pp. 3373–3418, 2015.

[35] R. Mayer, C. Mayer, and L. Laich, "The TensorFlow partitioning and scheduling problem: It's the critical path!" In *Proceedings of the 1st Workshop on Distributed Infrastructures for Deep Learning*, ACM, 2017, pp. 1–6.

[36] G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica, "Effective straggler mitigation: Attack of the clones," in *the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013, pp. 185–198.

[37] G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris, "Reining in the outliers in Map-Reduce clusters using Mantri," in *9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10)*, Vancouver, BC: USENIX Association, 2010. [Online]. Available: `https://www.usenix.org/conference/osdi10/reining-outliers-map-reduce-clusters-using-mantri`.

[38] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, *et al.*, "Apache Hadoop YARN: Yet another resource negotiator," in *Proceedings of the 4th Annual Symposium on Cloud Computing*, ACM, 2013, p. 5.

[39] M. Lin, L. Zhang, A. Wierman, and J. Tan, "Joint optimization of overlapping phases in MapReduce," *Performance Evaluation*, vol. 70, no. 10, pp. 720–735, 2013.

[40] B. Palanisamy, A. Singh, L. Liu, and B. Jain, "Purlieus: Locality-aware resource allocation for MapReduce in a cloud," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, ACM, 2011, p. 58.

[41] J. Tan, X. Meng, and L. Zhang, "Delay tails in MapReduce scheduling," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1, pp. 5–16, 2012.

[42] A. Verma, L. Cherkasova, and R. H. Campbell, "Two sides of a coin: Optimizing the schedule of MapReduce jobs to minimize their makespan and improve cluster performance," in *2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, IEEE, 2012, pp. 11–18.

[43] W. Wang, K. Zhu, L. Ying, J. Tan, and L. Zhang, "Maptask scheduling in MapReduce with data locality: Throughput and heavy-traffic optimality," *IEEE/ACM Transactions on Networking (TON)*, vol. 24, no. 1, pp. 190–203, 2016.

[44] M. Zaharia, A. Konwinski, A. D. Joseph, R. Katz, and I. Stoica, "Improving MapReduce performance in heterogeneous environments," in *Proceedings of the 8th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'08, San Diego, California: USENIX Association, 2008, pp. 29–42. [Online]. Available: `http://dl.acm.org/citation.cfm?id=1855741.1855744`.

[45] S. H. Hashemi, S. A. Jyothi, and R. H. Campbell, "TicTac: Accelerating distributed deep learning with communication scheduling," *CoRR*, vol. abs/1803.03288, 2018. arXiv: `1803.03288`. [Online]. Available: `http://arxiv.org/abs/1803.03288`.

[46] J. K. Lenstra and A. Rinnooy Kan, "Complexity of scheduling under precedence constraints," *Operations Research*, vol. 26, no. 1, pp. 22–35, 1978.

[47] O. Svensson, "Conditional hardness of precedence constrained scheduling on identical machines," in *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, ACM, 2010, pp. 745–754.

[48] N. Bansal and S. Khot, "Optimal long code test with one free bit," in *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pp. 453–462.

[49] L. A. Hall, A. S. Schulz, D. B. Shmoys, and J. Wein, "Scheduling to minimize average completion time: Off-line and on-line approximation algorithms," *Mathematics of Operations Research*, vol. 22, no. 3, pp. 513–544, 1997.

[50] A. Munier, M. Queyranne, and A. S. Schulz, "Approximation bounds for a general class of precedence constrained parallel machine scheduling problems," in *International Conference on Integer Programming and Combinatorial Optimization*, Springer, 1998, pp. 367–382.

[51] M. Queyranne and A. S. Schulz, "Approximation bounds for a general class of precedence constrained parallel machine scheduling problems," *SIAM Journal on Computing*, vol. 35, no. 5, pp. 1241–1253, 2006.

[52] A. Bazzi and A. Norouzi-Fard, "Towards tight lower bounds for scheduling problems," in *Algorithms-Esa 2015*, Springer, 2015, pp. 118–129.

[53] M. Drozdowski, *Scheduling for parallel processing*. Springer, 2009.

[54] Y.-K. Kwok and I. Ahmad, "Static scheduling algorithms for allocating directed task graphs to multiprocessors," *ACM Computing Surveys (CSUR)*, vol. 31, no. 4, pp. 406–471, 1999.

[55] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to NP-completeness*, 1979.

[56] A. Mirhoseini, H. Pham, Q. V. Le, B. Steiner, R. Larsen, Y. Zhou, N. Kumar, M. Norouzi, S. Bengio, and J. Dean, "Device placement optimization with reinforcement learning," in *International Conference on Machine Learning*, PMLR, 2017, pp. 2430–2439.

[57] Y. Gao, L. Chen, and B. Li, "Spotlight: Optimizing device placement for training deep neural networks," in *International Conference on Machine Learning*, PMLR, 2018, pp. 1676–1684.

[58] Y. Huang, Y. Cheng, A. Bapna, O. Firat, M. X. Chen, D. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, *et al.*, "Gpipe: Efficient training of giant neural networks using pipeline parallelism," *arXiv preprint arXiv:1811.06965*, 2018.

[59] S. Davies, J. Kulkarni, T. Rothvoss, J. Tarnawski, and Y. Zhang, "Scheduling with communication delays via LP hierarchies and clustering II: Weighted completion times on related machines," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, SIAM, 2021, pp. 2958–2977.

[60] A. Khosravi, L. L. Andrew, and R. Buyya, “Dynamic VM placement method for minimizing energy and carbon cost in geographically distributed cloud data centers,” *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 183–196, 2017.

[61] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, “Towards the systematic reporting of the energy and carbon footprints of machine learning,” *Journal of Machine Learning Research*, vol. 21, no. 248, pp. 1–43, 2020.

[62] A. Wierman, L. L. Andrew, and A. Tang, “Power-aware speed scaling in processor sharing systems,” in *IEEE INFOCOM 2009*, pp. 2007–2015.

[63] F. Yao, A. Demers, and S. Shenker, “A scheduling model for reduced CPU energy,” in *Proceedings of IEEE 36th Annual Foundations of Computer Science*, IEEE, 1995, pp. 374–382.

[64] S. Albers, F. Müller, and S. Schmelzer, “Speed scaling on parallel processors,” *Algorithmica*, vol. 68, no. 2, pp. 404–425, 2014.

[65] J.-J. Chen, H.-R. Hsu, K.-H. Chuang, C.-L. Yang, A.-C. Pang, and T.-W. Kuo, “Multiprocessor energy-efficient scheduling with task migration considerations,” in *Proceedings. 16th Euromicro Conference on Real-Time Systems, 2004. ECRTS 2004.*, IEEE, 2004, pp. 101–108.

[66] A. Gandhi, M. Harchol-Balter, and M. A. Kozuch, “Are sleep states effective in data centers?” In *2012 International Green Computing Conference (IGCC)*, IEEE, 2012, pp. 1–10.

[67] A. Paya and D. C. Marinescu, “Energy-aware load balancing and application scaling for the cloud ecosystem,” *IEEE Transactions on Cloud Computing*, vol. 5, no. 1, pp. 15–27, 2015.

[68] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, “Energy-efficient resource allocation and provisioning framework for cloud data centers,” *IEEE Transactions on Network and Service Management*, vol. 12, no. 3, pp. 377–391, 2015.

[69] N. Bansal, H.-L. Chan, T.-W. Lam, and L.-K. Lee, “Scheduling for speed bounded processors,” in *International Colloquium on Automata, Languages, and Programming*, Springer, 2008, pp. 409–420.

[70] N. Bansal, H.-L. Chan, K. Pruhs, and D. Katz, “Improved bounds for speed scaling in devices obeying the cube-root rule,” in *International Colloquium on Automata, Languages, and Programming*, Springer, 2009, pp. 144–155.

[71] N. Bansal, D. P. Bunde, H.-L. Chan, and K. Pruhs, “Average rate speed scaling,” in *Latin American Symposium on Theoretical Informatics*, Springer, 2008, pp. 240–251.

[72] R. A. Carrasco, G. Iyengar, and C. Stein, "Resource cost aware scheduling," *European Journal of Operational Research*, vol. 269, no. 2, pp. 621–632, 2018.

[73] D. S. Hochbaum and D. B. Shmoys, "Using dual approximation algorithms for scheduling problems theoretical and practical results," *Journal of the ACM (JACM)*, vol. 34, no. 1, pp. 144–162, 1987.

[74] S. Irani, S. Shukla, and R. Gupta, "Algorithms for power savings," *ACM Transactions on Algorithms (TALG)*, vol. 3, no. 4, 41–es, 2007.

[75] M. Li, B. J. Liu, and F. F. Yao, "Min-energy voltage allocation for tree-structured tasks," *Journal of Combinatorial Optimization*, vol. 11, no. 3, pp. 305–319, 2006.

[76] W.-C. Kwon and T. Kim, "Optimal voltage allocation techniques for dynamically variable voltage processors," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 4, no. 1, pp. 211–230, 2005.

[77] S. Albers and H. Fujiwara, "Energy-efficient algorithms for flow time minimization," *ACM Transactions on Algorithms (TALG)*, vol. 3, no. 4, 49–es, 2007.

[78] N. Bansal, K. Pruhs, and C. Stein, "Speed scaling for weighted flow time," *SIAM Journal on Computing*, vol. 39, no. 4, pp. 1294–1308, 2010.

[79] P. Singh, B. Khan, A. Vidyarthi, H. Haes Alhelou, and P. Siano, "Energy-aware online non-clairvoyant scheduling using speed scaling with arbitrary power function," *Applied Sciences*, vol. 9, no. 7, p. 1467, 2019.

[80] M. Kumar, S. C. Sharma, A. Goel, and S. P. Singh, "A comprehensive survey for scheduling techniques in cloud computing," *Journal of Network and Computer Applications*, vol. 143, pp. 1–33, 2019.

[81] K. Pruhs, R. van Stee, and P. Uthaisombut, "Speed scaling of tasks with precedence constraints," *Theory of Computing Systems*, vol. 43, no. 1, pp. 67–80, 2008.

[82] M. Mezmaz, N. Melab, Y. Kessaci, Y. C. Lee, E.-G. Talbi, A. Y. Zomaya, and D. Tuyttens, "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems," *Journal of Parallel and Distributed Computing*, vol. 71, no. 11, pp. 1497–1508, 2011.

[83] Y. C. Lee and A. Y. Zomaya, "Energy conscious scheduling for distributed computing systems under different operating conditions," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1374–1381, 2010.

[84] S. Wallace, X. Yang, V. Vishwanath, W. E. Allcock, S. Coghlan, M. E. Papka, and Z. Lan, "A data driven scheduling approach for power management on hpc systems," in *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE, 2016, pp. 656–666.

[85] S. Baskiyar and R. Abdel-Kader, "Energy aware DAG scheduling on heterogeneous systems," *Cluster Computing*, vol. 13, no. 4, pp. 373–383, 2010.

[86] L. Wang, G. Von Laszewski, J. Dayal, and F. Wang, "Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS," in *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, IEEE, 2010, pp. 368–377.

[87] V. Shekar and B. Izadi, "Energy aware scheduling for DAG structured applications on heterogeneous and DVS enabled processors," in *International Conference on Green Computing*, IEEE, 2010, pp. 495–502.

[88] J. Mei, K. Li, and K. Li, "Energy-aware task scheduling in heterogeneous computing environments," *Cluster Computing*, vol. 17, no. 2, pp. 537–550, 2014.

[89] M. Mitzenmacher, "Scheduling with predictions and the price of misprediction," *arXiv preprint arXiv:1902.00732*, 2019.

[90] M. Purohit, Z. Svitkina, and R. Kumar, "Improving online algorithms via ml predictions," in *Advances in Neural Information Processing Systems*, 2018, pp. 9661–9670.

[91] D. Rohatgi, "Near-optimal bounds for online caching with machine learned advice," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2020, pp. 1834–1845.

[92] H. J. Smith, T. Dinev, and H. Xu, "Information privacy research: An interdisciplinary review," *MIS Quarterly*, vol. 35, no. 4, pp. 989–1016, 2011.

[93] A. Acquisti, I. Adjerid, R. Balebako, L. Brandimarte, L. F. Cranor, S. Komanduri, P. G. Leon, N. Sadeh, F. Schaub, M. Sleeper, *et al.*, "Nudges for privacy and security: Understanding and assisting users' choices online," *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, pp. 1–41, 2017.

[94] A. Goldfarb and C. Tucker, "Shifts in privacy concerns," *American Economic Review*, vol. 102, no. 3, pp. 349–53, 2012.

[95] M. J. Keith, S. C. Thompson, J. Hale, P. B. Lowry, and C. Greer, "Information disclosure on mobile devices: Re-examining privacy calculus with actual user behavior," *International Journal of Human-Computer Studies*, vol. 71, no. 12, pp. 1163–1173, 2013.

[96] G. S. Kim, S.-B. Park, and J. Oh, "An examination of factors influencing consumer adoption of short message service (SMS)," *Psychology & Marketing*, vol. 25, no. 8, pp. 769–786, 2008.

[97] S. Barth and M. D. De Jong, "The privacy paradox–investigating discrepancies between expressed privacy concerns and actual online behavior–a systematic literature review," *Telematics and Informatics*, vol. 34, no. 7, pp. 1038–1058, 2017.

[98] S. Athey, C. Catalini, and C. Tucker, "The digital privacy paradox: Small money, small costs, small talk," National Bureau of Economic Research, Tech. Rep., 2017.

[99] D. Acemoglu, A. Makhdoumi, A. Malekian, and A. Ozdaglar, "Too much data: Prices and inefficiencies in data markets," National Bureau of Economic Research, Tech. Rep., 2019.

[100] A. Roth and G. Schoenebeck, "Conducting truthful surveys, cheaply," in *Proceedings of the 13th ACM Conference on Electronic Commerce*, 2012, pp. 826–843.

[101] Y. Chen, N. Immorlica, B. Lucier, V. Syrgkanis, and J. Ziani, "Optimal data acquisition for statistical estimation," in *Proceedings of the 2018 ACM Conference on Economics and Computation*, ACM, 2018, pp. 27–44.

[102] Y. Chen and S. Zheng, "Prior-free data acquisition for accurate statistical estimation," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019, pp. 659–677.

[103] J. Abernethy, Y. Chen, C.-J. Ho, and B. Waggoner, "Low-cost learning via active data procurement," in *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 2015, pp. 619–636.

[104] M. Babaioff, R. Kleinberg, and R. Paes Leme, "Optimal mechanisms for selling information," in *Proceedings of the 13th ACM Conference on Electronic Commerce*, 2012, pp. 92–109.

[105] J. Hörner and A. Skrzypacz, "Selling information," *Journal of Political Economy*, vol. 124, no. 6, pp. 1515–1562, 2016.

[106] A. Goldfarb and C. Tucker, "Online display advertising: Targeting and obtrusiveness," *Marketing Science*, vol. 30, no. 3, pp. 389–404, 2011.

[107] D. Bergemann and A. Bonatti, "Selling cookies," *American Economic Journal: Microeconomics*, vol. 7, no. 3, pp. 259–94, 2015.

[108] R. Montes, W. Sand-Zantman, and T. Valletti, "The value of personal information in online markets with endogenous privacy," *Management Science*, vol. 65, no. 3, pp. 1342–1362, 2019.

[109] D. Bergemann and A. Bonatti, "Markets for information: An introduction," *Annual Review of Economics*, vol. 11, pp. 85–107, 2019.

[110] Y. Liu and Y. Chen, "Learning to incentivize: Eliciting effort via output agreement," *arXiv preprint arXiv:1604.04928*, 2016.

[111] Y. Liu and Y. Chen, "Sequential peer prediction: Learning to elicit effort using posted prices," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[112] Y. Liu and Y. Chen, "Surrogate scoring rules and a dominant truth serum for information elicitation," *arXiv preprint arXiv:1802.09158*, 2018.

[113] O. Dekel, F. Fischer, and A. D. Procaccia, "Incentive compatible regression learning," *Journal of Computer and System Sciences*, vol. 76, no. 8, pp. 759–777, 2010.

[114] R. Meir and J. S. Rosenschein, "Strategyproof classification," *ACM SIGecom Exchanges*, vol. 10, no. 3, pp. 21–25, 2011.

[115] R. Meir, A. D. Procaccia, and J. S. Rosenschein, "Algorithms for strategyproof classification," *Artificial Intelligence*, vol. 186, pp. 123–156, 2012.

[116] J. Perote and J. Perote-Pena, "The impossibility of strategy-proof clustering," *Economics Bulletin*, vol. 4, no. 23, pp. 1–9, 2003.

[117] R. Cummings, K. Ligett, A. Roth, Z. S. Wu, and J. Ziani, "Accuracy for sale: Aggregating data with a variance constraint," in *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, 2015, pp. 317–324.

[118] Y. Cai, C. Daskalakis, and C. Papadimitriou, "Optimum statistical estimation with strategic data sources," in *Conference on Learning Theory*, 2015, pp. 280–296.

[119] A. Ghosh and A. Roth, "Selling privacy at auction," in *Proceedings of the 12th ACM Conference on Electronic Commerce*, 2011, pp. 199–208.

[120] L. K. Fleischer and Y.-H. Lyu, "Approximately optimal auctions for selling privacy when costs are correlated with data," in *Proceedings of the 13th ACM Conference on Electronic Commerce*, 2012, pp. 568–585.

[121] A. Ghosh, K. Ligett, A. Roth, and G. Schoenebeck, "Buying private data without verification," in *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, ACM, 2014, pp. 931–948.

[122] K. Nissim, S. Vadhan, and D. Xiao, "Redrawing the boundaries on purchasing data from privacy-sensitive individuals," in *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, 2014, pp. 411–422.

[123] G. Liao, X. Chen, and J. Huang, "Social-aware privacy-preserving correlated data collection," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2018, pp. 11–20.

[124] R. B. Myerson, "Optimal auction design," *Mathematics of Operations Research*, vol. 6, no. 1, pp. 58–73, 1981.

[125] W. Trockel, *Market demand: An analysis of large economies with non-convex preferences*. Springer Science & Business Media, 2012, vol. 223.

[126] D. G. Horvitz and D. J. Thompson, "A generalization of sampling without replacement from a finite universe," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 663–685, 1952.

[127] D. J. Brown, "Equilibrium analysis with non-convex technologies," *Handbook of Mathematical Economics*, vol. 4, pp. 1963–1995, 1991.

[128] R. Guesnerie, "Pareto optimality in non-convex economies," *Econometrica: Journal of the Econometric Society*, pp. 1–29, 1975.

[129] L. A. Wolsey, "Integer programming duality: Price functions and sensitivity analysis," *Mathematical Programming*, vol. 20, no. 1, pp. 173–195, 1981.

[130] H. E. Scarf, "Mathematical programming and economic theory," *Operations Research*, vol. 38, no. 3, pp. 377–385, 1990.

[131] H. E. Scarf, "The allocation of resources in the presence of indivisibilities," *The Journal of Economic Perspectives*, vol. 8, no. 4, pp. 111–128, 1994.

[132] G. Liberopoulos and P. Andrianesis, "Critical review of pricing schemes in markets with non-convex costs," *Operations Research*, vol. 64, no. 1, pp. 17–31, 2016.

[133] N. Azizan Ruhi, K. Dvijotham, N. Chen, and A. Wierman, "Opportunities for price manipulation by aggregators in electricity markets," *IEEE Transactions on Smart Grid*, 2017.

[134] M. Bjørndal and K. Jörnsten, "Equilibrium prices supported by dual price functions in markets with non-convexities," *European Journal of Operational Research*, vol. 190, no. 3, pp. 768–789, 2008.

[135] M. Bjørndal and K. Jörnsten, "A partitioning method that generates interpretable prices for integer programming problems," *Handbook of Power Systems II*, pp. 337–350, 2010.

[136] P. R. Gribik, W. W. Hogan, and S. L. Pope, "Market-clearing electricity prices and energy uplift," 2007.

[137] A. L. Motto and F. D. Galiana, "Equilibrium of auction markets with unit commitment: The need for augmented pricing," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 798–805, 2002.

[138] F. D. Galiana, A. L. Motto, and F. Bouffard, "Reconciling social welfare, agent profits, and consumer payments in electricity pools," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 452–459, 2003.

[139] V. Araoz and K. Jörnsten, "Semi-lagrangean approach for price discovery in markets with non-convexities," *European Journal of Operational Research*, vol. 214, no. 2, pp. 411–417, 2011.

[140] C. Ruiz, A. J. Conejo, and S. A. Gabriel, "Pricing non-convexities in an electricity pool," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1334–1342, 2012.

[141] B. Hua and R. Baldick, "A convex primal formulation for convex hull pricing," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3814–3823, 2017.

[142] R. Turvey, "Marginal cost," *The Economic Journal*, vol. 79, no. 314, pp. 282–299, 1969.

[143] P. Beato and A. Mas-Colell, "On marginal cost pricing with given tax-subsidy rules," *Journal of Economic Theory*, vol. 37, no. 2, pp. 356–365, 1985.