

The Adaptive Charging Network Research Portal: Systems, Tools, and Algorithms

Thesis by
Zachary J. Lee

In Partial Fulfillment of the Requirements for the
Degree of
Electrical Engineering

The logo for the California Institute of Technology (Caltech), featuring the word "Caltech" in a bold, orange, sans-serif font.

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2021
Defended May 18, 2021

© 2021

Zachary J. Lee

ORCID: 0000-0002-5358-2388

All rights reserved

ACKNOWLEDGEMENTS

When I arrived at Caltech almost five years ago, I could not have imagined the intellectual and personal growth I would experience here. Caltech has been an exceptional environment, and I could not have asked for a better place for my graduate studies.

There are many people whom I wish to thank. First, I would like to thank my advisor Steven Low, who has guided me throughout my Ph.D. journey and always made time to mentor me in research and in life. I will forever be thankful for his patience, as we tried to figure out how to turn a parking lot full of electric vehicle chargers into a Ph.D. thesis. His passion for research and for real-world impact is inspiring. He is a model for the type of researcher, engineer, and person I hope to be.

Next, I would like to thank Adam Wierman, who despite his busy schedule as Executive Officer for CMS always made time to chat. While I regret that we have not yet published together, his perspective and insights have been a major influence on this thesis and his advice on crafting great presentations has been invaluable. I also want to thank the other members of my committee, Venkat Chandrasekaran, and Katie Bouman, who have provided excellent advice and perspectives from beyond my immediate field. Venkat's course on Convex Optimization has been one of the most challenging and rewarding courses I have ever taken.

Beyond this committee, I am thankful for Neil Fromer and Stephanie Yanchinski, who have been the source of many great conversations on entrepreneurship and sustainability; Christine Ortega, whose warm smile and administrative wizardry have kept me, our project, and the entire lab on track; Paul Hines, who let me spend a summer in his lab playing with LEGOs which set me on a path toward a Ph.D.; and my professors and mentors at John Brown University, especially Ted Song, Tim Gilmour and Larry Bland, who invested in me academically, personally, and spiritually, and prepared me for the rigors of Caltech.

I would also like to acknowledge the incredible group of collaborators I have worked with over the last five years. In particular, I would like to express my gratitude to my office mates, John Pang and Sunash Sharma. John is a dear friend whose advice I covet and whose friendship made graduate school that much more enjoyable, albeit slightly less productive. Likewise, Sunash Sharma has been a great friend who has

worked closely with me over the last two years to turn the ACN Research Portal into a robust tool that researchers can rely on. I will miss our whiteboard sessions and design reviews.

Large-scale infrastructure projects like the Adaptive Charging Network require a herculean effort and coordination between researchers, administrators, private companies, and funding agencies. I would like to thank the team who created the first ACN: George Lee, Steven Low, Ted Lee, Rand Lee, Daniel Chang, Zhi Low, and Christine Ortega. Without George and Steven's vision, and hard work from the rest of the team, the ACN would not exist today. I am also thankful to Ted Lee, Cheng Jin, and the rest of the PowerFlex team, who have continued to support our goal of turning a production system into a testbed for researchers.

Outside of Caltech, Rachel and I have found a deep community in Pasadena. I am thankful for our church family at Grace Pasadena, which has grounded and encouraged us throughout our time here. We have also been blessed with life-long friends in Josh and Abbey Brake, and Nick and Chelsea Eshnaur.

To my parents, sister, and grandparents, I could never have gotten to this point without your constant love and support, from reading to me as a child to helping me wade through decisions as an adult. Thank you for always believing in me and encouraging me to chase big dreams. You have taught me to work hard, to value learning, and to seek a life of integrity.

And, to my wife Rachel, thank you for sticking by me through the roller coaster of graduate school; I could not have done it without you. From proofreading all of my applications and early papers, being a sounding board for ideas and complaints, and challenging me to think deeper and ask why, you have been a constant source of encouragement and help. I am so thankful that I get to go through the adventures of life with you by my side. I wouldn't have it any other way.

Finally, I am thankful to God for his love and mercy. *Soli Deo Gloria.*

ABSTRACT

Millions of electric vehicles (EVs) will enter service in the next decade, generating gigawatt-hours of additional energy demand. Charging these EVs cleanly, affordably, and without excessive stress on the grid will require advances in charging system design, hardware, monitoring, and control. Collectively, we refer to these advances as smart charging. While researchers have explored smart charging for over a decade, very few smart charging systems have been deployed in practice, leaving a sizeable gap between the research literature and the real world. In particular, we find that research is often based on simplified theoretical models. These simple models make analysis tractable but do not account for the complexities of physical systems. Moreover, researchers often lack the data needed to evaluate the performance of their algorithms on real workloads or apply techniques like machine learning. Even when promising algorithms are developed, they are rarely deployed since field tests can be costly and time-consuming.

The goal of this thesis is to develop systems, tools, and algorithms to bridge these gaps between theory and practice.

First, we describe the architecture of a first-of-its-kind smart charging system we call the Adaptive Charging Network (ACN). Next, we use data and models from the ACN to develop a suite of tools to help researchers. These tools include ACN-Data, a public dataset of over 80,000 charging sessions; ACN-Sim, an open-source simulator based on realistic models; and ACN-Live, a platform for field testing algorithms on the ACN. Finally, we describe the algorithms we have developed using these tools. For example, we propose a practical and robust algorithm based on model predictive control, which can reduce infrastructure requirements by over 75%, increase operator profits by up to 3.4 times, and significantly reduce strain on the electric power grid. Other examples include a pricing scheme that fairly allocates costs to users considering time-of-use tariffs and demand charges and a data-driven approach to optimally size on-site solar generation with smart EV charging systems.

PUBLISHED CONTENT AND CONTRIBUTIONS

- [1] Z. J. Lee, G. Lee, T. Lee, C. Jin, R. Lee, Z. Low, D. Chang, C. Ortega, and S. H. Low, “Adaptive charging networks: A framework for smart electric vehicle charging,” *IEEE Transactions on Smart Grid*, 2021. DOI: 10.1109/TSG.2021.3074437,
Z.J.L participated in the building the software used in the testbed, formulated the constraints for unbalanced three-phase infrastructure, proposed the post-processing algorithm for pilot discretization, proposed case studies, performed the simulations, prepared the results, and participated in the writing of the manuscript.
- [2] Z. J. Lee, J. Z. Pang, and S. H. Low, “Pricing EV charging service with demand charge,” en, *Electric Power Systems Research*, vol. 189, p. 106 694, Dec. 2020, ISSN: 03787796. DOI: 10.1016/j.epsr.2020.106694. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0378779620304971>,
Z.J.L participated in the conception of the project, helped develop the algorithm, performed the simulations, prepared the results, and participated in the writing of the manuscript.
- [3] Z. J. Lee, S. Sharma, D. Johansson, and S. H. Low, “ACN-Sim: An Open-Source Simulator for Data-Driven Electric Vehicle Charging Research,” *arXiv:2012.02809 [cs, eess]*, Dec. 2020, arXiv: 2012.02809. [Online]. Available: <http://arxiv.org/abs/2012.02809>,
Z.J.L participated in the conception of the project, formulated the simulator models, developed the software of the simulator, proposed case studies, performed the simulations, prepared the results, and participated in the writing of the manuscript.
- [4] Z. J. Lee, T. Li, and S. H. Low, “ACN-Data: Analysis and Applications of an Open EV Charging Dataset,” in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, 2019, pp. 139–149, ISBN: 978-1-4503-6671-7. DOI: 10.1145/3307772.3328313. [Online]. Available: <http://doi.acm.org/10.1145/3307772.3328313>,
Z.J.L participated in the conception of the project, developed the software to collect and clean the data, created the APIs and interfaces to distribute the data, helped formulate the proposed algorithms, performed the simulations, prepared the results, and participated in the writing of the manuscript.
- [5] Z. J. Lee, D. Johansson, and S. H. Low, “Acn-sim: An open-source simulator for data-driven electric vehicle charging research,” in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, IEEE, 2019, pp. 1–6. DOI: 10.1109/SmartGridComm.2019.8909765,

Z.J.L participated in the conception of the project, formulated the simulator models, developed the software of the simulator, proposed case studies, ran the simulation, and participated in the writing of the manuscript.

- [6] Z. J. Lee, D. Chang, C. Jin, G. S. Lee, R. Lee, T. Lee, and S. H. Low, "Large-Scale Adaptive Electric Vehicle Charging," in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids*, Oct. 2018, pp. 1–7. DOI: 10.1109/SmartGridComm.2018.8587550,

Z.J.L participated in the building the software used in the testbed, formulated the constraints for unbalanced three-phase infrastructure, proposed case studies, performed the simulations, prepared the results, and participated in the writing of the manuscript.

TABLE OF CONTENTS

Acknowledgements	iii
Abstract	v
Published Content and Contributions	vi
Table of Contents	vii
List of Illustrations	x
List of Tables	xii
I Background	1
Chapter I: Introduction	2
1.1 Challenges to EV Charging Infrastructure	3
1.2 Overcoming these challenges with smart charging	4
1.3 Contributions of this Thesis	11
II Systems & Tools	17
Chapter II: Adaptive Charging Networks	18
2.1 History of the Adaptive Charging Network	18
2.2 Other Smart Charging Systems	20
2.3 System Architecture	21
2.4 Modeling the ACN	26
2.5 Concluding Remarks	31
Chapter III: ACN-Data	33
3.1 Research Enabled by ACN-Data	33
3.2 Other Electric Vehicle Charging Data Sets	35
3.3 Data Records	35
3.4 Data Collection and Processing	38
3.5 Lessons from ACN-Data	45
3.6 Concluding Remarks	52
Chapter IV: ACN-Sim	54
4.1 Existing Tools and Simulators	55
4.2 Simulator Architecture and Models	56
4.3 Charging Algorithms	67
4.4 Simplified Research Pipeline	69
4.5 Concluding Remarks	70
Chapter V: ACN-Live	72
5.1 Design Goals	73
5.2 System Architecture	75
5.3 Using ACN-Live	78

5.4 Concluding Remarks	79
III Algorithms & Applications	80
Chapter VI: Adaptive Scheduling Algorithms	81
6.1 Online Scheduling Framework	82
6.2 Evaluating ASA Against Baseline Algorithms	88
6.3 Load Flattening to Reduce Grid Impacts	100
6.4 Concluding Remarks	103
Chapter VII: Pricing EV Charging Services	106
7.1 Demand Charge	107
7.2 Pricing rule	108
7.3 Online scheduling	117
7.4 Simulations	118
7.5 Concluding Remarks	123
Chapter VIII: Data-Driven Modeling	125
8.1 Modeling Charging Workloads	125
8.2 Predicting User Behavior	128
8.3 Evaluating Charging System Designs with Data	133
8.4 Optimal Sizing of On-site Solar for Workplace Charging	136
8.5 Smoothing the Duck Curve	142
8.6 Concluding Remarks	144
IV Impact and Future Directions	147
Chapter IX: Conclusions	148
9.1 The Research Gap	148
9.2 Tools to Bridge the Gap	149
9.3 Key Lessons	150
9.4 Suggestions for System Designers	153
9.5 Suggestions for Public Policy	154
9.6 Future Directions	154
Bibliography	157

LIST OF ILLUSTRATIONS

<i>Number</i>	<i>Page</i>
1.1 Cartoon example of charging profiles.	5
1.2 Cartoon examples of conventional vs. smart charging.	6
1.3 Overview of the ACN Research Portal.	11
1.4 The ACN smart EV charging testbed at Caltech.	12
2.1 Architecture of the ACN.	22
2.2 System topology for the California Parking Garage.	25
2.3 Circuit diagram for the California Parking Garage.	27
2.4 Line currents from uncontrolled charging.	27
3.1 Photos for the N Wilson Garage ACN	40
3.2 Data collection pipeline for the ACN.	43
3.3 Data processing pipeline for the ACN.	44
3.4 Session records in ACN-Data by month.	46
3.5 Histogram of arrival and departure times for free and paid charging.	46
3.6 Histogram of arrivals and departure times on weekdays before and after COVID-19.	47
3.7 Histogram of arrivals and departure times on weekdays vs. weekends.	49
3.8 Empirical complementary cumulative distribution of session laxity at Caltech and JPL using data from May 1, 2018 through March 1, 2020.	50
3.9 Sessions per month on the Caltech campus in 2019.	50
3.10 Fallback of the actual pilot to 8 A when communication times out.	51
4.1 Architecture of ACN-Sim along with related sub-modules.	56
4.2 Flow chart describing the simulator's run() function.	58
4.3 Comparison of Linear2Stage and idealized Battery models with real charging curves.	63
4.4 Flowchart describing the common research pipeline used when eval- uating an algorithm through simulation.	69
5.1 Architecture of ACN-Live.	75
5.2 ACN-Live swim lanes diagram.	76
6.1 Histogram of arrival and departure times at the California Garage 01.	89

6.2	Aggregate power draw and line-currents for single-phase and three-phase ASA-QC.	93
6.3	Comparison of percentage of energy delivered as a function of transformer capacity for single-phase and three-phase systems.	94
6.4	Comparison of percentage of energy delivered as a function of transformer capacity with realistic models.	96
6.5	Operator profit, costs, and revenue for various scheduling approaches when using SCE's EV TOU-4 tariff.	98
6.6	Comparison of charging profiles for one EV on September 13, 2018 with a 70 kW transformer capacity.	99
6.7	240-node distribution feeder from the Midwest United States.	101
6.8	Comparing the effect of charging scenarios on net power draw at Bus 2053 and minimum system voltage.	104
7.1	Comparison of energy cost and demand charge when optimizing for each in isolation and for total cost.	107
7.2	Comparison of total costs and revenues in the offline case.	120
7.3	Comparison of total costs and revenues in the online case.	122
7.4	Distribution of price per session and blended price per user.	123
8.1	Qualitative evaluation of Gaussian Mixture Model fit.	128
8.2	Prediction errors for session duration and energy delivered using GMM.	130
8.3	Correlation between prediction errors in duration and energy.	132
8.4	Savings achieved installing properly sized solar for each month of the year.	140
8.5	Effect of solar levelized cost of energy and site on optimal solar capacity.	141
8.6	Net demand curves after optimal smoothing.	144
8.7	15 minute maximum ramping rates and peak demand relative to the baseline without EVs.	145

LIST OF TABLES

<i>Number</i>	<i>Page</i>
3.1 Open EV Charging Data Sets.	35
3.2 Filenames for Session records.	36
3.3 Schema for Session Records.	37
3.4 Schema for User Inputs.	38
3.5 Schema for EVSE State Time Series.	38
3.6 Schema for Control Time Series.	38
3.7 Schema for Aggregate Power Time Series.	38
3.8 Schema for Station Information Records.	39
3.9 Description of charging clusters.	41
3.10 EVSEs at each charging cluster.	41
3.11 EVSE types.	42
6.1 Average Statistics for EV Charging Test Cases Per Day May 1, 2018 - Oct. 1, 2018	89
6.2 SCE EV TOU-4 Rate Schedule for EV Charging	91
6.3 Modeling Assumptions by Scenario	95
7.1 Offline Operator Surplus by Month	121
7.2 Online Operator Surplus by Month	121
7.3 Energy surcharges to ensure non-negative surplus (\$/kWh)	122
8.1 SMAPEs for Caltech and JPL datasets.	132
8.2 Infrastructure Solution Evaluation (100 EV / Day)	135
8.3 Infrastructure Solution Evaluation (200 EV / Day)	136
8.4 Evaluating Planned Solar Capacity for Caltech	139

Part I

Background

Chapter 1

INTRODUCTION

The first electric vehicles date back to the late 19th century when inventors and innovators around the world began using electric motors and batteries to create horseless carriages. By 1900, roughly one-third of all motor vehicles on the road were powered by electricity[1], including over 850 electric taxicabs operated by the Electric Vehicle Company in New York, Boston, Washington, and Chicago[2]. However, electric vehicles then suffered from the same challenges that persist today, such as, high initial costs, lower range, and lack of charging infrastructure. Higher initial costs pushed many consumers toward Henry Ford's Model T, which sold for 75% less than a comparable electric coupe [3]. At the same time, as rural roads improved in the 1920s, the range and ease of refueling gasoline vehicles led to the virtual disappearance of EVs in America by the mid-1930s [4].

Environmental regulations of the 1990s pushed automakers to reconsider electric vehicles [5], [6]. Perhaps the best-known electric vehicle of this period was General Motor's EV1 [7], [8]. Unfortunately, GM discontinued the EV1 in 2001. By 2008, Tesla released its Roadster, capturing the world's imagination with what an electric vehicle could be and shattering the stereotype of EVs as glorified golf carts. This spurred offerings from other automakers, including the Nissan LEAF and Chevy Volt (a plug-in hybrid) in 2010.

At the time of this writing, electric vehicle sales are rapidly increasing worldwide. This growth has been driven by advances in battery technologies, increasing consciousness around environmental impacts, and new policies and regulations that incentivize companies and individuals to go electric. In 2019, global EV sales topped 2.1 million, a 40% year-over-year increase, which brought the total number of EVs on the road to 7.2 million [9]. By 2030, the number of EVs is expected to reach 140 million, generating over 550 TWh of additional electricity demands [9].

As EVs themselves improve and costs decrease, the infrastructure around them can become the bottleneck to adoption. In a 2020 survey of consumer sentiment, Deloitte found that lack of charging infrastructure was the greatest concern for prospective EV drivers in the United States, United Kingdom, and Italy, eclipsing driving range and higher upfront costs [10]. This underscores the importance of

charging infrastructure in the transportation electrification movement. This is the same problem that plagued early EVs in the 1920s and '30s and eventually led to their demise.

To alleviate these concerns and meet EV charging demands will require significant investments in infrastructure. In their 2020 Global EV Outlook report, the IEA estimated the world will need over 135 million level-2 EVSEs at homes and workplaces [9]. They also point to the need for over 10 million public level-2 EVSEs and 1 million public DC Fast Chargers [9].

In this thesis, our goal is to develop practical systems and algorithms to support the widespread deployment of electric vehicle charging infrastructure at minimal cost. To support this goal, we have also developed tools and data sets that enable us to approach these problems in a data-driven way. We have publicly released these tools and data sets so that other researchers can do the same.

1.1 Challenges to EV Charging Infrastructure

Today, approximately 80% of EV charging occurs overnight at single-family residences [11]. This is made possible by the high correlation between homeownership and EV ownership, both of which are tied to higher incomes [12]. For homeowners, the convenience of simply plugging in a vehicle at night and leaving fully charged in the morning is one attractive benefit of driving an EV. However, not all EV drivers will live in single-family residences. In fact, only 61% of Americans live in detached, single-family homes [13]. This means that nearly 40% of Americans will need access to charging in multi-unit dwellings, workplaces, or public charging facilities.

Smart charging is especially crucial for large-scale charging facilities such as workplaces, apartment complexes, shopping centers, airports, and fleet charging facilities. Providing charging at these diverse sites is vital to the widespread adoption of electric vehicles. Doing so can reduce range anxiety and provide an alternative to personal charging ports for those who cannot install them at their homes. Since many of these sites will provide charging during daytime hours, they can use abundant solar energy production and enable EVs to provide grid services throughout the day.

However, with current technology, most sites cannot install more than a few charging ports due to limited infrastructure capacity and fear of high electricity bills.

Smart charging allows sites to scale their port capacity without costly infrastructure upgrades. Consider that most EV charging ports installed today deliver about 6.7 kW of power (32A/208V), which is roughly equal to the peak power draw of an average California home [14]. At the same time, the average EV requires about 10 kWh per day [15], which is approximately 50% of the total energy demands of a California home [16]. This means that charging a parking lot's worth of electric vehicles using conventional systems could require the same amount of power as an entire neighborhood.

Even for sites with enough local electrical capacity, adding this much load could negatively affect the health of the grid, leading to voltage issues and overloading transformers and power lines [17], [18]. These problems are only exacerbated when multiple charging systems are on the same distribution circuit, which is likely to happen as entire office parks or blocks or apartments install charging infrastructure. The traditional approach to alleviating these grid issues would be to upgrade the transformers and wires feeding this area. However, this is a costly and time-consuming approach with the potential to significantly delay charging infrastructure deployment and raise costs for all ratepayers [19].

Finally, even if the infrastructure can be upgraded to support EV charging, operating costs are a concern for many site hosts. Peak electricity prices can be extremely high. For example, Southern California Edison's TOU EV-8 tariff has a peak price is \$0.55 / kWh, 3.7 times higher than its off-peak price [20]. Moreover, some utilities have demand charges based on the maximum power draw during a billing period. While SCE and other investor-owned utilities in California have moved away from demand charges for electric vehicle charging sites, many utilities still have significant demand charges. For example, Pasadena Water and Power currently charges \$16.09 / kW [21]. Where demand charge is used, it can often make up over 50% of an EV charging site's electricity bill [22].

1.2 Overcoming these challenges with smart charging

These challenges are real but not insurmountable. While EVs are massive loads, they are also very flexible. Many EV batteries can store upwards of 60 kWh, which would take over 8.5 hours to charge from empty at 6.7 kW. However, most users only need about 10 kWh a day or about 1.5 hours at 6.7 kW [15]. This is far less time than the average car spends idle, whether at a home, workplace, or shopping center. We refer to the difference between the time necessary to charge at maximum power

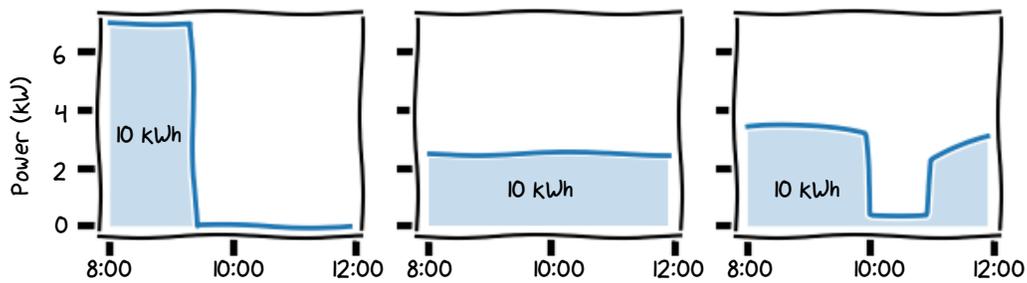


Figure 1.1: Three charging profiles to deliver 10 kWh of energy to Erin's EV before noon.

vs. the time a car is left parked as the EV's *laxity*. Laxity has long been used in deadline scheduling systems to quantify flexibility and prioritize tasks. The higher an EV's laxity, the more flexibility it offers for scheduling. Within a workplace environment, drivers have a median laxity of five hours, which provides plenty of flexibility in scheduling [15]. We can use this flexibility to reduce the need to install costly infrastructure and/or minimize operating costs.

Example 1. *To better understand EV charging flexibility, we consider the case of Erin. Erin arrives at work at 8 am and needs 10 kWh of energy to charge her EV fully. Erin has a lunch meeting across town, so she will need her fully charged EV by noon. Even with this shorter workday, Erin's EV still has significant flexibility. Fig. 1.1 shows three different charging profiles we could use to fully charge Erin's EV by her deadline. These are only a few examples of a continuum of feasible profiles we could choose.*

Scheduling has long been used to allocate scarce resources. On a personal level, many people use a prioritized to-do list to allocate the scarce resource of their time. In a more technical example, computer servers use scheduling approaches to prioritize which requests to process based on their resource needs, deadline, and priority. In our case, the scarce resource is the capacity of the electrical infrastructure. Our task is to schedule when and how fast each EV charges in order to maximize an objective while respecting the limitations of our infrastructure and EVs. With proper scheduling, we can deliver more energy without needing to increase the overall power capacity of the system.

Example 2. *To see how scheduling can work for more drivers, let's consider the simple case of five EVs and an aggregate power limit of 12.5 kW. We consider two cases, shown in Fig. 1.2. With conventional charging, this load would be enough*

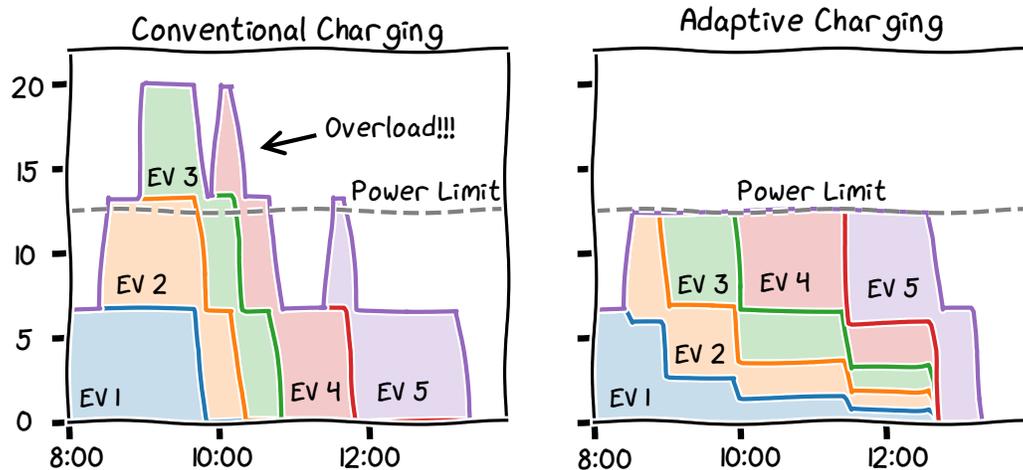


Figure 1.2: Smart charging can stay below power limits while still meeting everyone's energy needs.

to overload the system, but with load management, we can spread the charging throughout the day and stay below the power limit.

Scheduling Methods for Smart Charging

There is a vast literature of algorithms for smart EV charging, which are outlined in [23] and [24]. Many approaches are borrowed from the deadline scheduling community, which has long considered similar problems in multi-processor scheduling. These approaches include processor sharing and index-based algorithms like first-come-first-served and earliest-deadline-first. While these traditional scheduling algorithms can be applied to smart charging, they are not tailored to the EV charging problem. In particular, these approaches are only concerned with finishing jobs, while, in many cases, we would like to achieve other objectives such as load flattening, cost minimization, or grid support. Researchers have turned to approaches like model predictive control, dynamic programming, and reinforcement learning to meet these objectives.

Equal Sharing.

One simple way to ensure that load never exceeds the system's power capacity would be to divide the capacity by the number of EVs. In the deadline scheduling literature, this is known as processor sharing. In fact, this is a common approach in many charging systems today [25]. However, while this approach is in some sense "fair," it does not account for differences between user demands, meaning that some users

may finish charging long before their deadline, while others may be left unsatisfied.

Index Based Policies.

Another common class of scheduling algorithms are indexed-based policies. In these policies, jobs are first sorted by a given index, then processed in that order. The primary objective of these index policies is to maximize throughput. However, secondary objectives like fairness have also been evaluated [26]. Perhaps the most natural index-based policy is *First-Come First-Served* (FCFS). This policy prioritizes jobs by the order in which they arrive. Other scheduling approaches use additional information. *Earliest Deadline First* (EDF) orders users by their departure time, while *Least Laxity First* (LLF) prioritizes users with less flexibility using the laxity metric we discussed previously. These approaches are common in the deadline scheduling literature [27]. However, the EDF algorithm does not account for job size, which means it can fail to schedule large jobs with enough time to finish them. In contrast, the LLF algorithm correctly prioritizes jobs by their relative flexibility but can cause rapid fluctuations in charging (see Fig. 6.6). To combat this, Nakahira et al. propose a smoothed least laxity first algorithm [28]. Other traditional scheduling approaches, like *Least Remaining Processing Time* (LRPT), prioritize sessions with energy demands which can be completed quickly.

In [29], Y. Xu, Pan, and Tong propose to combine LLF and LRPT to form the Less Laxity and Longer remaining Processing time (LLLP) index¹. Under some mild assumptions, they prove that any algorithm that does not follow the LLLP ordering could be improved by prioritizing a session with less laxity and longer remaining processing time. Here *improved* means resulting in a lower non-completion penalty². However, this work makes the key assumptions that all vehicles are interchangeable at a given time, i.e. if an algorithm assigns $r_i(k) = 1$ and $r_j(k) = 0$, then $r_i(k) = 0$ and $r_j(k) = 1$ is also feasible. This assumption does not hold in the ACN, which considers unbalanced, hierarchical constraints.

In [26], Zeballos, Ferragut, and Paganini propose a new index-based scheduling approach called the least-laxity ratio (LLR), which achieves a notion of proportional fairness when not all energy demands can be met. They compare this LLR approach to EDF, LLF, processor sharing. The authors find that to maximize en-

¹Under LLLP, if two vehicles have the same laxity, the vehicle with a longer remaining processing time is prioritized.

²This non-completion penalty is assumed to be a non-negative, non-decreasing convex function.

ergy throughput, the choice of index policy is secondary, assuming the algorithm fully utilizes the infrastructure capacity, so they instead focus on fairness measures. The argument behind this throughput observation is that either the system meets all energy demands, so the amount of energy delivered is fixed by the demands, or the capacity of the infrastructure caps the amount of energy delivered. However, as we see in Chapter 6, this observation does not hold in our practical systems. First, it relies on the assumption that the system has a constraint stream of incoming work. If we instead consider the practical case where there is a finite amount of work to be done, then algorithms must properly allocate resources to do that work. Otherwise, jobs might reach their deadline without being fully satisfied, and potential throughput is lost. In addition, when considering unbalanced three-phase systems, an efficient algorithm must account for unbalanced to fully utilize infrastructure capacity, which naive index policies fail to do. This requires us to develop more advanced algorithms, like ASA in Chapter 6, even for the relatively simple task of maximizing throughput.

Load Flattening.

Another common application of smart EV charging is to reduce peaks in electricity demand, also known as load flattening. This objective can help reduce demand charges and stress on the grid. For example, in [30], Gan, Topcu, and Low propose a decentralized algorithm for coordinating EV charging to minimize load variance (or equivalently any strictly convex function in the aggregate rates). They prove that in the offline case their algorithm converges to the optimal solution. In the online case, the algorithm is used to solve the inner optimization in a model predictive control approach. Simulations show promising results for this online version. In another work by Tang and Y. J. Zhang, [31], the authors propose a different model predictive control approach for load flattening. In this approach, the authors first use MPC to find an optimal aggregate charging profile. They then propose using an earliest-deadline-first algorithm to assign rates to individual EVs. They use the optimal aggregate profile as a power limit for each time interval. One limitation of this work is that it does not account for individual EVs' power limits in the aggregate MPC problem, so the aggregate profile may not be feasible. In another work, G. Zhang, Tan, and G. G. Wang propose a water filling algorithm based on model predictive control [32]. A unique feature of their approach is a time-varying lower bound on each EVs' charging rate, which allows them to reduce the optimization horizon and thus problem size. The authors evaluate this algorithm using real data

from a large retail venue, where they achieve a 20-25 kW reduction in peak demand.

Lowering cost of electricity.

Many utilities offer electricity prices which vary with time. A popular approach is to use smart charging algorithms to schedule EV charging during low-cost periods, reducing operating costs. In [33], B. Wang et al. propose an MPC approach with predictions of user behavior to reduce operating costs. They find that their approach can reduce operating costs by nearly 30% compared to a simple equal sharing algorithm. They also consider the case of on-site solar, which results in a 66.7% reduction in cost. Likewise, in [34], L. Zhang and Li propose an approximate dynamic programming approach to minimize costs in the face of uncertainty around arrival patterns and prices.

In [35], Gou et al. propose a two-stage approach to maximizing operator profits consider time-of-use tariffs, on-site solar energy production, parking fees, and fixed costs. In the first stage, they use stochastic programming to set fixed prices (\$/kWh) for the next day. In the second stage, an MPC approach is used to minimize costs, using a stochastic model for solar production to account for uncertainty. At a broader level, in [36], Z. Xu et al. propose a three-stage algorithm for minimizing costs for chargers across multiple cities. In the first stage, they assign a target aggregate charging profile for each city for the next day. Then, in real-time, a city-level optimization is run to track the given profile while also achieving secondary objectives like minimizing costs and charging quickly. The output of this second stage is a reference curve for each charging site. Finally, at the site level, a variant of least-laxity-first is used to track the given reference curve.

In addition to time-varying energy prices, most commercial customers pay a demand charge based on their peak electricity use throughout the month. These charges can be significant, up to 50% of a site's total electricity costs.

By limiting the peak power draw and scheduling charging to fill in valleys in the background demand of a site, we can significantly reduce these demand charges. Since demand charge can account for up to 50% of a site's electricity costs, the savings can be significant.

Using green electricity.

EVs are only as green as the electricity that fuels them. To lessen the environmental impact of charging, smart charging algorithms can shift charging toward times of high solar and wind production. This can be especially important for sites with on-site solar, where EVs can modulate demand depending on solar output. In [33] and [35] the authors consider how this on-site generation can also reduce operating costs. While in [37], Gan et al. propose an approach similar to load flattening to follow fluctuating renewable generation.

Reducing grid impacts.

As we have discussed, electric vehicles are large loads that could significantly impact the grid. These impacts can be reduced using load flattening and responding to time-of-use prices. In addition, smart charging systems can respond to demand response (DR) signals from the utility. During a DR event, the utility will request participating loads to reduce their power consumption to balance supply and demand. This can prevent blackouts and reduce the need to overbuild infrastructure for worst-case scenarios.

EV charging could also be directly scheduled based on grid constraints. In [38], de Hoog et al. propose an optimal power flow (OPF) based algorithm for controlling a network of residential chargers while respecting transformer ratings, line ratings, and voltage constraints.

Limitations of existing algorithms.

Despite this vast literature, very few smart charging approaches have been deployed in practice. This is largely because of the gap between the theoretical models used in research and the practical constraints we find in real systems. While these simpler models can make analysis tractable, they can also lead to a sizable gap between theoretical results and robust, high-performance implementations of algorithms.

For example, most of the above works assume that EV charging can be modeled as a continuous variable. However, we find in practice that most EVSEs only support a finite number of set-points (called pilot signals). In addition, EVs can deviate from their set-point, especially as their battery nears 100% state-of-charge. Accounting for these hardware limitations and non-ideal battery behaviors require us to rethink our scheduling approaches. In addition, to enable over-subscription

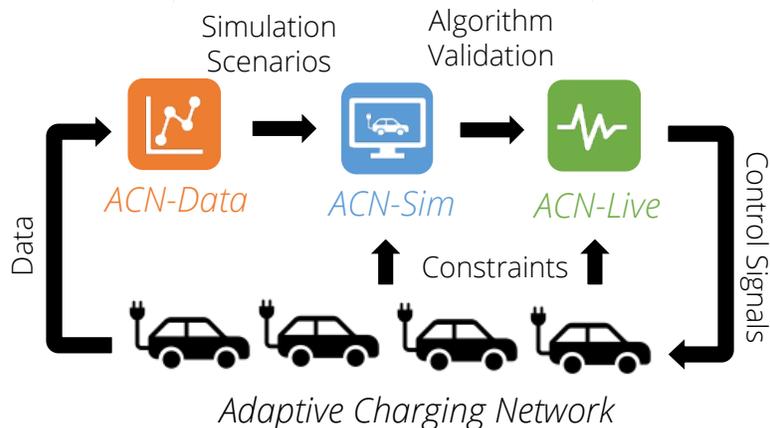


Figure 1.3: The ACN Research Portal gives users many of the benefits of an EV charging testbed without needing to build one themselves. It includes data collected from real charging sessions (ACN-Data), a simulator to evaluate new ideas (ACN-Sim), and access to run on real hardware (ACN-Live).

of infrastructure, we need to account for electrical constraints beyond simple peak limits. These infrastructure constraints are complicated by the unbalanced, three-phase electrical systems and delta-connected loads common in commercial charging systems.

In addition, the lack of openly accessible data has prevented researchers from applying the latest in data-driven methods, including machine learning approaches and trace-driven simulations. Even when algorithms could be applied in practice, researchers rarely have the infrastructure available to deploy their ideas, so promising approaches are left in papers only.

1.3 Contributions of this Thesis

The primary contributions of this thesis are the development of infrastructure, tools, and algorithms to bridge the gap between theory and practice in the smart EV charging field. To do this, we have developed the Adaptive Charging Networks framework for smart EV charging systems and deployed ACNs at Caltech and around the country. Using these systems, we collected data, developed an open-source simulator, and designed a system for field testing new algorithms. Using these tools, we can develop new algorithms for control, pricing, and forecasting. We can also answer questions about the impact of these algorithms on infrastructure design and the grid.



Figure 1.4: The ACN smart EV charging testbed at Caltech.

Adaptive Charging Networks – Chapter 2

Given the urgency in which EV charging infrastructure needs to be deployed and the clear benefits of smart charging approaches, we have sought to bridge this gap between research and practice. To do so, we first identified the need for a robust platform for smart charging. Such a platform would allow us to collect data and deploy algorithms in a real system. To meet this need, we developed a smart charging system that we call the Adaptive Charging Network (ACN)

The first ACN was installed in a parking garage on the Caltech campus in early 2016, with 54 custom EVSEs. This installation has now grown to over 144 level-2 EVSEs and six DCFCs on the Caltech campus. The technology has also spawned a company, PowerFlex Systems, which, as of April 2021, operates over 4,500 EVSEs at over 200 sites.

The ACN is a cyber-physical system that consists of five interacting subsystems: (1) the information system which is responsible for collecting information and computing control actions; (2) the sensor system which gathers information from the physical system; (3) the actuation system (made up of EVSEs and the EVs' battery management systems) which controls each vehicle's charging rate; (4) the physical system (electrical infrastructure) which delivers power to the EVs and other loads within the system; and (5) drivers who provide data to the system and decide when their vehicles are available to charge.

ACN Research Portal

Beyond serving as a model for smart charging systems, the ACN has led to the creation of the ACN Research Portal. This portal has three parts: ACN-Data, a collection of real fine-grained charging data collected from the Caltech ACN and similar sites [15]; ACN-Sim, an open-source simulator that uses ACN-Data and realistic models derived from actual ACNs to provide researchers with an environment to evaluate their algorithms and test assumptions [39]; and ACN-Live, a framework for safely field testing algorithms directly on the Caltech ACN.

ACN-Data – Chapter 3. ACN-Data is an open dataset of over 80,000 real EV charging sessions. The dataset is collected from seven charging sites at Caltech, NASA’s Jet Propulsion Laboratory (JPL), and an office building in Northern California. These sites provide various use-cases, including campus/public use charging and access-controlled workplace charging. The smallest of these sites has only four level-2 chargers, while the largest has 78 level-2 chargers and two DC Fast Chargers (DCFCs) in a single parking structure. In total, it contains data from 204 Level-2 EVSEs and six 50 kW DC Fast Chargers. Each session includes arrival time, departure time, and energy delivered, as well as user inputs and vehicle information. ACN-Data also includes high-fidelity time series of charging current, control signals, and voltage.

ACN-Sim – Chapter 4. ACN-Sim is an open-source, high-fidelity, data-driven EV charging simulation environment. This simulator includes realistic models of charging stations, batteries, and electrical infrastructure, as well as the ability to play back real scenarios from ACN-Data or generate events from statistical models.

This simulator provides researchers who may lack access to real EV charging systems with a realistic environment to evaluate their algorithms and test their assumptions. It also provides a common platform on which algorithms can be evaluated head-to-head using scenarios from ACN-Data. This allows researchers to better understand and articulate how their work fits into the existing literature.

ACN-Live – Chapter 5. ACN-Live is a framework to allow field testing of online algorithms using the Caltech ACN. By utilizing the same algorithm interface as ACN-Sim, researchers can thoroughly test their algorithms in ACN-Sim and then

deploy them to the real testbed with *no code changes*. These field tests help to bridge the gap between theory and practice.

Practical Applications of These Tools

Using the ACN and ACN Research Portal, we have addressed practical questions relating to large-scale EV charging systems and propose new algorithms and methods for real-time scheduling, pricing charging services, and data-driven modeling.

Adaptive Scheduling Algorithm – Chapter 6.

To schedule EV charging so that it does not overload the system's power limit, we propose the Adaptive Scheduling Algorithm (ASA) framework, based on model-predictive control and convex optimization. We consider a discrete-time model. At each time interval, we solve a convex optimization problem to arrive at a set of charging rates for each EV in the system. This algorithm takes, as inputs, the energy requested by the user and their estimated departure time, both of which are provided via a mobile app.

Within the ASA framework, we propose a modular objective function that can be customized to meet the needs of a particular site host. For example, some hosts would like to charge all vehicles as quickly as possible. Others would like to minimize their costs or environmental impact. Still, others may want to maximize their utilization of on-site generation or minimize their impact on the distribution grid. By providing a flexible framework, we make it easy for site hosts to select from this menu of options.

One of the most challenging parts of scheduling charging in a real system is that the algorithm does not know what the future holds. As new EVs arrive, or when users need to change their input parameters, the algorithm must adapt its schedule accordingly. ASA also incorporates many realistic constraints such as unbalanced three-phase infrastructure, control signal quantization, and battery management system behavior.

Because it handles this uncertainty and these practical constraints, ASA can be deployed on real charging systems like the ACN. ASA has been actively controlling EV charging systems since 2017 and has been responsible for safely distributing over 5 GWh of electricity as of January 2021. With adaptive scheduling, we have

shown that the charging demands of most office charging systems can be met with 75% lower infrastructure capacity than conventional systems.

Pricing EV Charging Services – Chapter 7.

The dual problem (in a mathematically precise sense) to the power allocation problem solved by ASA is the problem of fairly dividing costs amongst users. While most charging systems today are either free or charge a flat price per kWh or minute, we propose a dynamic pricing scheme that accounts for both time-of-use energy tariffs and demand charges. Our method optimizes the trade-off between inexpensive time-of-use pricing and peak power draw. We then assign a per session energy price that precisely captures the costs of energy, demand charges, and infrastructure congestion for which that session is responsible. The method provably ensures revenue adequacy in the offline case. Moreover, simulations show that even in the online case, when ASA with a cost minimization objective is used, we approximate the offline optimal within 10%. We find this results in negligible averages losses (<\$18 / month) using our method.

Data-Driven Modeling – Chapter 8.

With access to real data, we can apply data-driven methods to model and forecast charging demands. In particular, we have focused on the problems of predicting use departure times and energy demands using Gaussian Mixture Models trained on data from ACN-Data. We find that these predictions are more accurate than the estimates provided by users via our mobile app in many cases.

The ACN Research Portal has also been useful for improving and evaluating the design of smart charging systems. Using ACN-Sim and models learned from ACN-Data, we can quantify the benefits of smart charging over traditional uncontrolled charging systems easily. This is a useful tool for convincing site hosts, policymakers, and other stakeholders of the value of smart charging systems. We also demonstrate a novel method for designing optimally sizing on-site solar for use with large-scale charging systems. This method uses stochastic optimization paired with statistical models learned from ACN-Data.

Finally, using the ACN Research Portal, we can better answer important questions about the impact that smart EV charging will have on the grid. We demonstrate that by coordinating the charging of millions of workplace chargers, we can minimize

the ramping rate challenges associated with California's demand "duck curve."

Part II

Systems & Tools

Chapter 2

ADAPTIVE CHARGING NETWORKS

To enable smart charging at scale, we first needed a platform that would allow us to take measurements from the physical system, compute control actions, and apply those actions back to the physical system, closing the loop. Such systems are referred to as *cyber-physical systems*. When we began our work on smart electric vehicle charging, these systems did not exist for the scale of smart charging we envisioned, so we decided to build our own. Doing so provided us the platform we needed to run experiments and gather data. It also yielded insights into the practical challenges of smart EV charging systems. This chapter will expound on the history of the Adaptive Charging Network (ACN), describe its architecture, and show how the insights we gain from the ACN inform the models we will use in the rest of this thesis.

2.1 History of the Adaptive Charging Network

In early 2015, Steven Low, and a former student, George Lee¹, began discussing possible technology transfer pathways for the smart grid research being done in Low's lab. G. L. and S. L. had previously partnered together at FastSoft, another Caltech startup. Low's lab had been researching optimal power flow and control of distributed energy resources (DERs). However, both knew that a business model that was reliant on utility partnerships would be difficult.

One day, when the two of them were having lunch at the Caltech Athenaeum, G. L. brought along a prototype for a smart EVSE he had built based on the OpenEVSE project. G. L. had been an early adopter of EVs, and Low's lab had done early work in control algorithms for charging electric vehicles, so this was a perfect fit. They envisioned a system that would use algorithms developed in Low's lab to enable sites to host far more charging ports than they could with conventional systems. In 2015, this had never been done in practice, especially at the scale that they envisioned. By providing value directly to the site host, this plan would allow the young company to scale, independent of utility company buy-in. They hoped that once they had proven

¹No relation to the author.

the idea, they could then leverage their controllable assets to provide services to the grid.

To get started, they needed to prove that the idea would work in practice. Early proposals were met with skepticism, but they forged ahead. With early funding from the Caltech Innovation Initiative (now the Rosenberg Innovation Initiative) and the Caltech Rocket Fund, they installed the first 50 smart EVSEs in the California Parking Garage on campus. This in itself was a herculean feat. After working with John Onderdonk and the Caltech facilities team to identify the site for the testbed and the necessary electrical work, bids for the project came back far above their budget. Undeterred, G. L. completed the designs and permitting himself and enlisted a family friend and contractor to help build the project with the available funds. At this point, Ted Lee and Rand Lee, along with Daniel Chang and Christine Ortega, joined the project to build the hardware, develop software, and oversee the project. The first ACN went live in February 2016.

The initial system provided a valuable proof of concept, which unlocked additional funding from the National Science Foundation and the Wells Fargo Innovation Incubator program. As the idea grew, it became clear that building custom EVSEs would not be scalable, so half of the initial 50 custom EVSEs were replaced with commercially available chargers from AeroVironment and ClipperCreek. The second half were replaced about a year later. To control these chargers, they added wireless communication chips and installed an industrial computer in the garage to run control software.

In parallel with hardware development, S. L. began working with Zhi Low to develop the algorithms to control these chargers. After realizing that existing approaches in deadline scheduling and DER control were insufficient for their problem, they eventually settled on a model predictive control approach. This was a forerunner of the Adaptive Scheduling Algorithm, which we describe in Chapter 6. Over the summer of 2016, Karl Eriksson, a Summer Undergraduate Research Fellow from ETH Zurich, used data from a large technology company to demonstrate the potential of this system to reduce infrastructure requirements significantly.

By the time I came to Caltech in August 2016, the first ACN had been operating for over 6 months. After my arrival, one of our first projects was to demonstrate using the ACN to align charging loads with solar generation. Over the next year, we continued to refine the software and algorithms behind the ACN. During this time, we transitioned our control software from C++ to Python to enable faster

development cycles. We also enhanced our algorithms to account for the issues we found along the way, including hardware limitations, limited information, and unbalanced three-phase infrastructure constraints. We will discuss these limitations in Section 2.4.

In 2017, PowerFlex Systems was officially incorporated, and Cheng Jin, a former postdoc in Low's lab and FastSoft co-founder, joined as VP of Engineering. Over the next two years, PowerFlex deployed several more ACNs, launched a mobile app, and continued to mature its technology. In late 2019, PowerFlex was acquired by EDF Renewables. With the backing of a multi-national energy company, ACN deployment accelerated further. Today, PowerFlex controls over 4,500 EVSEs at over 200 sites. During this time, the ACN system has grown from managing only EV charging to an integrated energy management system that controls EV charging, stationary storage, and on-site solar generation.

2.2 Other Smart Charging Systems

Several smart EV charging systems have been developed, though usually at a smaller scale than the ACN.

Smart Energy Plaza.

The Smart Energy Plaza (SEP) at Argonne National Laboratory consists of six level-2 EVSEs controlled via a direct serial connection to the EVSE board or the Open Charge Point Protocol (OCPP) [40], [41]. Like the ACN, SEP is designed to use oversubscribed infrastructure, but on a smaller scale.

WinSmartEV.

Likewise, the WinSmartEV system at UCLA consists of quad-port level-2 EVSEs capable of sharing a single oversubscribed circuit and multiple level-1 chargers with binary control [25], [42]. A key difference between the ACN and WinSmartEV is that the ACN allows for a more general constraint set based on local electrical infrastructure. Meanwhile, WinSmartEV considers congestion only on the line feeding the quad-port EVSE.

My Electric Avenue.

The My Electric Avenue (MEA) trial tested the Esprit smart charging system that controlled 200 Nissan LEAFs with a mixture of residential and workplace charging. Esprit consists of a controller at the substation and connected level-2 EVSEs that can be cycled when lines approach their thermal limits [43]–[45]. While the ACN targets oversubscribing behind-the-meter electrical infrastructure, MEA was primarily concerned with limits on the distribution system.

Parker.

The Parker project utilized a testbed of 10 bi-directional EVSEs at a commercial site to investigate the potential of EVs to provide frequency regulation services and adapt to marginal emissions signals [46]. The Parker project was primarily interested in vehicle-to-grid (V2G) applications while the ACN considers uni-directional managed charging. While V2G is a promising technology, it is not yet deployed at scale due to its higher cost and uncertainty around business models to support it.

TRADE EVs.

In [47], Frendo, Gaertner, and Stuckenschmidt describe the development of an open-source implementation of the algorithms proposed in [48]. To test the algorithm, they conducted a one-year trial with eight commuters using the TRADE EVs testbed. This testbed uses the OCPP protocol for monitoring and control. Unlike the ACN, which uses a mobile app for authentication and gathering user inputs, TRADE EVs uses RFID cards to authenticate users and does not collect user inputs. Like the ACN, this testbed is oversubscribed. Eight 32 A three-phase level-2 EVSEs (22 kW each) share a 50 A fuse (34.5 kW). This results in a 5X oversubscription ratio, which is higher than the 3.4X ratio for some parts of the Caltech ACN. Another key difference is that the TRADE EVs testbed uses a European style, three-phase EVSE, while the ACN uses an American style single-phase EVSE. This means that the ACN must deal with phase unbalanced, while European systems can generally assume balanced operation.

2.3 System Architecture

The architecture of the ACN is shown in Fig. 2.1. The ACN is a cyber-physical system that consists of five interacting subsystems:

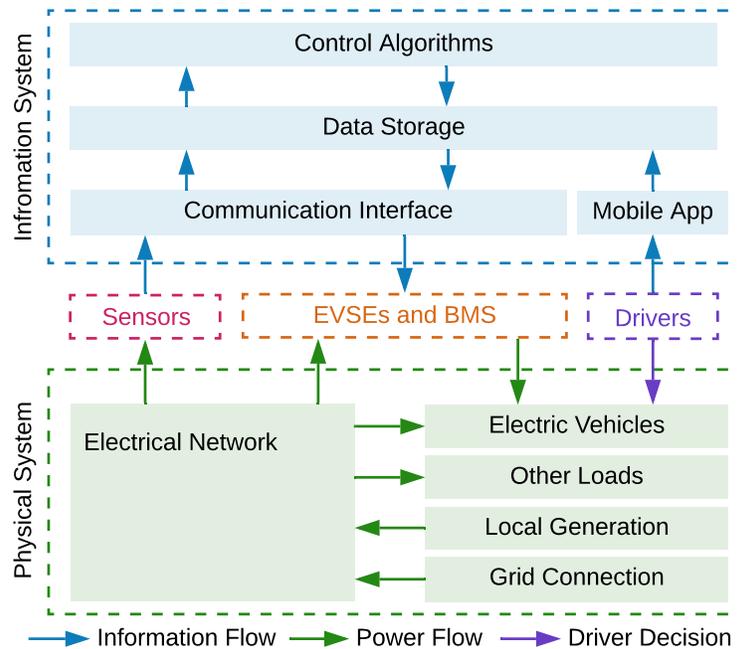


Figure 2.1: Architecture of the ACN. Blue and green arrows signify the flow of information and power, respectively. Sensors measure power flowing in the electrical network and convert this into information. Likewise, EVSEs and the EVs' onboard battery management system (BMS) work together as actuators to control the flow of power into each EV's battery based on signals from the information system. Drivers provide information to the system via a mobile app and directly control when EVs are plugged in or unplugged from the system (signified by the purple arrow).

1. **Information System** - collects and stores data from the system, makes predictions, and computes control actions
2. **Sensor System** - gathers information from the physical system
3. **Actuation System** - EVSEs and the EVs' battery management systems which control each vehicles charging rate
4. **Physical System** - electrical infrastructure which delivered power to the EVs and other loads in the system
5. **Drivers** - provide data to the system and decide when their vehicles are available to charge

Information System.

ACN's information system collects and stores relevant data and computes control actions. It consists of four components:

Communication interface. The communication interface collects sensor data and passes it to the data storage layer. It also passes signals generated by the control algorithms to the corresponding EVSEs. An industrial computer within the parking garage controls this communication interface. The computer connects to the cloud-based components through a cellular internet connection and uses a Zigbee-based mesh network to communicate with sensors and EVSEs.

Data storage. The ACN utilizes a relational database to store information such as site configurations, driver profiles, and charging session parameters. A dedicated time-series database stores measurements, like the voltage, current, and power readings, taken from sensors in the electrical network. The data storage layer allows us to create visualizations for drivers and site operators, which helps them understand the state of the system and their own EV's charging trajectory in real-time, as seen at caltech.powerflex.com.

Mobile app. Our mobile app collects data directly from drivers. After setting up an account, a driver scans a QR code on the EVSE and then provides an estimated departure time and requested energy. If the driver's plans change, they can update these parameters throughout the charging session. The app also allows the site to collect payment and, if desired, implement access control. To ensure that drivers provide information through the app, an EV will only charge at 8 amps until the driver claims the session. After 15 minutes, if the session is not claimed, it will be terminated, and the EVSE will cease charging.

Control algorithms. The control layer takes inputs from the data layer and calculates a charging schedule for each EV in the system. We use an event-based system to trigger the scheduling updates. The events considered include a vehicle plugging in or unplugging, a driver changing request parameters, or a demand response signal from the utility. A publish-subscribe model handles events. Whenever an event occurs or the time since the last charging schedule update exceeds a threshold (for example, 5 min), we compute a new charging schedule. These periodic computations close the control loop and account for discrepancies between the control signal sent to each EV and its actual current draw. We describe this model predictive control framework in detail in Chapter 6.

EVSEs and Battery Management System (Actuation System)

To control charging rates, we use the pilot signal mechanism defined by the J1772 standard for level-2 EVSEs [49]. According to this standard, the EVSE can communicate an upper bound to the EV's battery management system (BMS) that limits the amount of current it may draw from the EVSE. Because it is only an upper bound, the vehicle's BMS may charge at a lower rate. This can occur for various reasons, such as the pilot signal being higher than the vehicle's maximum charging rate or the BMS limiting current draw as the battery reaches a high state of charge. It can be difficult to diagnose why a car is charging below its allocated pilot signal since the J1772 standard does not provide a way to gather the EV's state of charge. Also, most EVSEs on the market today, including the ClipperCreek, AeroVironment, Webasto, and Tesla EVSEs in the Caltech ACN, only support a finite set of pilot signal values and require quantization of the control signal.

Sensor System

Sensors provide a bridge between the physical system and the information system. These sensors measure power, current, and voltage within the local electrical network, allowing us to monitor the system state and accurately track energy usage. The sensors also provide feedback for the control algorithm.

Physical System

The physical system of the ACN includes the local electrical network (including transformers, lines, breakers, loads, and local generation), a connection to the grid, and the electric vehicles. Fig. 2.2 shows the topology of the local electrical network for one garage of the Caltech ACN. Power is delivered to the garage from the distribution transformer via three-phase service at $480 V_{LL}$.

From there, power is distributed throughout the garage via the main switch panel. The ACN is connected to this panel by two 150 kVA delta-wye transformers t_1 and t_2 , which step the voltage down to $120 V_{LN}$. Each level-2 EVSE is a single-phase load connected line-to-line ($208 V_{LL}$) with a maximum current draw between 32 A to 80 A depending on its type. Because of unequal loading between phases, which is unavoidable due to the stochastic nature of driver demands on the system, we cannot assume balanced operation. This makes protection of transformers t_1 and t_2 challenging which we discuss in Section 2.4. Another interesting feature of the

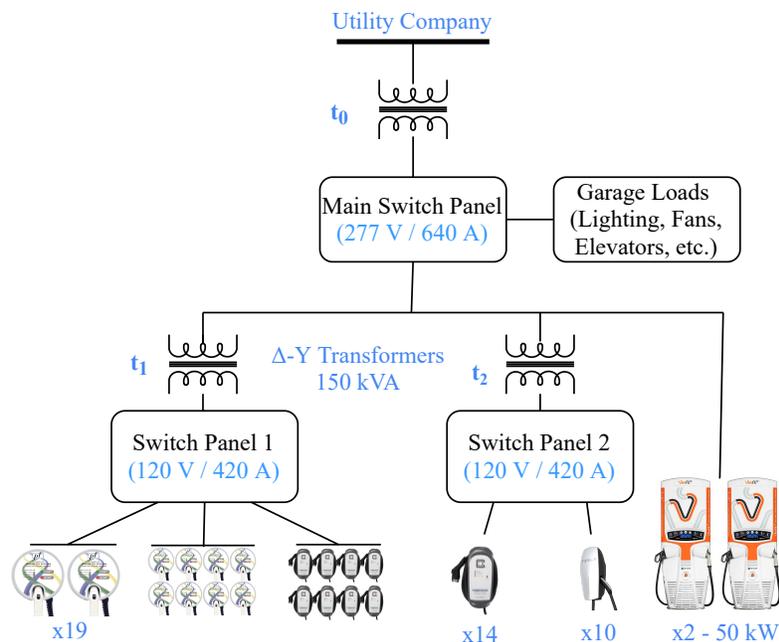


Figure 2.2: System topology for the California Parking Garage in the Caltech ACN. The system consists of 78 EVSEs and two 50 kW DC Fast Chargers. Switch Panel 1 is fed by a 150 kVA transformer and feeds 54 6.7 kW EVSEs, leading to a 2.4X over-subscription ratio. Nineteen of these lines feed pairs of two 6.7 kW AeroVironment EVSEs. Two additional 80 A lines feed pods of eight 6.7 kW EVSEs each, one pod of AeroVironment stations and the other of Clipper Creek stations. Switch Panel 2 is fed by an identical 150 kVA transformer and feeds 14 13.3 kW Clipper Creek EVSEs and 10 16.6 kW Tesla EVSEs. Each of these EVSEs has a dedicated 80 A line. All EVSEs in the system are connected line-to-line at 208 V. The two 50kW DC Fast Chargers from Tritium are balanced 3-phase loads connected directly to the main switch panel. We do not directly control the DCFCs at this time.

Caltech ACN is the two pods of eight EVSEs. These pods are each fed by an 80 amp line. Since each EVSE in the pod has a maximum charging rate of 32 A, these lines are oversubscribed by 3.2 times. This demonstrates how smart charging can allow sites to scale EVSE capacity with existing infrastructure.

In addition to the 78 EVSEs in the garage, the ACN also includes two 50 kW DC fast chargers (DCFCs). These DCFCs are balanced three-phase loads. While this garage does not have local generation, other garages in the Caltech ACN and other PowerFlex sites have on-site solar generation.

Drivers

Human behavior can add significant randomness to the system. Drivers may arrive, depart, or change their input parameters at any time. Drivers are also difficult to model. Input through the mobile app can be highly inaccurate, as shown in [15].

To combat this, in Chapter 8 we use machine learning techniques to predict driver parameters better than the drivers themselves.

2.4 Modeling the ACN

By building and operating the Caltech ACN, we have identified several important features of the physical system that have not been addressed in the EV charging literature but pose real problems for implementing practical EV scheduling algorithms. Among these are proper modeling of the unbalanced three-phase electrical network, incorporating EVSE quantization, and adaptation to non-ideal battery behavior. We describe these models in this section and explain how we incorporate them into an MPC framework in Chapter 6. These models also form the basis of the component models included in ACN-Sim, which we will describe in Chapter 4.

Notation

We use a discrete time model, with time indexed by k in $\mathcal{K} := \{1, 2, 3, \dots\}$. The length of each time period is δ e.g. 5 minutes. At time k , $\hat{\mathcal{V}}_k$ is the set of all EVs present at the ACN and $\mathcal{V}_k \subseteq \hat{\mathcal{V}}_k$ is the subset of *active* EVs, i.e. the set of EVs whose energy demands have not been met. The state of EV $i \in \mathcal{V}_k$ at time k is described by a tuple $(e_i(k), d_i(k), \bar{r}_i(k))$ where $e_i(k)$ is the remaining energy demand of the EV at the beginning of the period, $d_i(k)$ is the remaining duration of the session, and $\bar{r}_i(k)$ is the maximum charging rates for EV i . We denote the charging rate of each EV at time, k , $r_i(k)$. In addition, we define $\hat{e}(k)$ to be the measured energy delivered to the EV over time interval k . For simplicity of notation, we express $r_i(k)$ in amps and $e_i(k)$ and $\hat{e}_i(k)$ in $\delta \times$ amps, assuming nominal voltage.

Infrastructure modeling

The EVSEs in Caltech ACN are connected in a delta configuration as shown in Fig. 2.3. Because of differences in demand, the loads in this delta configuration are often imbalanced, requiring us to carefully consider the infrastructure constraints to ensure safe operation. In our data, we observe that these phase imbalances can be significant without proper control, as seen in Fig. 2.4. While many algorithms have been proposed to handle charging with an aggregate power limit or even a hierarchy of limits, most previous work has focused on single-phase or balanced three-phase

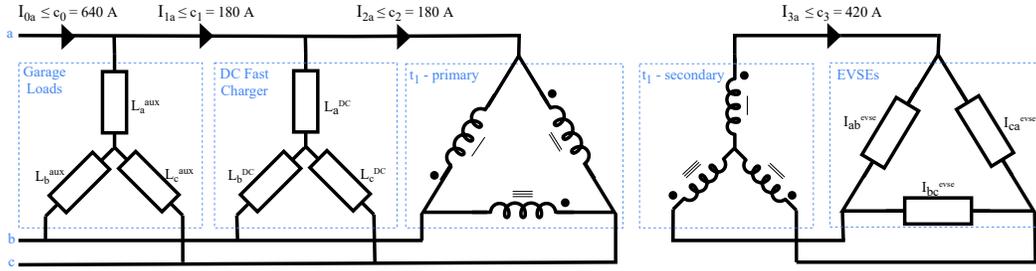


Figure 2.3: Circuit diagram depicting the connection of loads within the California Parking Garage. For simplicity, transformer t_2 is omitted, and all EVSEs between phases A and B have been lumped together as I_{ab}^{evse} , and so forth for BC and CA.

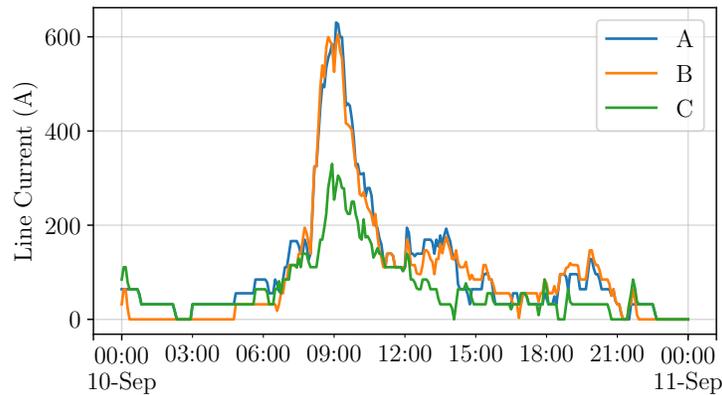


Figure 2.4: Line currents from uncontrolled charging. We note significant current imbalances caused by differences in the allocation of EVSEs to phases and driver preferences. In the ACN, phase AB has 26 stations, whereas phases BC and CA each have 14. This imbalance is caused by the two 8-EVSE pods, which are both on phase AB.

systems, making them inapplicable to charging systems like the ACN. An exception to this is the work of De Hoog et al. [38], which considers an unbalanced three-phase distribution system but only in the case of wye-connected EVSEs. In contrast, the EVSEs in the ACN and most large charging systems in the United States are connected line-to-line.

In general, infrastructure constraints can be expressed as upper bounds on the magnitudes of currents within the system. By Kirchhoff's Current Law (KCL), we can express any current within the system as the sum of load currents. The current draw of EV i at time k can be expressed as a phasor in the form $r_i(k)e^{j\phi_i}$ where ϕ_i is the phase angle of the current sinusoid. We assume in this model that each charging EV has a unity power factor, so ϕ_i is known based on how the EVSE is connected and the voltage phase angles (which we assume are separated by $\pm 120^\circ$).² We can

²This is reasonable since EVs' onboard charger generally includes power factor correction and voltage phase angles can be easily measured.

then model constraints within the electrical system as

$$|I_l(k)| := \left| \sum_{i \in \mathcal{V}} A_{li} r_i(k) e^{j\phi_i} + L_l(k) \right| \leq c_l(k) \quad k \in \mathcal{K}, l \in \mathcal{L} \quad (2.1)$$

Infrastructure limits of the network are indexed by resources $l \in \mathcal{L}$, e.g., l may refer to a transformer or a breaker on a phase. For each constraint l , $|I_l(k)| \in \mathbb{R}$ is the magnitude of the aggregate current through the resource, $c_l(k) \in \mathbb{R}$ is a given capacity limit for time t and $L_l(k) \in \mathbb{C}$ is the current draw through the resource from uncontrollable loads. $A = (A_{li}) \in \mathbb{R}^{|\mathcal{L}| \times |\mathcal{V}|}$ is a matrix which maps individual EVSE currents to aggregate currents within the network. Matrix A can account for both the connection of loads and lines and the effect of transformers, such as the delta-wye transformers in the Caltech ACN. The constraints in (2.1) are second-order cone constraints, which are convex and can be handled by many off-the-shelf solvers such as ECOS, MOSEK, and Gurobi. In some applications, however, these constraints could be too computationally expensive or difficult to analyze. Simpler but more conservative constraints can be derived by observing

$$\left| \sum_{i \in \mathcal{V}} A_{li} r_i(k) e^{j\phi_i} \right| \leq \sum_{i \in \mathcal{V}} |A_{li}| r_i(k)$$

This yields conservative affine constraints in the form

$$\sum_{i \in \mathcal{V}} |A_{li}| r_i(k) + |L_l(k)| \leq c_l(k) \quad k \in \mathcal{K}, l \in \mathcal{L} \quad (2.2)$$

Example 3. *Let's consider how we can calculate A for the subset of the California Parking Garage ACN under transformer t_1 . For modeling simplicity, we assume that the line impedances leading to each station are negligible, allowing us to lump all EVSEs between common phases into a single load represented by current phasors I_{ab}^{evse} , I_{bc}^{evse} , I_{ca}^{evse} , as shown in Fig. 2.3.*

We define the set of all EVSEs connected between lines a and b to be Φ_{ab} , likewise for bc and ca . Then, by KCL, we have

$$\begin{aligned} I_{3a}(k) &= I_{ab}^{evse}(k) - I_{ca}^{evse}(k) \\ I_{3b}(k) &= I_{bc}^{evse}(k) - I_{ab}^{evse}(k) \\ I_{3c}(k) &= I_{ca}^{evse}(k) - I_{bc}^{evse}(k) \end{aligned} \quad (2.3)$$

where

$$\begin{aligned}
I_{ab}^{evse}(k) &:= \sum_{i \in \Phi_{ab}} r_i(k) e^{j\phi_i} \\
I_{bc}^{evse}(k) &:= \sum_{i \in \Phi_{bc}} r_i(k) e^{j\phi_i} \\
I_{ca}^{evse}(k) &:= \sum_{i \in \Phi_{ca}} r_i(k) e^{j\phi_i}
\end{aligned} \tag{2.4}$$

For simplicity we will now focus on phase *a*. We can express the constraint $|I_{3a}| \leq c_3$ in the form of (2.1) with

$$A_{[3a]i} := \begin{cases} 1, & \text{if } i \in \Phi_{ab} \\ -1, & \text{if } i \in \Phi_{ca} \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

To account for the constraint on I_{2a} , we must consider the effect of the delta-wye transformer t_1 . Using circuit analysis, we can relate I_a^2 to the aggregated EVSE currents:

$$I_{2a} = \frac{1}{n} (I_{ab}^{evse} + I_{bc}^{evse} - 2I_{ca}^{evse})$$

where n is the turns ratio of the transformer, which in our system is 4. Hence the constraint on I_{2a} can be expressed in terms of EVSE current magnitudes as:

$$|I_{2a}| = \frac{1}{n} |I_{ab}^{evse} + I_{bc}^{evse} - 2I_{ca}^{evse}| \leq c_2 \tag{2.6}$$

This gives us

$$A_{[2a]i} := \begin{cases} \frac{1}{n}, & \text{if } i \in \Phi_{ab} \\ \frac{1}{n}, & \text{if } i \in \Phi_{bc} \\ -\frac{2}{n}, & \text{if } i \in \Phi_{ca} \end{cases} \tag{2.7}$$

Finally, for I_{1a} and I_{0a} we can use $A_{[0a]} := A_{[1a]} := A_{[2a]}$ with $L_{1a} := I_a^{DC}$ and $L_{0a} := I_a^{DC} + I_a^{aux}$.

Battery Management System behavior

For level-2 EVSEs, each EV's onboard charger and battery management system (BMS) controls its charging rate. The BMS is responsible for estimating the parameters and state of the vehicle's battery, then controlling the charging process to

maintain safe operations [50]. Since we are only interested in the terminal behavior of the BMS and not its internal operations, we can model it as a black-box that solves a possibly non-convex optimization problem.

$$\max_{\hat{r}} U(\hat{r}) \quad (2.8a)$$

$$\text{s.t.} \quad \hat{r} \leq r(k) \quad (2.8b)$$

$$\hat{r} \in \mathcal{B}_k \quad (2.8c)$$

Here \hat{r} is the current drawn by the battery from the EVSE, $r(k)$ is the pilot signal sent by the EVSE at time k , $U(\hat{r})$ is the objective function of the BMS, and \mathcal{B}_k describes constraints on the batteries internal state and dynamics. Reference [51] provides an example of the constraints set and objective for a BMS designed to jointly optimize charging quickly and minimizing battery degradation.

Within this model, there are two important observations. First, the pilot signal, $r(k)$, is an *upper bound* on the current the BMS can draw from the EVSE. This means that if the pilot signal satisfies (2.1), then the actual current draw, \hat{r} , will as well. This allows us to use the pilot signal to ensure safety constraints. However, this also means that relying on the pilot signal can result in imperfect actuation. The difference between the actual charging rate of the vehicle and its allocated pilot is wasted capacity and can adversely affect the objective of the charging algorithm. Second, in practice, we do not know the BMS's objective function, U , or the feasible set, \mathcal{B}_k , for any given vehicle. This means that we cannot incorporate this optimization directly into our scheduling algorithm. In some cases, the objective of the BMS may conflict with the objective of the scheduling algorithm. However, the scheduling algorithm is unaware of this unless it can be inferred from data.

Accounting for BMS behavior requires us to develop algorithms that accurately model the BMS or are robust against deviations from simpler models. While many tractable models for battery charging behavior exist, these models require information about the specific battery pack and the initial state of charge of the vehicle [52], [53]. Other models rely on machine learning to learn the relationship between the state of charge of the battery and its current draw [54]. However, information like state of charge and other internal battery parameters are not available with current charging hardware. Instead, in Chapter 6, we use a model-free approach to estimate battery behavior in real-time and use closed-loop control to account for modeling errors.

EVSE limitations

In practice, EVSEs impose limits on the pilot signals which they support. For example, the J1772 standard does not allow pilot signals below 6 A (except 0). Also, most commercially available EVSEs only support a discrete set of pilot signals. Within the Caltech ACN, we have four types of EVSEs. EVSEs from ClipperCreek only support five pilot signals $\{0, 8, 16, 24, 32\}$ for 32 amp EVSEs and $\{0, 16, 32, 48, 64\}$ for 64 amp EVSEs. EVSEs from AeroVironment, Webasto, Tesla, and OpenEVSE offer more control with 1 A (AeroVironment and Webasto) or 0.1 A (Tesla and OpenEVSE) increments between 6 A and their maximum rate (32 A for Webasto and 80 A for Tesla and OpenEVSE). We can express these limitations mathematically as:

$$r_i(k) \in \rho_i(k) \quad \forall i, t$$

where $\rho_i(k)$ denote the set of allowable charging rates for EV i at time k , which can depend on both the EVSE and our model of the EV's BMS described in Section 2.4.

We also require the charging rate to be non-zero from the time when a car plugs in until its charging demand is met. This constraint helps prevent contactor wear in the EVSEs and improves user experience since most vehicles will notify their owner when charging stops. We can encode this constraint as:

$$r_i(k) \in \begin{cases} \rho_i(k) \setminus \{0\} & \text{if } \sum_{t=1}^T r_i(k) < e_i \\ \{0\} & \text{otherwise} \end{cases} \quad (2.9)$$

where e_i is the energy request of EV i . Unfortunately, these constraints are discrete, making it difficult to incorporate them into optimization-based algorithms. In Chapter 6, we propose heuristics to deal with these discrete constraints.

2.5 Concluding Remarks

This chapter has introduced the Adaptive Charging Network, a framework for large-scale, managed electric vehicle charging facilities. The ACN has been proven at scale through deployments throughout the United States, including the first ACN installed on the Caltech campus in 2016.

Through building the ACN, we have identified practical challenges, including unbalanced three-phase infrastructure, quantization of pilot signals, and non-ideal battery behavior, which require us to rethink classical scheduling approaches. In Chapter 4,

we will show how these realistic models have been incorporated into our open-source simulation environment, ACN-Sim. Then, in Chapter 6, we will describe our scheduling approach based on model predictive control that accounts for these realistic models while giving operators the flexibility to configure the algorithm based on their objectives.

For maximum benefit, ACNs are best suited for sites where at least a portion of the demand has high laxity. Places that generally fit this description are workplaces, schools, multi-unit dwellings, and public charging facilities with long dwell times. ACN sites can incorporate low laxity sessions as well, but these should be a small fraction of the total sessions to get the maximum benefit from load management. Load management like that found in the ACN is necessary for places without available electrical capacity. However, it is still helpful in all sites since we can minimize demand charges / respond to TOU rates, even if the electrical infrastructure is not a problem.

Beyond its operational role of charging hundreds of EVs each week, the Caltech ACN and similar sites at research institutions like JPL, NREL, SLAC, and UC San Deigo, provide a valuable platform for research in managed EV charging. To facilitate this new frontier of research, ACN Research Portal provides open-access data from the Caltech and JPL ACNs. In the next chapter, we will examine this dataset in detail.

Published Works

This chapter contains text and figures originally published in:

Z. J. Lee, G. Lee, T. Lee, C. Jin, R. Lee, Z. Low, D. Chang, C. Ortega, and S. H. Low, “Adaptive Charging Networks: A Framework for Smart Electric Vehicle Charging,” *IEEE Transactions on Smart Grid*, 2021. DOI: 10.1109/TSG.2021.3074437.

Z. J. Lee, D. Chang, C. Jin, G. S. Lee, R. Lee, T. Lee, and S. H. Low, “Large-Scale Adaptive Electric Vehicle Charging,” in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids*, Oct. 2018, pp. 1–7. DOI: 10.1109/SmartGridComm.2018.8587550. © IEEE 2018

Chapter 3

ACN-DATA

The Adaptive Charging Networks at Caltech, JPL, and elsewhere have produced a vast amount of data which has enabled new lines of research within our lab. However, most researchers do not have access to real data from systems like the ACN, precluding them from applying data-driven methods such as machine learning and trace-driven simulations in their work. Instead, most existing works focus on assumed data distributions [29], [55]–[57] or data collected from internal combustion engine vehicles [38], [58]–[61].

To meet this need in the community, we have collected and published ACN-Data. When ACN-Data was first released in 2019, it included data from three sites, totaling 106 level-2 EVSEs [15]. Since this initial release, the dataset has expanded to include a total of 207 level-2 EVSEs and six DC Fast Chargers (DCFC). ACN-Data now covers seven clusters: five at Caltech (including one at a LIGO facility in Louisiana), one at NASA’s Jet Propulsion, and one at an office building in Northern California. These clusters cover common use-cases, including campus/public-use charging and access-controlled workplace charging. The smallest of these clusters has only four level-2 chargers, while the largest has 78 level-2 chargers and two DC Fast Chargers (DCFCs) in a single parking structure. The dataset includes over 80,000 sessions. Each session includes arrival time, departure time, energy delivered, and user data from a mobile app, including vehicle information, estimated departure time, and estimated energy request. The dataset also includes time-series data, including charging current, power, and voltage.

This chapter describes the dataset, explains how it was collected, and provides exploratory analysis to help users understand its important features.

3.1 Research Enabled by ACN-Data

ACN-Data enables new lines of data-driven research which have previously been difficult or impossible for most researchers, such as

- Evaluation of scheduling algorithms and infrastructure designs using trace-driven simulations

- Development and validation of EV charging simulation models
- Load forecasting at the station and aggregate level
- User behavior prediction, such as estimating arrival/departure times and energy needs
- Trace-based studies of grid impacts of large-scale EV charging
- Training of reinforcement learning algorithms for managed EV charging based on real data
- Charging workload analytics
- Evaluation of battery charging curves from real EVs

Works using ACN-Data

At the time of writing, ACN-Data had been downloaded by researchers at 137 organizations across 37 countries. These organizations include universities, national laboratories, K-12 schools, private companies, and non-profits.

Within our lab, we have used ACN-Data to develop and evaluate new algorithms for managed EV charging [28], [62], analyze infrastructure configurations [15], [39], model and predict user behavior [15], analyze impacts of large-scale EV charging facilities on the grid [39], cluster EVs by battery behavior [63], quantify EV charging flexibility [64], and evaluate new pricing schemes to distribute demand charge in large-scale charging facilities fairly [22].

Outside of Caltech, Meenakumar, Aunedi, and Strbac used ACN-Data to evaluate business cases for V2G operators participating in UK electricity markets [65]. Likewise, Schlund, Pruckner, and German used ACN-Data to evaluate and visualize a novel approach to quantifying demand flexibility [66], and Lin, Shang, and Sun used ACN-Data to fit arrival rates for a time-varying Poisson process to evaluate their dynamic pricing scheme for DCFC stations [67]. Both ACN-Data and ACN-Sim were used by Corea et al. to evaluate a novel auction-based mechanism for simultaneously scheduling and pricing EV charging services within a large-scale charging facility [68]. Al Zishan, Haji, Ardakanian use ACN-Data and ACN-Sim to train and evaluate a reinforcement learning agent to schedule EV charging services [69].

Table 3.1: Open EV Charging Data Sets.

Data Set	ACN-Data	Boulder[71]	Palo Alto[72]	Belib Paris[73]	SAP[74]	Dundee[75]	ElaadNL[76]	Pecan Street[14]
Connection Time	✓	✓	✓	✓	✓	✓	✓	✓*
Disconnect Time	✓	✓	✓	✓	✓	✓	✓	-
Done Charging Time	✓	-	✓	-	✓	✓	-	✓*
Energy Delivered	✓	✓	✓	✓	✓	✓	✓	✓*
Gas Savings	-	✓	✓	-	-	-	-	-
GHG Savings	-	✓	✓	-	-	-	-	-
Location	✓	✓	✓	✓	✓	✓	✓	-
User ID	✓	-	-	✓	-	-	✓	✓*
User Inputs	✓	-	-	-	-	-	-	-
State of Charge	✓**	-	-	-	-	✓	-	-
Price	-	-	✓	-	-	✓	✓	-
State Time Series	✓	-	-	-	-	-	✓	✓
Control Time Series	✓	-	-	-	-	-	-	-

*Field can be inferred from charging power time series.

**State of charge is only available for DCFC sessions.

3.2 Other Electric Vehicle Charging Data Sets

ACN-Data is among a growing number of public EV charging datasets. Amara-Ouali et al. provide a survey of these datasets, including datasets of charging sessions (like ACN-Data) and locations of charging ports, vehicle registration data, and traffic surveys [70]. Table 3.1 provides a summary of the open datasets most similar to ACN-Data. These other datasets contain many of the same fields as ACN-Data, however, none include user inputs, and very few have time-series data. In addition, only the SAP dataset includes workplace charging, which is an important use case with unique workload features. The other datasets are focused on public use chargers or home charging in the case of Pecan Street.

3.3 Data Records

ACN-Data contains five types of data records.

Sessions

Each session record contains information about a single charging session. The fields included for each session are outlined in Table 3.3. Each session also includes zero or more user inputs collected via the mobile app. The fields from these user inputs are given in Table 3.4.

Session data is available in four JSON files, which are outlined in Table 3.2. Since ACN-Data is continuously updated, these files are appended to with each release. We expect to release new data monthly. We use an append model rather than releasing a new file each month to make loading and analyzing the dataset easier for

Table 3.2: Filenames for Session records.

Filename	Description
level2_sessions_all.json.gz	All sessions from level-2 EVSEs before filtering.
dcfc_sessions_all.json.gz	All sessions from DCFCs before filtering.
level2_sessions_clean.json.gz	Level-2 sessions after filtering.
dcfc_sessions_clean.json.gz	DCFC sessions after filtering.

users.

EVSE State Time Series

Table 3.5 gives the fields included in the EVSE state time series. The time series has an approximately four-second resolution for level-2 stations and a 10-second resolution for DCFCs. Some EVSEs do not support all the fields in Table 3.5. For a breakdown of supported fields by EVSE type, see Table 3.11. Due to an error in data retention policies, only *Charging Current (A)* is available for sessions from Arroyo_Garage_01 before June 17, 2020.

For ease of analysis, we provide the EVSE state time series in two formats, per session and per station. Per session time series are particularly helpful for analyzing battery behavior or other analysis which is session-specific. We provide the EVSE state from the session's connectionTime to disconnectTime in a single, compressed CSV file for these records.

Per station time series are more appropriate for other types of analysis, such as analyzing congestion in the network. For these records, we divide the time series into monthly chunks, each of which is stored in a compressed CSV file.

Control Time Series

Likewise, we include a time series of the allocations made by the control algorithm. The fields included in this time series are given in Table 3.6 with a four-second resolution. Because we do not actively control DCFCs, control time series are only available for level-2 stations. While this time series has a four-second resolution, the charging algorithm generally only updates its allocations every minute. The exception to this rule is when an event, such as a plugin, unplug, or parameter change occurs between periodic calls to the algorithm. Like the EVSE state time series, we provide the control time series per session and per station.

Aggregate Power Time Series

For some use cases, only the aggregate power draw of the site is necessary. Examples include time-series forecasting of aggregate EV load or evaluating the impact of EV charging on the grid. For these use cases, we provide a time series of the aggregate power draw of each cluster. The fields included in this time series are given in Table 3.6 with a 10-second resolution. We divide the aggregate power time series into monthly chunks, each of which is stored in a compressed CSV file. This aggregate power data is taken from a power meter at the point of interconnection, so it includes losses and standby power and charging loads.

Station Information

To help users analyze the dataset, we describe the charging stations at each site. This dataset includes the fields shown in Table 3.8.

Table 3.3: Schema for Session Records.

Field	Type	Description
clusterID	string	Unique identifier for a subset of EVSEs at a site, such as a single garage.
clusterName	string	Name of the cluster.
connectionTime	datetime	Time when the EV plugged in.
disconnectTime	datetime	Time when the user unplugged.
duration	float	Duration of the session in minutes.
doneChargingTime	datetime	Time of the last non-zero charging rate.
evseLevel	string	Power level of the EVSE. Either level-2 or DCFC.
filename	string	sessionID modified to remove characters not valid for filenames.
kWhDelivered	float	Amount of energy delivered during the session in kWh.
sessionID	string	Unique identifier for the session.
siteID	string	Unique identifier for the site.
siteName	string	Name of the site.
spaceID	string	Identifier of the parking space. Unique within a cluster.
stationID	string	Unique identifier of the EVSE.
timezone	string	Timezone for the site. Based on pytz format.
userID*	string	Unique identifier of the user.
userInputs*	list[User Input]	Inputs provided by the user. See Table 3.4
vehicleDoeID	string	United States Department of Energy ID for the vehicle.
vehicleMake	string	Manufacture of the vehicle.
vehicleModel	string	Model name of the vehicle, including variant, i.e. Long Range.
vehicleYear	int	Model year of the vehicle.
vehicleType	string	Either "EV" for a pure electric vehicle or "PHEV" for plug-in hybrids.

*Field not available for every session.

Table 3.4: Schema for User Inputs.

Field	Type	Description
kWhDeliveredWhenModified	float	Energy delivered so far when the user updated their parameters in kWh.
kWhRequested	float	Energy requested by the user in kWh.
milesRequested	float	Miles requested by the user through the app.
minutesAvailable	float	Minutes the user expects to be available for charging, measured from connectionTime.
modifiedAt	datetime	Time when the user updated their parameters.
requestedDeparture	datetime	When the user expects to leave.
source	string	How the parameters were set, either "user" or "default".
WhPerMile	float	Efficiency of the user's EV in Wh/mile.

Table 3.5: Schema for EVSE State Time Series.

Field	Type	Description
timestamp	datetime	When the measurement occurred.
Charging Current (A)	float	Measured current draw of the EV.
Actual Pilot (A)	float	Pilot signal sent to the EV.
Voltage (V)	float	Voltage at the EVSE.
Energy Delivered (kWh)	float	Energy delivered so far in the session.
Power (kW)	float	Power draw of the EV.
State of Charge	float	State of charge of the EV. Only available for DCFC sessions.

Table 3.6: Schema for Control Time Series.

Field	Type	Description
timestamp	datetime	When the measurement occurred.
Pilot Signal (A)	float	Pilot signal calculated by the control algorithm.
Previous Charging Current (A)	float	Measured charging current from the previous time step.
Maximum Rate Estimate (A)	float	Estimate of the maximum charging rate of the EV at the next time step. Calculated using the Rampdown Algorithm[62].

Table 3.7: Schema for Aggregate Power Time Series.

Field	Type	Description
timestamp	datetime	When the measurement occurred.
Aggregate Power (kW)	float	Measured aggregate power of the cluster.

3.4 Data Collection and Processing

In this section, we describe the process of gathering, cleaning, and formatting the dataset. Understanding how the dataset was collected and processed is important in understanding the dataset and evaluating its use.

Table 3.8: Schema for Station Information Records.

Field	Type	Description
Station ID	str	Unique identifier of the EVSE.
Space ID	str	Identifier of the parking space. Unique within a cluster.
Site Name	str	Name of the site.
Cluster Name	str	Name of the cluster.
EVSE Type	str	Make and model of the EVSE.
Maximum Rate (A)	float	Maximum charging rate supported.
Minimum Rate (A)	float	Minimum non-zero rate supported.
Rate Increment (A)	float	Pilot signal increment supported.
Nominal voltage (V)	float	Nominal voltage of the EVSE. At many sites, the actual voltage is higher to increase throughput.
Fallback Rate (A)	float	Default value used after a timeout when no pilot signal has been received.
Timeout (s)	float	Time to wait after receiving a pilot signal before using fallback rate.

Data Sources

We begin by describing the sites and hardware used to collect the dataset. Depending on the purpose of a study, data from certain sites might be more useful than others. We provide only a brief description of this hardware here, as it is described in detail in Chapter 2.

Sites.

Data is collected from seven charging clusters within three sites. An overview of these clusters is shown in Table 3.9 and Table 3.10. A cluster refers to a collection of EVSEs in a single location sharing the same electrical infrastructure. Each site can have one or more clusters. The clusters belonging to a single site may not be geographically close. For example, LIGO_01 is at a research location owned by Caltech in Louisiana, while the other Caltech clusters are all on its campus in Pasadena, CA. Other clusters can be very close geographically. For example, California_Garage_01 and California_Garage_02 are in the same parking structure with stations very close to one another. A brief description of each cluster is given below:

- *California_Garage_01* - Is an underground parking garage on the south edge of the Caltech campus near the campus gym, frequently used by Caltech professors, staff, and students, as well as community members. Because of its proximity, many users charge their vehicles in the evening while using the gym.
- *California_Garage_02* - Is a collection of higher power EVSEs in the same



Figure 3.1: Photos of the N_Wilson_Garage_01 ACN, which is one of the charging sites used to collect data.

garage as California_Garage_01. These chargers were added after those in California_Garage_01.

- *S_Wilson_Garage_01* - Is a parking garage on the eastern edge of the Caltech campus. These EVSEs replaced four existing EVSEs from a different vendor in Dec. 2019. The site is often used by the Caltech community and the general public. An example of the hardware at the North Wilson parking garage is shown in Fig. 3.1.
- *N_Wilson_Garage_01* - Is a parking garage on the eastern edge of the Caltech campus adjacent to the S. Wilson Garage. DCFCs are on the ground floor, while level-2 chargers are on the roof under a solar canopy. This site is often used by Caltech and public users.
- *LIGO_01* - Is a parking lot at the LIGO research facility owned by Caltech in Walker, LA. The site is not open to the public.
- *Arroyo_Garage_01* - Is a parking garage on the Jet Propulsion Laboratory Campus. Access is limited to JPL employees. Because of the high number of EVs at JPL, drivers use a swapping system where drivers who arrive in the morning move their vehicle after charging so that another user can plugin in the afternoon. JPL also has a unique schedule where many employees do not travel to the lab every other Friday, which affects the usage patterns.
- *Parking_Lot_01* - Is a parking lot at an office building in Northern California. Access is limited to employees, and usage is typical for a small office.

Four of the seven clusters are oversubscribed, meaning that the actual capacity of their electrical infrastructure is less than the sum of the maximum power draw of

Table 3.9: Description of charging clusters.

Site Name	Cluster Name	Capacity	Unmanaged Max	Latitude	Longitude	Open to Public
Caltech	California_Garage_01*	150 kW	410 kW	34.134765	-118.127183	✓
	California_Garage_02*	150 kW	436 kW	34.134765	-118.127183	✓
	S_Wilson_Garage_01	36 kW	26.6 kW	34.140269	-118.128354	✓
	N_Wilson_Garage_01	230 kW	665.9 kW	34.138826	-118.128361	✓
	LIGO_01	65 kW	53.25 kW	30.563496	-90.773777	
JPL	Arroyo_Garage_01	195 kW	346.1 kW	34.198841	-118.170402	
Office01	Parking_Lot_01	115 kW	53.25 kW	-	-	

*California_Garage_01 and California_Garage_02 are in the same parking structure.

Table 3.10: EVSEs at each charging cluster.

Site Name	Cluster Name	AV	CC32	CC64	DX	TWC	DCFC
Caltech	California_Garage_01*	46	8	-	-	-	1
	California_Garage_02*	-	-	14	-	12	1
	S_Wilson_Garage_01	4	-	-	-	-	-
	N_Wilson_Garage_01	-	-	-	45	10	4
	LIGO_01	-	-	-	8	-	-
JPL	Arroyo_Garage_01	52	-	-	-	-	-
Office01	Parking_Lot_01	8	-	-	-	-	-

*California_Garage_01 and California_Garage_02 are in the same parking structure.

their EVSEs (see Table 3.9). This actual capacity is limited either by the size of the transformer feeding the EVs or by another bottleneck in the electrical infrastructure. Because actual capacity is limited, all EVSEs cannot be utilized at their maximum power draw simultaneously. This necessitates managed charging.

Charging Hardware.

Chargers within the ACN can be divided into two groups; level-2 EVSEs, which provide AC power to a vehicle's onboard charger, and DC Fast Chargers, which provide DC power directly to the vehicle's battery. There are five types of level-2 EVSEs included in this dataset. These are outlined in Table 3.11. Each EVSE type has its own limitations on the maximum charging rate and the granularity of the pilot signal. EVSEs also differ in the data that they collect. All EVSEs collect current, power, and energy delivered (on a per session basis). However, some also record the actual pilot signal received by the EVSE and voltage. Level-2 EVSEs in our system do not support communication from the vehicle, so we do not have access to the vehicle's state of charge or other internal parameters. Every four seconds, this data

Table 3.11: EVSE types.

EVSE Type	Charging Capabilities			Data Collection Capabilities				
	Max Rate	Min Rate	Increment	Current	Power	Energy	Pilot	Voltage
AeroVironment (AV)	32 A	8 A	1 A	✓	✓	✓	✓	✓
ClipperCreek - 32 A (CC32)	32 A	8 A	8 A	✓	✓	✓	-	-
ClipperCreek - 64 A (CC64)	64 A	16 A	16 A	✓	✓	✓	-	-
WebastoDX (DX)	32 A	8 A	1 A	✓	✓	✓	✓	✓
Tesla Wall Connect (TWC)	80 A	8 A	1 A	✓	✓	✓	✓	-

is transmitted to the central controller via the wireless mesh network.

The dataset includes sessions from six DC Fast Chargers, all of which are Tritium Veefil RT50 50kW level-3 DCFCs. Unlike the level-2 EVSEs, we do not actively limit the charging rate of these DCFCs. The Veefil chargers measure charging current, power, session energy delivered, and the vehicle's state of charge. State of charge is available because, unlike our level-2 chargers, DCFCs have a bi-directional communication channel to the vehicle. Data from the DCFC is communicated to the central controller via TCP/IP over an Ethernet cable.

Mobile App.

Users interact with the ACN via a mobile app. Users connect to a charging session by scanning a QR code on the EVSE with an app on their phone. They then enter the amount of time they will leave their vehicle (*minutesAvailable*) and the number of miles they want to be added to their vehicle's range (*milesRequested*). The app then uses information about the vehicle's efficiency, which the user provides when registering, and computes the energy the system must deliver to the EV by the user's departure time. This claiming process can occur before or after the user actually plugs in their vehicle.

Users can update these parameters throughout their charging session. We store user inputs as a list within each session record. Each entry is one user update. In each update, we include when the user updated the parameters and how much energy had been delivered that that time. This is helpful in analyzing why the user might have requested the change.

Control Algorithm.

For several of the clusters in ACN-Data, electrical capacity is limited. These clusters require managed charging to prevent overloading the infrastructure. The ACN uses

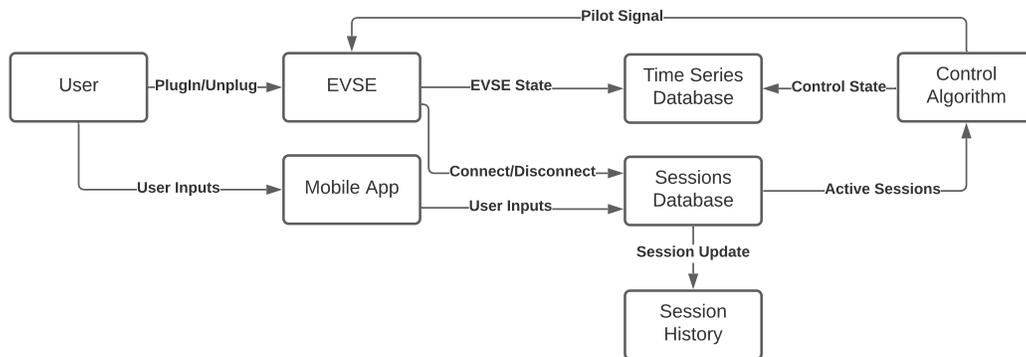


Figure 3.2: Data collection pipeline for the ACN.

a model predictive control algorithm to determine the charging rate allocated to each EV based on congestion in the electrical system and each user’s inputs. This algorithm is described in detail in Chapter 6. The outputs of this control algorithm are included in the control time series for each level-2 station.

Power Meter.

Each cluster has a power meter that measures the aggregate power draw of the cluster at the cluster’s root node. This data is sent to the central controller via TCP/IP over an Ethernet cable.

Data Collection

The data collection pipeline for the ACN is outlined in Fig. 3.2. When a user first plugs in their vehicle (or enters their parameters in the app), the PowerFlex system generates a new charging session which is stored in a relational database. This session is given a unique ID and stores basic information about the session, including which EVSE it belongs to and when it began. Initially, each session is given a default duration and energy request. These defaults are replaced by user inputs when the user claims the session via the app. The session is also updated by events, such as when users update their parameters or unplug their vehicles. Each time an event occurs, the new state of the session is stored in the session history table.

In addition, as the EV charges, each EVSE reports its state, including charging current, voltage, power draw, and energy delivered to the central controller, where it is stored in a time-series database. A control algorithm manages the charging process.

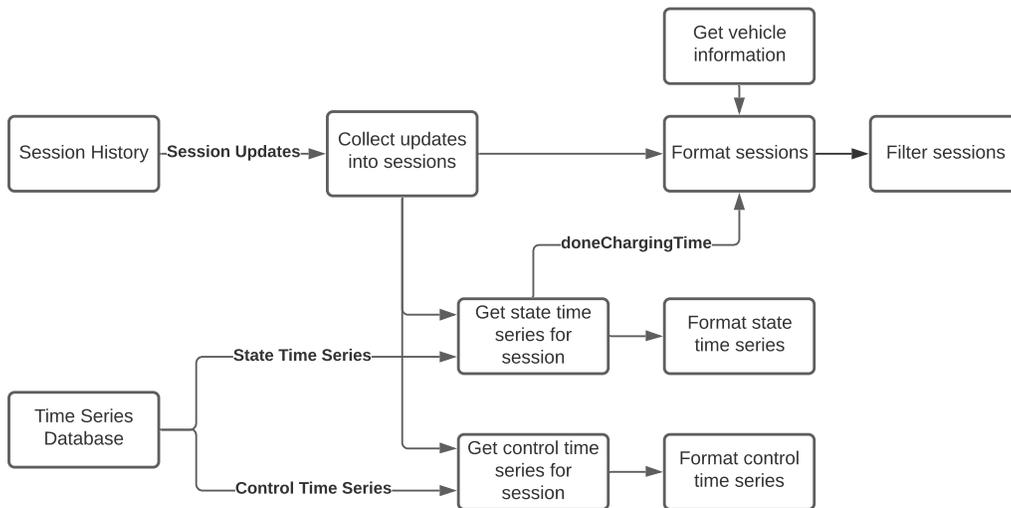


Figure 3.3: Data processing pipeline for the ACN.

The inputs for this algorithm are the parameters of all active charging sessions, and it returns a pilot signal for each EV. This pilot signal and the algorithm's estimate of the vehicle's maximum charging rate are stored in the time-series database. The pilot signal is then sent to the EVSE. If this pilot signal is not delivered within the timeout window, the EVSE will fall back to its minimum charging rate. Most EVSEs include the pilot signal they receive in their state. If fallback occurs, the pilot signal in that EVSE's state will be its minimum pilot.

Data Processing

The pipeline used to clean and format ACN-Data is shown in Fig. 3.3. To build sessions for ACN-Data, we extract all updates from a given time range from the session history table. We then group these updates by their *sessionID* to identify events that belong to the same session and sort these events by the time they occurred. For user input fields, which can change multiple times throughout the session, we store these updates in a list.

After validating the events, we consolidate them into a single session, rename fields, and convert units to make the object easier for researchers to use. After formatting the sessions, we store them in two zipped JSON documents, one for level-2 sessions and one for DCFC sessions. We also do a final filtering step to remove sessions with missing or small values of *duration* and *kWhDelivered*. We use a threshold of 1 minute for *duration* and 0.02 kWh for *kWhDelivered*. We also remove any sessions

which have an average charging rate ($kWhDelivered / duration$) more than 10% higher than the power rating of their EVSE. In total this removes 9,472 (of 89,296) level-2 sessions. For DCFC sessions we remove 50,122 erroneous sessions, leaving 1,823 sessions. This high error rate for DCFCs is because of a bug in the ACN software, which produced many erroneous sessions for each true session DCFC session. This bug has since been fixed. Additionally, some DCFC sessions have start and end times entirely contained within the start and end times of other sessions. These so-called "contained" sessions are removed from the dataset. We choose to include both the raw and cleaned datasets so that researchers can understand our process and choose to clean the dataset differently if they so choose.

We extract the time series of EVSE state and control signals from the time-series database and reformat them into a CSV file for each session. Using the state time series, we find the session's *doneChargingTime*, which is defined as the time of the last non-zero charging current measured. If kWhDelivered is missing for the session, we take the maximum of the Energy Delivered (kWh) field from the state time series and replace this value. After extracting, cleaning, and formatting the data, we save time series as zipped CSV files.

To get monthly EVSE state and control time series, we iterate over all charging stations and extract data for the given month from the time series database. We reformat this data into CSV files for each station and month. Similarly, each month we gather the aggregate power data for each cluster from the time-series database and reformat it into CSV files.

3.5 Lessons from ACN-Data

In this section, we analyze the dataset and provide important lessons which are helpful for researchers who want to apply this dataset in their work. Many of these lessons are not surprising, but we include them here to show that our intuitions around workplace and public-use charging hold true.

Pricing can have a major influence on charging behavior.

Intuitively, the price of energy influences the utilization of a charging facility. In late 2018 we observed a natural experiment at the California_Garage_01 cluster. Before November 1, 2018, charging at California_Garage_01 (the only Caltech cluster fully operational at the time) was free. We provided generous default parameters for

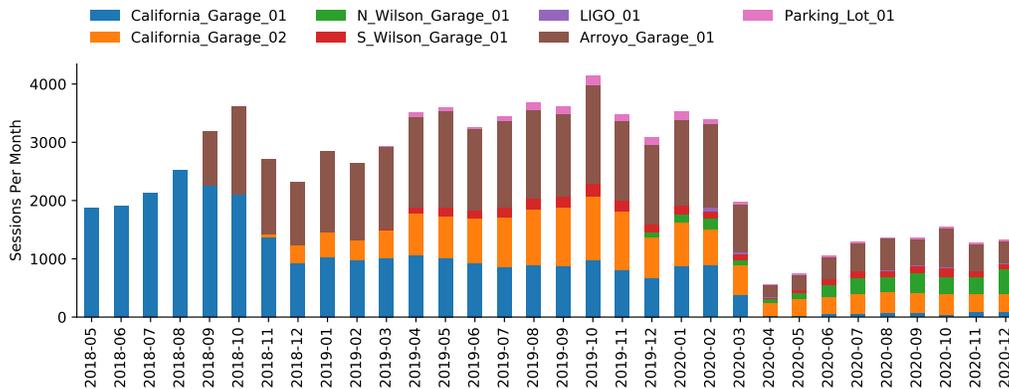


Figure 3.4: Session records collected per month at each of the seven clusters in ACN-Data. The earliest sessions are from California_Garage_01 in May 2018.

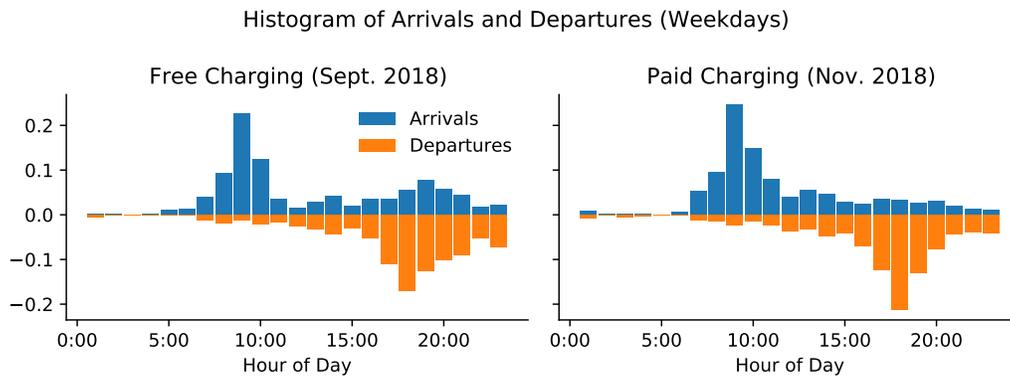


Figure 3.5: Histogram of arrival and departure times before and after payment was required at California_Garage_01.

session duration and energy requested for drivers who did not use the mobile app. On October 1, 2018, we began transitioning to a paid model, where users of the mobile app paid \$0.12 / kWh to cover energy costs. Users who did not use the app, however, still received free charging. Beginning November 1, 2018, sessions were terminated after 30 minutes if a user did not claim their charging session, so payment was required for all sessions longer than 30 minutes.

From Fig. 3.4, we see a small dip in the number of charging sessions at the California_Garage_01 between September and October and a sharp decline in usage between October and November. This matches our intuition that free charging would be more popular than paid. However, the magnitude of this change is surprising.

This change also influenced the distribution of arrivals and departures at the site. From Fig. 3.5, we can see that when charging was free, there was a second uptick in arrivals around 7:00 pm. This corresponds to community members using the

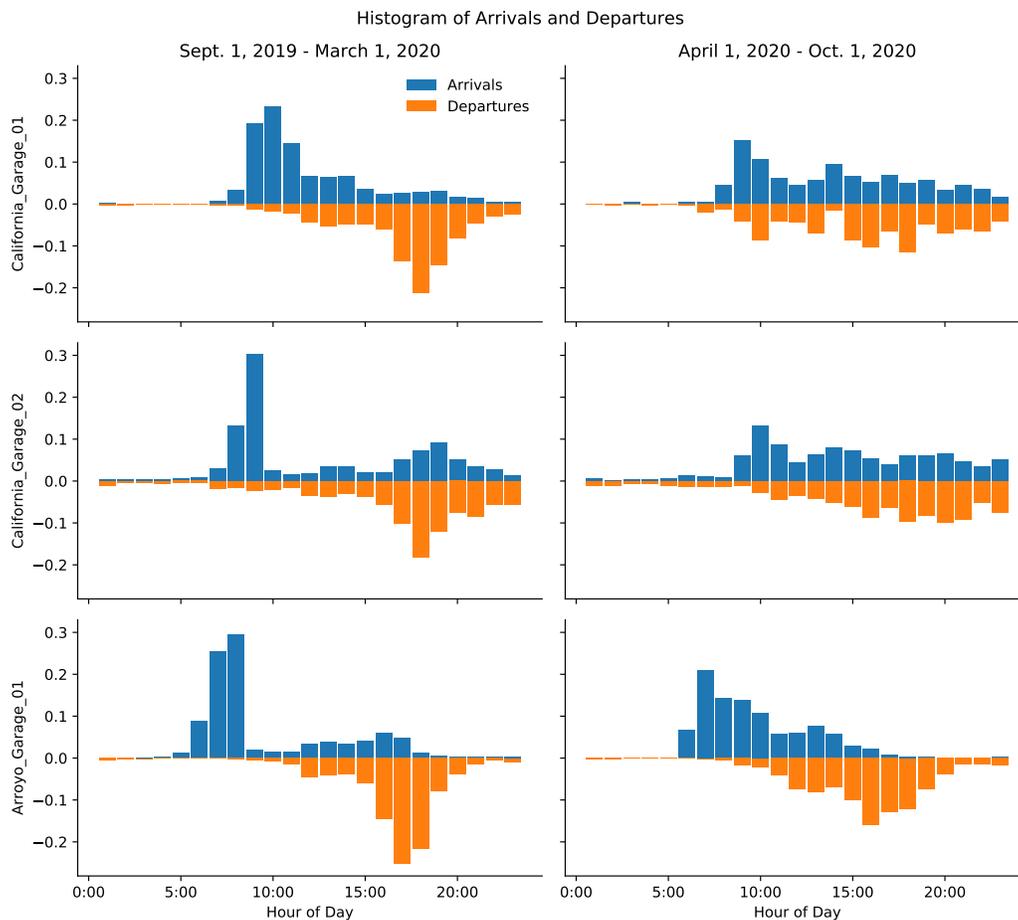


Figure 3.6: Histogram of arrivals and departure times on weekdays in the six months before and after the beginning of COVID-19 lockdowns.

charging system after work or while at the campus gym located beside the California Parking Garage. However, just one month later, we see a significant drop in usage after work hours. Our hypothesis is that community members were willing to drive to this site to charge when it was free but preferred to charge at home or another site when the price increased. While this is not a surprising result, it does indicate the price sensitivity of EV drivers. This sensitivity could be harnessed to influence behavior.

The COVID-19 pandemic has significantly changed the utilization of workplace charging.

Campus closures due to COVID-19 have had a major impact on the usage patterns at all clusters in the dataset. Lockdowns at Caltech and JPL began in mid-March 2020. Limited operations began again in late 2020, but as of January 2021, no

sites had returned to normal operation. From Fig. 3.4, we can observe a significant decrease in the number of charging sessions between February and April. These numbers increase slightly over the next eight months but are still roughly one-third of their previous values. Fig. 3.6 shows histograms of the arrival and departure times for sessions in the six months before and after the COVID-19 shutdowns. We observe that there is still a peak in arrivals in the morning after the shutdowns, however, arrivals and departures are much more dispersed at all three clusters. This is expected as very few people were working normal work hours during the lockdown period.

This period is important in two respects. First, researchers should be aware that usage patterns during this period are likely not representative of normal behavior. However, this period also provides an interesting natural experiment. Researchers can use this step change to investigate how robust their approaches are to changes in the underlying distribution of user behavior. Second, since ACN-Data continues to grow, it will provide insight into how charging behavior has changed throughout the pandemic and the recovery period afterward. Some changes, such as an increase in remote work, are likely to change charging behavior long term. However, we do expect workplace charging patterns to resume as employees and students are allowed to return.

Usage patterns are different on weekdays and weekends.

We observe significant differences between charging patterns on weekdays and weekends, as shown in Fig. 3.7. The figure shows that the weekday arrival distribution has a morning peak at both sites. For conventional charging systems, these peaks necessitate a larger infrastructure capacity and lead to higher demand charges. In addition, as EVs adoption grows, these morning spikes in demand could prove challenging for utilities as well. As expected, departures are analogous to arrivals. They begin to increase as the workday ends, with peaks in the period 5-7 pm at both the California Garage and Arroyo Garage. Departures at the Arroyo Garage (JPL) tend to begin earlier, which is consistent with the earlier arrival times at that cluster. Departures at the California Garage tend to stretch into the night owing to the heterogeneity of individual schedules and later arrivals.

In contrast, on weekends, we see a much more uniform distribution of arrivals and departures throughout the day. This uniformity is probably due to the aggregation of many highly heterogeneous weekend schedules. We note that charging systems

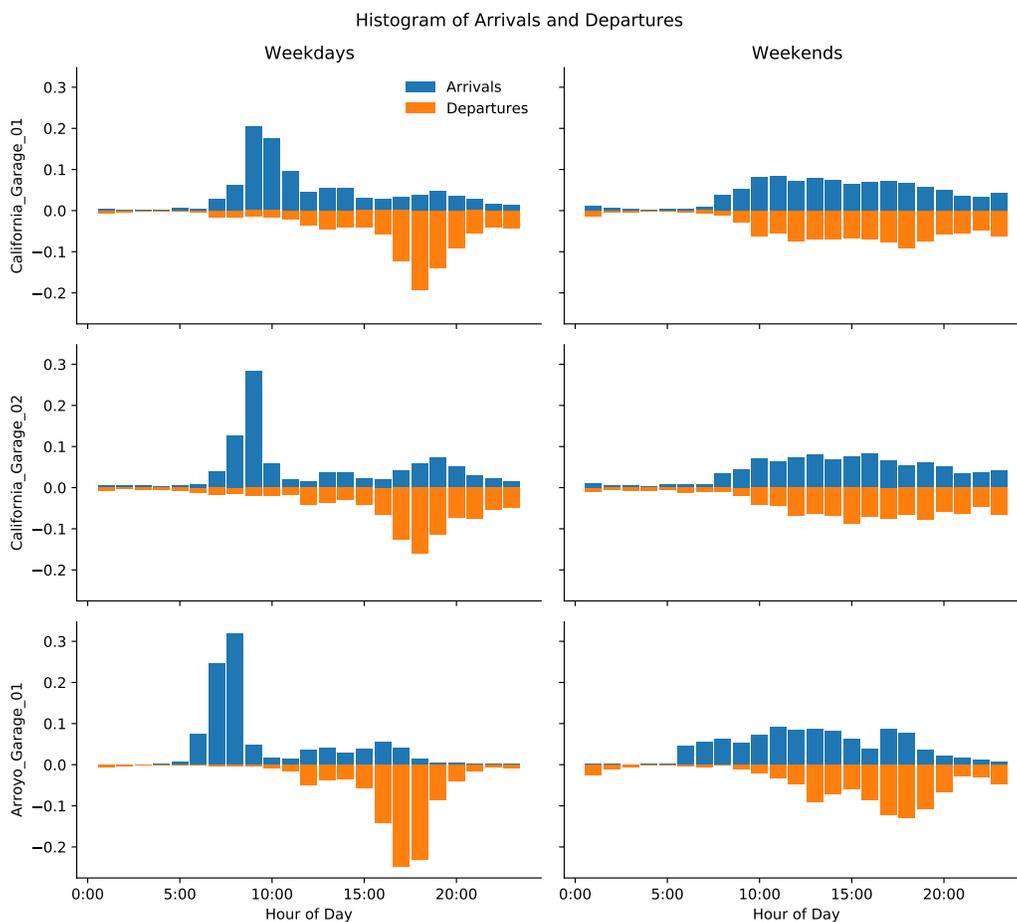


Figure 3.7: Histogram of arrivals and departure times on weekdays vs. weekends. Data is taken from before March 1, 2020, which was before any COVID-19 lockdowns. We include data from California_Garage_01, California_Garage_02, and Arroyo_Garage_01, which had sufficient usage during this period to generate meaningful histograms.

should account for both types of workloads, which we will discuss in more detail in Chapter 6.

Users in workplace environments have significant laxity.

Laxity is a measure of job flexibility. We define the initial laxity of an EV charging session i as

$$\text{LAX}(i) = d_i - \frac{e_i}{\bar{r}_i}$$

$\text{LAX}(i) = 0$ means that EV i must be charged at its maximum rate \bar{r}_i over the entire duration d_i of its session in order to meet its energy demand e_i . A higher value of $\text{LAX}(i)$ means more flexibility in satisfying its energy demand. Figure 3.8 shows the distribution of initial laxities in our dataset. From it, we see that for weekdays,

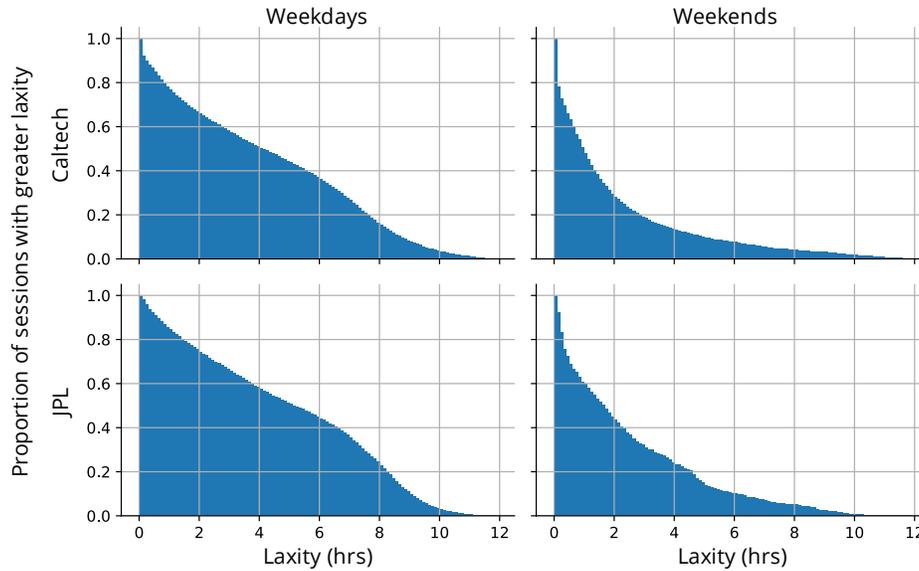


Figure 3.8: Empirical complementary cumulative distribution of session laxity at Caltech and JPL using data from May 1, 2018 through March 1, 2020.

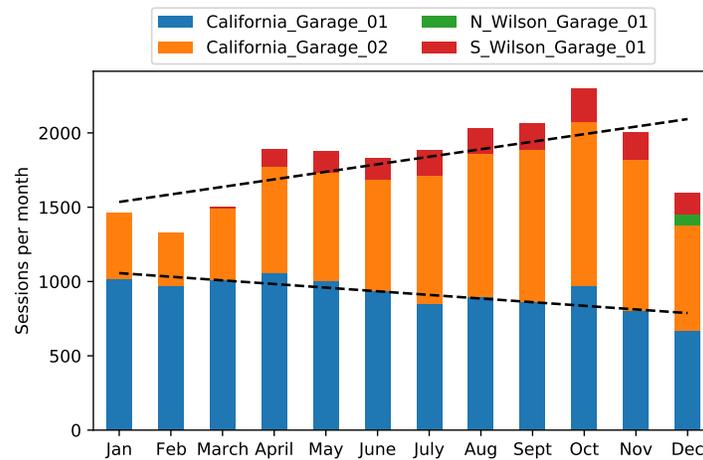


Figure 3.9: Sessions per month on the Caltech campus in 2019. Top dashed line shows a linear fit on total sessions while the bottom line shows a linear fit on the usage of California_Garage_01.

most EVs have high laxity. On weekends laxity tends to be lower as drivers tend to want to get charged and get on with their day.

Charging workloads redistribute as additional charging stations are installed in an area.

We find that as additional charging clusters are installed in an area, drivers will redistribute among them based on their preferences, such as which is more conve-

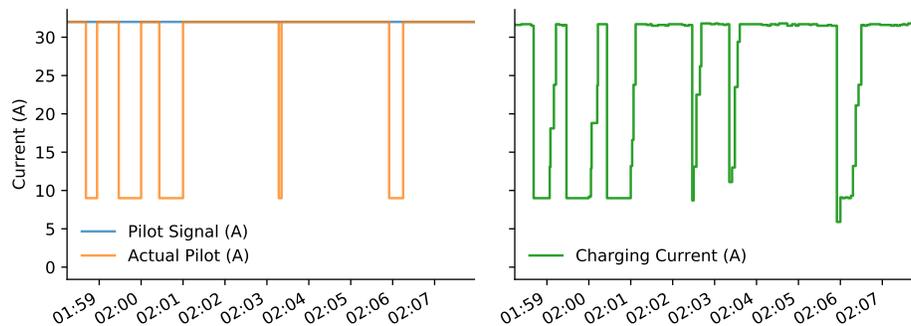


Figure 3.10: Fallback of the actual pilot to 8 A when communication times out. The left plot shows the pilot signal generated by the algorithm and the pilot actually sent to the vehicle. The right panel shows the actual charging current of the vehicle.

nient or offers faster charging. From Fig. 3.9, we observe a slight negative trend in utilization for the California_Garage_01 cluster in 2019, despite overall growth in the number of level-2 charging sessions at Caltech. This is not surprising. Many users prefer the EVSEs in the California_Garage_02 cluster because they are higher power than the California_Garage_01 EVSEs. Likewise, some users prefer to use the S_Wilson_Garage_01 chargers, as they are in a different area of the campus, which is closer to many academic buildings and graduate student housing. N_Wilson_Garage_01 was not added until December 2020.

This observation is not surprising but does demonstrate the importance of considering EV charging infrastructure as a whole rather than just individual clusters. As more charging infrastructure is added to an area, the usage patterns of existing sites are likely to change. The coupling between charging stations that are geographically close should be considered when analyzing usage patterns and making recommendations for charging system designs.

Coupled with our observations on price sensitivity, an interesting line of research would be studying how pricing signals can encourage drivers to choose a particular charging cluster, which might be beneficial in alleviating congestion at the local or grid level.

Congestion in the communication network can cause fallback.

Communication between the central controller and level-2 EVSEs is done over a wireless mesh network. This network is imperfect, and some messages never reach their intended recipients. As a safety mechanism, if an EVSE does not receive a valid pilot signal from the controller within a specified time period (usually 20

seconds), it will fall back to a safe minimum charging current (usually 8 A). We can see the effect of fallback by comparing the Actual Pilot (A) field in the EVSE state time series with the Pilot Signal (A) from the control time series.

When doing analysis, sudden reductions in Actual Pilot (A) to 8 A generally indicates fallback and not congestion. Fig. 3.10 shows one example of fallback. We can see that while the pilot signal generated by our control algorithm is consistently at 32 A, the actual pilot at the EVSE will sometimes dip to 8 A. This indicates that the station is in fallback. These dips are usually quite short but can be frequent when the communication network is heavily loaded.

3.6 Concluding Remarks

This chapter presents ACN-Data, a public dataset of EV charging sessions and time series collected from the Adaptive Charging Networks at Caltech, JPL, and an office building in Northern California. ACN-Data is one of the largest public datasets of its kind and the only one to include user inputs and high-fidelity time series with 4 to 10-second resolution.

ACN-Data opens up new lines of data-driven research in the EV charging space, including trace-based simulation, load forecasting, workload modeling. The high-fidelity time-series data also enables researchers to examine battery management system behavior and its interactions with smart charging systems.

Since its initial release in 2019, ACN-Data has been downloaded by over 200 researchers at 137 organizations in 37 countries. These include universities, K-12 schools, non-profits, national laboratories, and companies. As of April 2021, the dataset has been used in 13 published works, see Section 3.1.

This chapter outlines the second major release of the dataset. In this release, we have added sessions from four new clusters and additional data fields like vehicle information in session records. We have also expanded our time-series data to include voltage, power, and energy delivered, as well as state of charge for DCFC. This highlights an important feature of ACN-Data: it is frequently updated with new sites, new sessions, and new fields. We hope this will mean that the dataset stands the test of time and enables many new lines of EV charging research.

In the future, we hope to include additional sites to ACN-Data representing different use cases, such as multi-unit dwellings, shopping centers, hospitals, or fleet depots. Because of our partnership with PowerFlex systems, we have access to data from

hundreds of sites, though privacy concerns prevent us from publishing data from all of them. For this reason, privacy-preserving methods are being explored to release aggregated data from additional sites.

We also plan to include additional data sources, such as on-site solar generation and building load, which are often considered in EV charging research.

Chapter 4

ACN-SIM

While the Adaptive Charging Network has allowed us to identify many interesting challenges in real-world EV charging systems, we recognize that most researchers will not have access to such physical testbeds. However, it is still important that these researchers have a realistic environment in which to evaluate their algorithms. To this end, we have developed a modular, data-driven, realistic simulation environment for testing scheduling algorithms for EV charging systems called ACN-Sim. ACN-Sim is open-source and available on Github [77]. It includes realistic models of the many components of a real EV charging system, such as electrical infrastructure, charging stations, and EV battery management systems. It is also designed to be highly modular, meaning that users can replace each component to model different types of hardware or different levels of fidelity.

ACN-Sim has three main objectives:

Accelerating algorithm research ACN-Sim helps to accelerate algorithm research by removing the burden of designing realistic simulation environments from researchers and allowing them to instead focus on designing new algorithms

Exposing practical challenges Most researchers interested in designing algorithms for DER applications do not have direct experience with building EV charging systems. Doing so ourselves has exposed us to some of the challenges of practical systems such as unbalanced three-phase infrastructure, unreliable user inputs, non-ideal battery behavior, and discrete control settings. By exposing these issues to researchers, they can evaluate how these practical constraints affect their algorithms and propose new innovative approaches to address these issues.

Facilitating reproducibility Good science requires reproducible results and fair comparisons between new methods and prior-art. However, the lack of a common simulation environment and benchmarks has made reproducing other's results and comparing them to new methods more difficult than it should be. By providing a common, open-source platform on which algorithms can be developed and

evaluated, ACN-Sim allows researchers to easily share implementations of their algorithms and code for their experiments with other researchers in a portable and easy-to-understand way.

4.1 Existing Tools and Simulators

Open-source tools and simulators have a long history of supporting smart grid research. MATPOWER[78] makes it easy to solve power flow and optimal power flow problems in MATLAB. It has inspired projects in other languages, including PandaPower[79] in Python and PowerModels.jl in Julia[80]. Other important simulators include OpenDSS[81] and GridLab-D[82], which enable large-scale studies of the distribution system. These tools have demonstrated the importance and impact of open tools within the smart grid community. ACN-Sim integrates with many of these, including MATPOWER, PandaPower, and OpenDSS, to enable studies of the grid impacts of EV charging.

ACN-Sim is not the first simulator specific to EV charging. V2G-Sim was developed at Lawrence Berkeley National Laboratory and has been used to evaluate EVs' ability to meet drivers' mobility needs in the context of level-1 charging [83]; battery degradation [84]; and demand response [84]. V2G-Sim has also been used to examine grid-level effects of smart charging, such as smoothing the duck curve[61]. EVLib and EVLibSim were developed at Aristotle University of Thessaloniki to model many types of EV charging, including standard conductive charging, inductive charging, and battery swapping [85]. These simulators address a different problem space from ACN-Sim. While ACN-Sim is designed to evaluate online and closed-loop control strategies, these simulators only allow precomputed schedules or simple controls. ACN-Sim is also unique in modeling unbalanced, behind-the-meter electrical infrastructure, allowing it to evaluate algorithms that support oversubscribed local infrastructure.

More recently, the Open Platform for Energy Networks (OPEN) from Oxford was released to facilitate simulation and optimization of smart local energy systems, including electric vehicle charging [86]. OPEN supports model predictive control algorithms at the distribution feeder level and unbalanced three-phase infrastructure. It also allows for control of other distributed energy resources such as stationary storage and building loads. However, it has not been used to consider the electrical infrastructure behind the meter.

Despite these open-source tools, many researchers still utilize custom simulators,

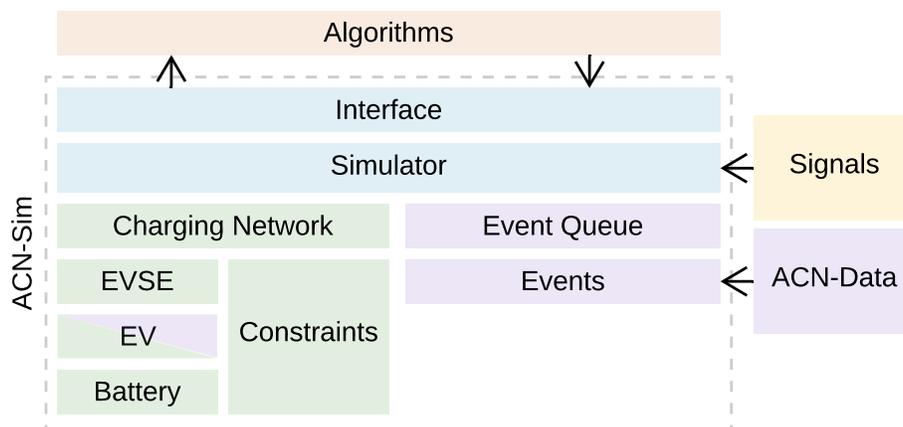


Figure 4.1: Architecture of ACN-Sim along with related sub-modules Signals, Algorithms, and ACN-Data. Note that EV models both the physical vehicle and session information, such as departure time and energy requested.

which are often simple MATLAB or Python scripts. Building a custom simulator takes time and distracts researchers from focusing on their research questions. The history of MATPOWER and other open-source projects has shown that researchers can accomplish more by using open-source tools. Moreover, one-off simulators can be error-prone. ACN-Sim has over 12,000 unit and integration tests, which help ensure that bugs are caught before they affect research results. Finally, reproducibility is key in modern research. Using an open-source simulator like ACN-Sim, researchers can easily share code and use original implementations as baselines to compare against.

Overall, ACN-Sim’s realistic models and data taken from real charging systems, along with its simple interfaces for defining new control algorithms and a suite of baseline algorithms, set it apart from existing open-source and custom simulators, making it a useful addition to the suite of tools available to researchers.

4.2 Simulator Architecture and Models

ACN-Sim utilizes a modular, object-oriented architecture which is shown in Fig. 4.1. This design models physical systems as closely as possible and makes it easier to extend the simulator for new use cases. Each of the boxes in Fig. 4.1 refers to a base class that can be extended to model new behavior or add functionality. While ACN-Sim includes several models of each component, users are free to customize the simulator to meet their needs. We encourage researchers to contribute extensions back to the project so that others can utilize them.

Simulator

A `Simulator` object forms the base of any ACN-Sim simulation. This `Simulator` holds models of the hardware components in the simulated environment and a queue of events that define when actions occur in the system. ACN-Sim is based on a discrete-time, event-based simulation model. Figure 4.2 describes its operation. During a simulation, the `Simulator` stores relevant data, such as the event history, EV history, and time series for the pilot signal and charging current for each EVSE¹, for later analysis.

Within the `Simulator` we adopt the mathematical model described in Section 2.4. We generally treat the charging network as stateless, so the state of the simulator at time k is simply the concatenation of the states of all currently charging EVs, i.e., $i \in \mathcal{V}_k$. The simulator's action space is the pilot signal for each EV, which we denote $r_i(k)$. This is an upper bound on the charging rate of the EV. This action space is constrained by the charging network's infrastructure, the EVSE's limits, the EV's maximum charging rate $\bar{r}_i(k)$, and energy requested by the EV.

After taking an action, we can observe from the environment the actual charging rate of the EV, $\hat{r}_i(k)$ and the actual energy delivered to the EV $\hat{e}_i(k)$. The state is updated according to the rule:

$$d_i(k+1) := d_i(k) - 1 \quad (4.1)$$

$$e_i(k+1) := e_i(k) - \hat{e}_i(k) \quad (4.2)$$

The maximum charging rate of the EV, $\bar{r}_i(k)$ is not observable, but can be estimated based on $\hat{r}_i(k)$. EVs can also arrive and depart at the beginning of each timestep.

Charging Network

Electrical Infrastructure.

ACN-Sim uses the `ChargingNetwork` class to model the electrical infrastructure of the charging system, including EVSEs, transformers, switch panels, and cables. Each `ChargingNetwork` instance contains a set of EVSE objects, as well as a set of constraints.

¹EVSE stands for Electric Vehicle Supply Equipment. They are more commonly known as charging stations or charging ports.

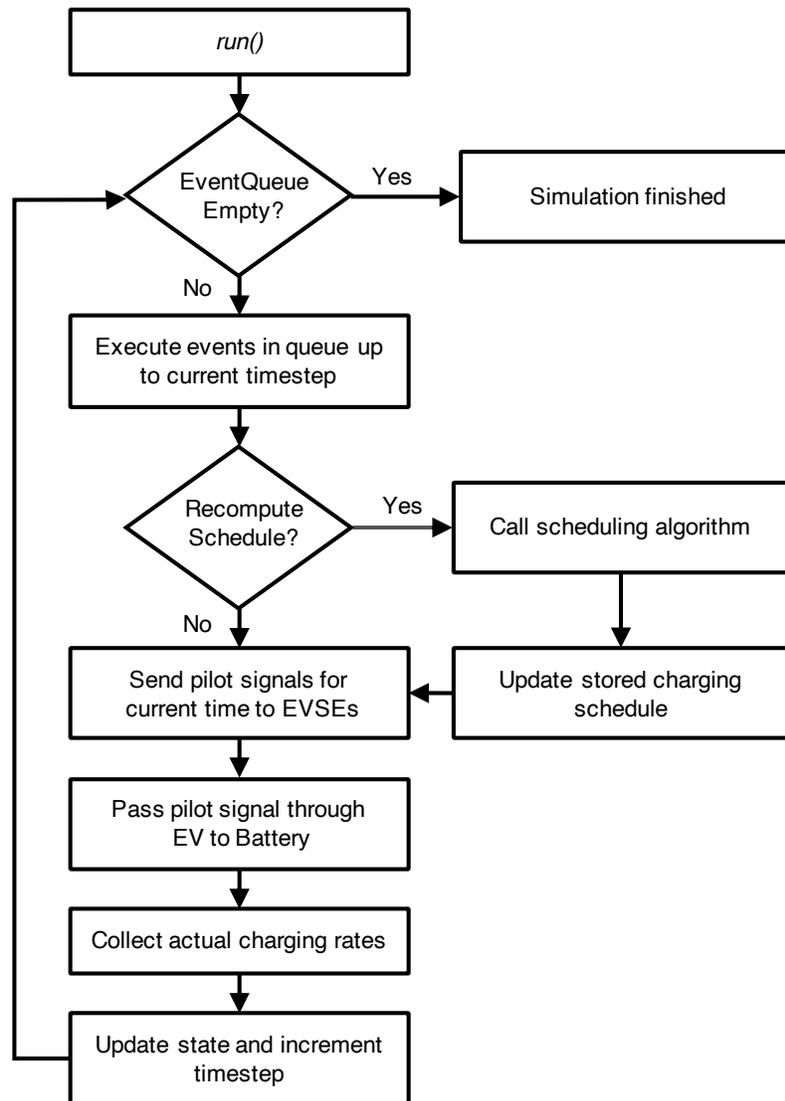


Figure 4.2: Flow chart describing the simulator's `run()` function. Each timestep consists of a single iteration of this loop. The simulation ends when the last event from the `EventQueue` is executed, at which time the user can analyze the simulation results.

We model constraints by limiting the current through each bottleneck component in the network. Because charging systems are radial networks and electrical codes specify ampacity limits that keep voltages within specifications, it is sufficient to model only constraints on current magnitudes. Recall from Chapter 2, we can express these constraints by

$$|I_l(k)| = \left| \sum_{i \in \mathcal{V}_k} A_{li} r_i(k) e^{j\phi_i} \right| \leq c_l, \quad \forall k \in \mathcal{K} \quad (4.3)$$

where $I_l(k)$ is the current through the bottleneck, c_l is the limit on the current magnitude, $r_i(k)$ is the charging current of EV i at time k . The parameter ϕ_i is the phase angle of the current phasor, which we can calculate based on how EVSE s_i is connected in the network. For simplicity, we assume ϕ_i is fixed, and voltages in the network are nominal. A_{li} can be found via circuit analysis, as shown in Example 3 for a subset of the Caltech ACN. ACN-Sim will allow (4.3) to be violated in the simulation but will raise a warning at run-time to alert the user this schedule would not be valid on a real system. This allows the user to evaluate the severity of such an overload.

To incorporate these constraints, algorithms can either parse the constraints and include them directly in the algorithm, as is done in model predictive control, or use the built-in `is_feasible()` method, which returns if the proposed charging rates are allowable under the given network model.

Stochastic Space Assignment.

`ChargingNetwork` assumes that each EV is preassigned to a specific EVSE, and no two EVs are ever assigned to the same EVSE at the same time. This holds when applying a workload from ACN-Data to its corresponding network model. However, in some cases, such as when generating events from a statistical model or applying a real workload to a new network configuration, it can be helpful to allow for non-deterministic space assignments. ACN-Sim accomplishes this through the `StochasticNetwork` class (which is a subclass of `ChargingNetwork`). Using this network model, EVs are assigned to a random open EVSE when they arrive instead of using a predefined `station_id` for assignment. Since it is possible for no EVSEs to be available when a new EV arrives, `StochasticNetwork` also includes a waiting queue for EVs which arrive while all EVSEs are in use. When an EV leaves the system, the first EV in the queue takes its place. By default, we assume

that the presence of EVs in the waiting queue does not affect drivers' departure times. However, with the `early_departure` option, drivers swap places with the first EV in the queue as soon as they finish charging.

Included Site Models.

While users are free to develop their own charging networks, ACN-Sim includes functions to generate network models that match the physical infrastructure of the seven clusters currently included in ACN-Data. In addition, the `auto_acn` function allows users to quickly build simple single-phase and three-phase networks by providing just a list of station ids and a transformer capacity. In these `auto_acn` networks, it is assumed that the transformer is the only source of constraints. These functions work with both `ChargingNetwork` and `StochasticNetwork`, which can be set as a parameter.

EVSE

As we discussed in Chapter 2, different EVSEs provide different levels of granularity in the pilot signals they support. While some EVSEs provide nearly continuous control, most are limited to a finite set of pilot signals. In addition, according to the J1772 standard, no pilot signals are allowed between 0 to 6 A [49]. In most current research, the additional constraints imposed by EVSEs without continuous control are neglected [23]. However, including these constraints is important for practical algorithms and is non-trivial.

ACN-Sim provides three EVSE models that cover most ideal and practical level-2 EVSEs:

- `EVSE` allows any pilot signal between an upper and lower bound. By default, `EVSE` allows any non-negative charging rate.
- `DeadbandEVSE` also allows continuous pilots but excludes 0 - 6 A as required by the J1772 standard.
- `FiniteRatesEVSE` only allows pilot signals within a finite set, accurately modeling most commercial EVSEs. For example, many of the EVSEs used in the Caltech ACN allow {6, 7, ..., 31, 32} or {8, 16, 24, 32} amps.

Within ACN-Sim, EVSE is also the interface between the charging network and an EV. When an EV plugs into the system, a reference to that EV is added to the corresponding EVSE. When it is time to update the pilot to an EV, the Simulator first passes the pilot to the EVSE, which in turn passes it on the EV and eventually the Battery. This mimics the flow of information in a real charging system. Similarly, when an EV leaves the system, the reference to that EV is removed from the EVSE.

EV

The EV object contains relevant information for a single charging session, such as arrival time, departure time, estimated departure time, and requested energy. The estimated departure time may differ from the actual departure time. Likewise, it may be infeasible to deliver the requested energy in the allotted time due to maximum charging rate restrictions, system congestion, or insufficient battery capacity. By allowing this, ACN-Sim models the case where user inputs or predictions are inaccurate, which is common in practice.

Battery

Most EV charging research utilizes an ideal battery model, where EVs are assumed to follow the given pilot signal exactly. However, in practice, we see that the charging rate is often strictly lower than the pilot signal and decays as the battery approaches 100% state-of-charge [23]. This can significantly increase the total time required to charge the battery and results in under-utilization of infrastructure capacity.

ACN-Sim jointly models the vehicle's battery and battery management system. The battery's actual charging rate depends on the pilot signal and the vehicle's onboard charger, state-of-charge, and other environmental factors. ACN-Sim currently includes two battery models.

The Battery class is an idealized model and serves as the base for all other battery models. The actual charging rate of the battery, $\hat{r}(k)$, in this idealized model is described by

$$\hat{r}_i(k) := \min\{r_i(k), \bar{r}_i(k), \bar{e}_i(k)\}$$

where $r_i(k)$ is the pilot signal passed to the battery, $\bar{r}_i(k)$ is the maximum charging rate of the on-board charger, and $\bar{e}_i(k)$ is the difference between the capacity of the battery and the energy stored in it at time k in the units of A-periods. We do not consider discharging batteries, so all rates are positive.

`Linear2StageBattery` is an extension of `Battery` that approximates the roughly piecewise linear charging process used for lithium-ion batteries, often referred to as Constant Current - Constant Voltage (CC-CV) charging. The first stage, referred to as *bulk charging*, typically lasts from 0% to between 70 to 90% state-of-charge. During this stage, the current draw, neglecting changes in the pilot, is nearly constant. In the second stage, called *absorption*, the battery's voltage is held constant while the charging current decreases roughly linearly. The actual charging rate of the `Linear2StageBattery` is given by

$$\hat{r}_i(k) := \begin{cases} \min\{r_i(k), \bar{r}_i(k), \bar{e}_i(k)\} & \text{if } \text{SoC}_i \leq \text{th} \\ \min\left\{(1 - \text{SoC}_i) \frac{\bar{r}_i(k)}{1 - \text{th}}, r_i(k)\right\} & \text{otherwise} \end{cases}$$

where SoC_i is the state-of-charge of the battery and `th` marks the transition from the *bulk* stage to the *absorption* stage of the charging process. Figure 4.3 shows how these two models compare for two charging profiles taken from ACN-Data.

We find that while the piecewise linear model is a good approximation, it does not capture all the battery/BMS behaviors we observe in practice (as in the right panel of Fig. 4.3). For many experiments, the `Linear2Stage` model is sufficient. However, for evaluating the interactions between battery management systems and smart charging or studying the effect of smart charging on battery aging, more advanced models may be needed. ACN-Sim's modular architecture allows new battery/BMS models to be easily implemented as subclasses of `Battery`. By doing this, a researcher can implement various battery/BMS systems such as electrochemical models [50], electro-thermal and aging models [51], or manufacturer specification (spec) based models [52], [53].

Event Queue / Events

ACN-Sim uses events to describe actions in the simulation. There are two types of events currently supported:

- `PluginEvent` signals when a new EV arrives at the system. A `PluginEvent` also contains a reference to the EV object which represents the new session.
- `UnplugEvent` signals when an EV leaves the system at the end of its charging session.

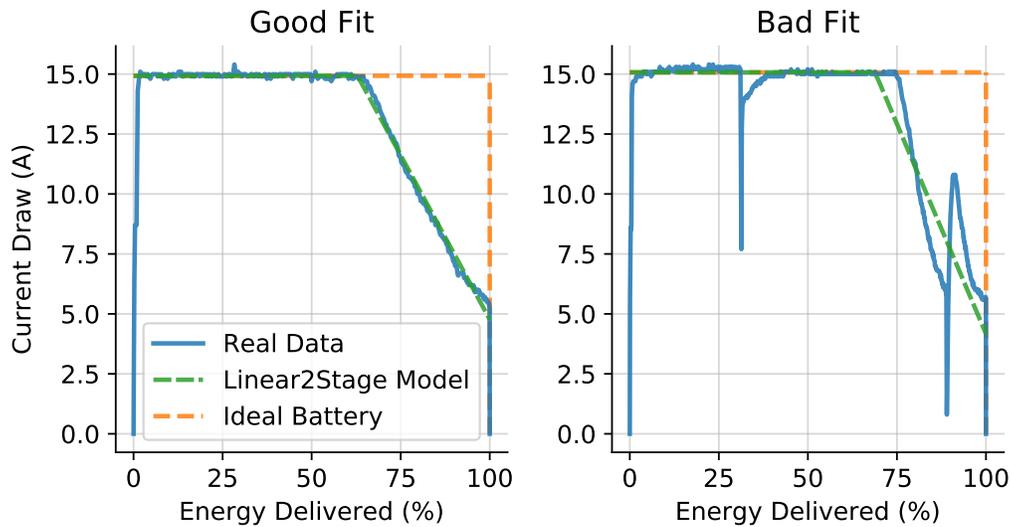


Figure 4.3: Comparison of `Linear2Stage` and idealized `Battery` models with a real charging curve collected from two distinct users of the Caltech ACN when the pilot signal is not binding. We can see that the `Linear2Stage` model with appropriate parameters matches the battery behavior well in the first case. However, in the second case, there are dynamics in the joint battery/battery management system that the `Linear2Stage` model does not capture, namely as the double-tail behavior (which is recurring for this user).

Each event has a timestamp describing when the event should occur. Events are stored in a queue sorted by their timestamp. Since multiple events could occur at the same timestep, we further sort by event type, first executing `UnplugEvents`, then `PluginEvents`. At each timestep, the `Simulator` executes all events left in the queue with timestamps on or before the current timestep. After any event, the scheduling algorithm is called to adapt to the new system state. Users are free to create new events by extending the `Event` class.

To generate events, users can either get real event sequences from `ACN-Data`, generate event sequences from statistical models, or manually create events to investigate edge cases. To make accessing `ACN-Data` simpler for users, `ACN-Sim` provides direct integration with the `ACN-Data` API. This allows the user to specify a site and date range, and `ACN-Sim` will gather the actual workload from that `ACN` and generate the appropriate `PluginEvents` and `UnplugEvents`. `ACN-Sim` also provides utilities for learning statistical models such as `Gaussian Mixture Models`, directly from data using tools from `scikit-learn`. These techniques are described in Chapter 8.

Users could also generate events through co-simulation with the transportation or power network. For example, a `PluginEvent` might be generated by a transportation

model that calculates user's arrival time and energy demands based on their travel patterns. Likewise, a power network simulator might generate a demand response event when a distribution transformer's loading becomes too high. These are not included in the current release of ACN-Sim but are planned as future work.

The above models for generating `PluginEvents` and `UnplugEvents` assume that driver behavior is fixed. In reality, charging operators interact with drivers who may strategically respond to scheduling algorithms or pricing schemes by adjusting their arrival time, departure time, or energy request. While not currently included, we plan to incorporate strategic driver models in a future release of ACN-Sim.

Signals

The signals sub-module allows ACN-Sim to integrate with external signal sources, which can be an important part of EV charging systems such as 1) utility tariffs, 2) solar generation curves, 3) external loads.

Utility Tariffs.

Operating costs are an important concern for EV charging facilities. To support utility tariffs, ACN-Sim includes the `TimeOfUseTariff` class, which supports time-varying and seasonal tariff schedules with or without demand charges. To make integration easier for users, ACN-Sim includes several utility tariff schedules. Users can define new schedules in a simple JSON format. This functionality allows users to investigate cost minimization strategies and accurately estimate the operating costs of charging system designs under different tariff structures. In Section 8.3, we use these tariff schedules to calculate operating costs, and in Section 6.2, we provide the tariff as an input to the minimum cost objective for the MPC algorithm.

Solar Generation.

As many sites with EV charging also have on-site solar generation, studying the behavior of an EV charging facility that takes solar generation into account is an important use case. ACN-Sim allows users to input a solar generation signal as a CSV file to the Simulator. Solar data can be user-generated, downloaded from an external source, or generated by an external solar generation simulator such as NREL's system advisor model (SAM)[87]. Such functionality allows users to study

the effects of on-site solar on cost, energy demand, grid loading, and other metrics associated with large-scale EV charging.

External Load.

EV charging facilities often share a meter with other loads, such as the buildings on a university or corporate campus. To reduce demand charge and stress on the grid, it can be advantageous to consider these other loads when scheduling EV charging. To facilitate the study of algorithms that do this, ACN-Sim allows users to input an external load profile as a CSV file. External load data can be user-generated or downloaded from an external source.

Other Signals.

Other signals such as pollution indexes or demand response profiles can be loaded into the simulator using the `signals` dictionary within the `Simulator` constructor or passed directly to the control algorithm.

Co-simulation with Grid Simulators

ACN-Sim also provides co-simulation with popular grid-level simulators, including MATPOWER, PandaPower, and OpenDSS. This allows researchers to investigate Vehicle-Grid Integration (VGI) problems such as algorithms to alleviate voltage and overload issues in the local distribution system or aggregation approaches to bid into markets. In the current version, simulations are run sequentially, with the output of the ACN-Sim experiment serving as an input to the grid simulator. In future releases, we plan to support feedback from the grid simulation into ACN-Sim.

OpenAI Gym Integration

Reinforcement learning (RL) has long been applied to scheduling and resource allocation problems [88]. Smart EV charging is a particularly interesting application for RL as it involves a complex and uncertain environment with large state and action spaces and safety-critical constraints. To help researchers apply new and existing RL algorithms to problems in EV charging, we have integrated ACN-Sim with OpenAI's Gym package[89]. Gym uses a standard `Environment` interface to make it easy for researchers to apply RL algorithms to problems ranging from video games to robotics.

To integrate ACN-Sim with Gym we implement this `Environment` interface to wrap an ACN-Sim Simulator. The `gym-acnportal` package also provides an expanded `Interface` object (see Section 4.3), which allows the simulation to proceed one time-step at a time during training. At each step, the agent receives a state, which is a partial observation of the environment. This consists of the concatenation of each EV’s parameters, as described in Section 4.2, along with the index of the timestep, k , and network constraints C , where

$$C := (A_{li}, \phi_i, c_l) \quad \forall l \in \mathcal{L}, \forall i \in \mathcal{V}$$

The agent also receives a reward from its previous action. Users can customize this reward to the particular objective of the agent. For example, a simple reward for an agent whose only objective is to charge vehicles without violating infrastructure constraints might be

$$R(k) := u^{ED} - \eta u^{CV} \quad (4.4a)$$

where

$$u^{ED}(k) := \sum_{i \in \mathcal{V}} e_i(k) - e_i(k-1) \quad (4.4b)$$

$$u^{CV}(r, k) := \max \left(0, \sum_{l \in \mathcal{L}} \left| \sum_{i \in \mathcal{V}} A_{li} r_i(k) e^{j\phi_i} \right| - c_l \right)^2 \quad (4.4c)$$

Here (4.4b) rewards the agent for the energy it delivered in time step k , while (4.4c) penalizes the agent for any constraint violation. We use the coefficient η to adjust the magnitude of this penalty.

The agent then calculates an action, which here is a charging rate for each EV. To keep the size of the state and action spaces consistent, we pad the state with additional triplets of 0’s for each EVSE without an EV plugged in, such that the number of $(e_i(k), d_i(k), \bar{r}_i(k))$ triplets is always equal to the number of EVSEs in the network. We also pad schedules (actions) with 0’s for each EVSE without an EV. This action is then passed to the `Environment`’s step function, and the process repeats. The reset function allows the environment to return to a known state after a training episode. After training an RL agent, the `gym-acnportal` package also provides a wrapper that allows the agent to be deployed as an ACN-Sim algorithm using the standard `Interface`.

In addition to customizing the reward function, researchers can also adjust the environment’s state and action space. For example, in [69] the action space is a set

of parameters for an additive-increase, multiplicative-decrease (AIMD) algorithm. Additions to the state space might include congestion metrics, prices, or renewable generation forecasts.

4.3 Charging Algorithms

Interface

To make algorithm implementations more flexible, we introduce an interface that abstracts away the underlying infrastructure, whether that be simulated or real, allowing us to use the same algorithm implementation with both ACN-Sim and ACN-Live. This means that users can thoroughly test algorithms with ACN-Sim before testing on physical hardware. It also means algorithms developed to work with ACN-Sim can work with other platforms simply by extending the `Interface` class.

Defining an algorithm

To define an algorithm in ACN-Sim users only needs to extend the `BaseAlgorithm` class and define the `schedule()` function. This function takes in a list of active sessions, meaning that the EV is plugged in and its energy demand has not been met and returns a charging schedule for each. This schedule is a dictionary that maps `station_id` to a list of charging rates in amps. Each entry in the schedule is valid for one timestep beginning at the current time. Algorithms have access to additional information about the simulation through the `Interface` class, such as the current timestep, infrastructure constraints, and allowable pilot signals for each EVSE.

Included algorithms

ACN-Sim is packaged with many common online scheduling algorithms that can be used as benchmarks.

Uncontrolled Charging.

Most charging systems today do not manage charging. With Uncontrolled Charging, each EV charges at its maximum allowable rate. This algorithm does not factor in infrastructure constraints, so they may be violated.

Round Robin.

Round Robin (RR) is a simple algorithm that attempts to share charging capacity equally. It creates a queue of all active EVs. For each EV in the queue, it checks if it is feasible to increment its charging rate by one unit. If it is, it increments the rate and replaces the EV at the end of the queue. If it is not, the charging rate of the EV is fixed, and the algorithm does not return the EV to the queue. This continues until the queue of EVs is empty. In this context, a feasible charging rate is one that does not cause an infrastructure constraint to be violated and is less than the maximum charging rate, $\bar{r}_i(k)$, and the energy demand, $e_i(k)$, of the EV.

Sorting Based Algorithms.

Sorting based algorithms are commonly used in other deadline scheduling tasks such as job scheduling in servers due to their simplicity [27]. ACN-Sim includes several of these algorithms, including First-Come First-Served (FCFS), Last-Come First-Served (LCFS), Earliest-Deadline First (EDF), Longest Remaining Processing Time (LRPT), and Least-Laxity First (LLF). These algorithms work by first sorting the active EVs by the given metric, then processing them in order. Each EV is assigned its maximum feasible charging rate, which is calculated using a bisection algorithm, given that the assignments to all previous EVs are fixed. This process continues until all EVs have been processed. For any algorithm which uses departure time, e.g., EDF and LLF, estimated departure time is used. The researcher is left to decide the accuracy of the estimated departure time.

Model Predictive Control.

Many approaches to the EV scheduling problem rely on model predictive control (MPC). In Chapter 6, we will present the Adaptive Scheduling Algorithm (ASA), which is one example of an MPC algorithm for managed EV charging applications.

An open-source implementation of the ASA algorithm is available in the `adacharge` package[90]. This package uses using CVXPY[91], [92]—a popular modeling language for convex optimization problems. With this package, users can easily choose from existing objective functions and constraints or create their own.

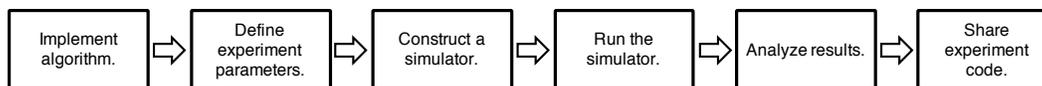


Figure 4.4: Flowchart describing the common research pipeline used when evaluating an algorithm through simulation. ACN-Sim helps to accelerate this process by simplifying and automating as much of this pipeline as possible.

4.4 Simplified Research Pipeline

ACN-Sim is designed to simplify the process of evaluating new algorithms for smart EV charging. We consider a researcher who wants to simulate their algorithm’s performance against other algorithms proposed in the literature. Their workflow can be broken into the six steps shown in Fig. 4.4. We will now demonstrate how ACN-Sim simplifies each step in this process.

Implementing an algorithm

As described in Section 4.3, ACN-Sim greatly simplifies the process of implementing an algorithm. In addition, since researchers can easily share their algorithms, others do not need to re-implement baseline algorithms to compare against.

Defining an experiment

Without ACN-Sim, defining an experiment can be difficult, as researchers need to model the charging system they want to use and generate a realistic set of scenarios. ACN-Sim simplifies this process by providing predefined charging networks. It also integrates with ACN-Data to allow researchers to generate scenarios from real charging data. With these utilities, the researchers only need to define the start and end times of the simulation, what site they want to simulate, and what algorithm(s) they want to test.

Constructing and running a simulation

After an experiment has been defined, the process of constructing and running a simulation is as simple as constructing a `Simulator` object then invoking its `run()` method, which is described in Section 4.2. ACN-Sim allows the researchers to achieve this task with only a few lines of code versus the thousands necessary to build a realistic simulator from scratch.

Analyzing results

After a simulation, researchers can analyze the results using data stored in the `Simulator`. This data includes the pilot signals sent to each EVSE and the actual charging rate of the EV attached. The `Simulator` can also save the results of each call to the scheduling algorithm for more in-depth analysis. To make analysis even easier, ACN-Sim includes tools to perform common calculations such as finding aggregate currents, measuring phase imbalances, and calculating the proportion of energy demands met.

Sharing experiment

Reproducibility is key to good science. However, when experiments are run on in-house simulators and closed datasets, it can be difficult for researchers to share their work and verify the results of others. With ACN-Data and ACN-Sim, all researchers have access to the same dataset and simulation environment, making it much easier to share experiments and benchmark algorithms.

To make this process even easier, ACN-Sim works with Google Colab, a free cloud-based service that hosts Jupyter notebooks [93]. Using this service, the researchers can share links to their experiments which can be run in the cloud without downloading or installing software locally. Other researchers can then easily try the experiment themselves, even changing parameters or trying different scenarios.

Moreover, once the algorithm has been implemented to work with the ACN-Sim framework, it can be easily shared by contributing it to the `algorithms` module or sharing the code separately. Other researchers can then directly test against the original authors' implementation of the algorithm.

4.5 Concluding Remarks

This chapter presents ACN-Sim, a data-driven simulator designed to aid in developing practical online scheduling algorithms for EV charging. ACN-Sim significantly reduces the software engineering burden on researchers and exposes them to practical issues present in real charging systems. ACN-Sim also makes it easier for researchers to share their experiment code, improving transparency and code reuse in the community. Finally, ACN-Sim integrates with the Adaptive Charging Network Research Portal, a larger suite of tools that includes a database of real charging

sessions and a framework for field testing algorithms. ACN-Sim will continue to grow to meet the community's needs, including new models of systems components and charging networks.

Areas of future expansion for ACN-Sim include support for co-located DERs, dynamic pricing, access control, co-simulation with the transportation network, tighter integration with grid simulators to investigate providing grid services, and adding physics-based battery models. We also plan to develop a suite of test cases (networks and scenarios) that can be used as a standard evaluation set for algorithms.

Published Works

This chapter contains text and figures originally published in:

Z. J. Lee, S. Sharma, D. Johansson, and S. H. Low, "ACN-Sim: An Open-Source Simulator for Data-Driven Electric Vehicle Charging Research," arXiv:2012.02809 [cs, eess], Dec. 2020, arXiv: 2012.02809. [Online]. Available: <http://arxiv.org/abs/2012.02809>

Z. J. Lee, D. Johansson, and S. H. Low, "ACN-Sim: An open-source simulator for data-driven electric vehicle charging research," in 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), IEEE, 2019, pp. 1–6. DOI: 10.1109/SmartGridComm.2019.8909765. © IEEE 2019

Chapter 5

ACN-LIVE

Thus far, we have discussed how ACN-Data and ACN-Sim help researchers apply data-driven methods to EV charging research. While these tools help bridge the gap between theoretical work and algorithms that could be deployed in practice, they are not substitutes for field tests and pilots. These field tests are important to prove to stakeholders like utilities, funding agencies, policymakers, and consumers the viability of new technologies. However, field tests and pilots in the energy space are rare, as they require vast amounts of time, funding, and expertise. For example, the ACN system has required over five years of work and millions of dollars of funding to reach its current state. Because of these challenges, only a handful of researchers have access to systems like the ACN.

This lack of access hampers research progress and technology transfer into the marketplace. To meet this need, we have designed ACN-Live, a framework for field testing algorithms on the Caltech Adaptive Charging Network. ACN-Live allows researchers who have thoroughly tested their algorithms with ACN-Sim to deploy them on the physical ACN. By utilizing the same interface as ACN-Sim, we enable researchers to perform field tests with no changes to their algorithm implementation. This is a unique opportunity, which requires the specialized hardware of the ACN and close collaboration between our group, Caltech facilities, and PowerFlex. As such, we are likely the only facility in the world able to provide this type of hardware in the loop testing to the research community.

This chapter focuses on the design and architecture of ACN-Live. In our design, we emphasize reducing the burden on researchers by introducing abstractions. These abstractions allow the researcher to write an algorithm once and then deploy that code in simulation and real systems. We also ensure safety and reliability in the physical system by implementing real-time checks on proposed control actions. If these checks fail, we seamlessly fall back to a business-as-usual algorithm. We also provide researchers with data after their test to enable analysis, including session and time series data using the same software and format used in ACN-Data. As of May 2021, ACN-Live has not been released externally. However, we intend to deploy it to beta testers by the end of 2021.

5.1 Design Goals

Four broad goals drive the design of ACN-Live.

Ease of Use

For any research tool, ease of use is an important factor. If the tool is too complex, researchers will not use it. In the case of ACN-Live, our goal was to make deploying an algorithm on physical hardware as easy as running a simulation using ACN-Sim. Algorithms interact with the underlying system, whether simulated or real, through an interface that abstracts away the underlying implementation details of gathering data. This means that researchers do not need to modify their algorithm implementation between testing with ACN-Sim and ACN-Live.

Safety and Reliability

Whenever dealing with physical hardware, safety must be a primary concern. While ACN-Sim allows researchers to violate safety constraints (with a warning), ACN-Live *cannot* allow infrastructure constraints to be violated. While circuit breakers and other protection devices in the system are designed to prevent long-term damage, tripping a breaker will cause one or more vehicles to lose power, which is unacceptable. If ACN-Live receives a charging schedule that violates safety constraints, the test will terminate, and the ACN will return to a business-as-usual algorithm that is known to be safe.

Let $r(k)$ be the charging rates for time k proposed by algorithm A_1 . These rates are *safe* if-and-only-if $r(k) \in \mathcal{S}$, where \mathcal{S} is defined as:

$$r_i(k) \in \rho_i(k) \quad i \in \mathcal{V} \quad (5.1a)$$

$$\left| \sum_{i \in \mathcal{V}} A_{li} r_i(k) e^{j\phi_i} \right| \leq c_l(t) \quad l \in \mathcal{L} \quad (5.1b)$$

Constraints (5.1a) ensure that the charging rate is within the set of pilot signals supported by the EVSE, while (5.1b) ensure that no infrastructure constraints are violated. We require this to be true for all k . See Chapter 2 for further discussion of these constraints.

We must also consider the constraints of drivers in the system. Drivers and site hosts expect that their energy needs will still be met, even when running a field test of a new algorithm.

In particular, we require that our system maintains ϵ -reliability. A given algorithm, \mathcal{A}_1 , is defined to be ϵ -reliable when compared with a baseline algorithm, \mathcal{A}_0 , if-and-only-if for each EV in the system, \mathcal{A}_1 delivers at least ϵ times the energy delivered by \mathcal{A}_0 , for some $\epsilon \leq 1$. More formally, let $\hat{e}_i^{(1)}(d_i)$ be the energy delivered to EV i by its departure time under \mathcal{A}_1 , and likewise $\hat{e}_i^{(0)}(d_i)$ for \mathcal{A}_0 . Then for all EVs in the system, \mathcal{V} , we require

$$\hat{e}_i^{(1)}(d_i) \geq \epsilon \hat{e}_i^{(0)}(d_i) \quad \forall i \in \mathcal{V} \quad (5.2)$$

Unfortunately, we cannot check for ϵ -reliability in real-time, since we are operating with limited information about the future. One way to address this is to evaluate if an algorithm is ϵ -reliable in simulation. However, we would also like to provide reliability checks in real-time. For this, we introduce online ϵ -reliability. Let $r^{(0)}$ be the schedule of charging rates produced by algorithm \mathcal{A}_0 , and likewise $r^{(1)}$ for \mathcal{A}_1 . We then require that the schedule produced at each timestep satisfies:

$$\sum_{t=0}^{d_i} r^{(1)}(t) \geq \epsilon \sum_{t=0}^{d_i} r^{(0)}(t) \quad \forall i \in \mathcal{V} \quad (5.3)$$

This definition assumes that \mathcal{A}_0 and \mathcal{A}_1 produce a full charging trajectory. This is true for \mathcal{A}_0 , since we use a model predictive control algorithm. However, some algorithms, such as LLF and EDF, only produce a single charging rate for each EV. To ensure that a recourse action exists for these algorithms, we project what the state of the system will be in the next timestep using (4.2) and (4.1). We then produce the rest of the charging trajectory by calling \mathcal{A}_0 on this projected state. This ensures that if the system falls back to \mathcal{A}_0 in the next iteration, we still achieve online ϵ -reliability.

Comprehensive Data Collection

To be useful, a field test or pilot should collect all relevant data for analysis. Within ACN-Live, we leverage the data collection and reporting functionality of the ACN and ACN-Data. To enable debugging, ACN-Live also logs the messages (inputs) sent to the researcher's algorithm along with the algorithm's responses.

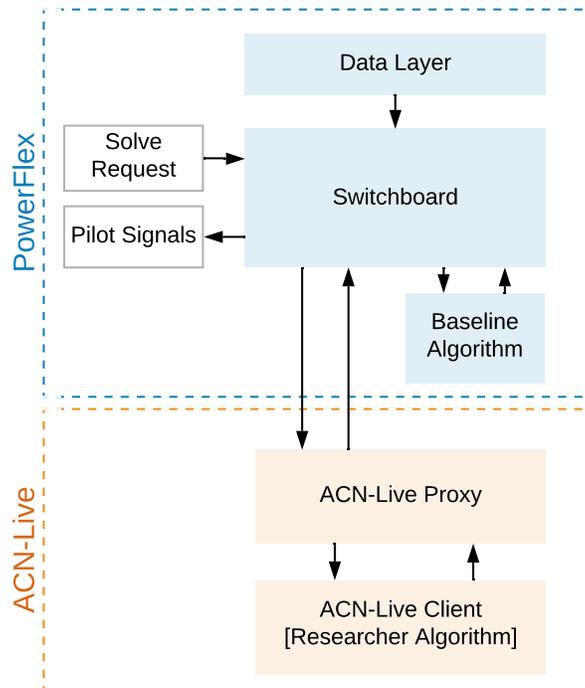


Figure 5.1: Architecture of the ACN-Live. Blue boxes (top) denote the PowerFlex system. Orange boxes (bottom) denote the ACN-Live components.

Iterative Feedback

By collecting this data, ACN-Live enables iterative feedback for algorithm debugging and tuning. Researchers can improve their algorithms, evaluate these improvements in simulation, test on the real system, then repeat the process to improve their algorithm further. If a real-world test uncovers an unexpected behavior, researchers can use the logged inputs and outputs of the algorithm to understand the behavior or debug the issue. If this results in modifications to the algorithm, researchers should be able to test these changes in ACN-Sim using the same workload data as they experienced during the field test.

5.2 System Architecture

Overview

An overview of the ACN-Live architecture is shown in Fig. 5.1. The operations of ACN-Live are summarized in the Swimlane Diagram in Fig. 5.2.

The ACN-Live System is made up of two parts. The Proxy is an intermediary between the PowerFlex system and ACN-Live, which runs in the cloud. It takes

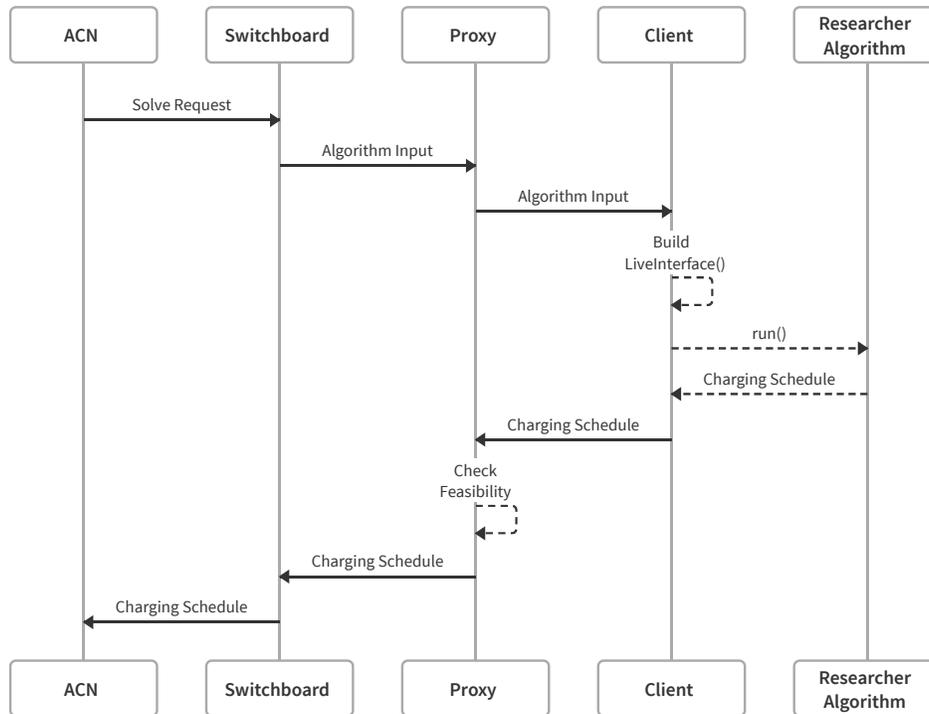


Figure 5.2: Swimlane diagram describing the operations of ACN-Live *assuming* that the researcher’s algorithm is feasible. For clarity, this diagram *omits* the baseline algorithm, which would be used if the researcher’s algorithm were not feasible. This baseline algorithm runs in parallel with the researcher’s algorithm to avoid delays.

requests from the PowerFlex Switchboard and distributes those requests to Clients via a messaging queue. A Client can run on the researcher’s computer or in the cloud. The Client accepts work from the message queue, calculates new charging schedules using the researcher’s algorithm, and returns that schedule to the Proxy. The Proxy then passes the schedule back to Switchboard if it is valid. If the schedule fails either the safety or ϵ -reliability tests, Proxy will instead return an error, in which case Switchboard will fall back to a baseline algorithm.

Message Queue (NATS)

Within ACN-Live and the Adaptive Charging Network, we use NATS as a message queue to communicate between modules [94]. NATS is an open-source message queue written in Go. It supports a publish-subscribe messaging model. (Other messaging models are also supported but not relevant in this case.) A key benefit of using a message queue approach like NATS over a 1-to-1 model like HTTP or remote procedure calls (RPCs) is that publishers do not need to be aware of the address (or existence) of the subscribers of their messages. This makes it easy for

new services to join the system with minimal configuration. In addition, NATS supports one-to-many messaging, which is helpful when distributing parameters to many algorithms in parallel.

Switchboard

Switchboard is a multiplexer that allows the ACN to run multiple scheduling algorithms in parallel, then choose the highest precedence schedule returned. This enables us to run potentially unreliable or long-running algorithms, then seamlessly fall back to baseline algorithms if those algorithms fail or timeout.

Switchboard accepts schedule requests over a message queue. These requests can be generated periodically or by events in the system. See Chapter 5 for a description of the ACN’s event-driven architecture. Switchboard then gathers information about the system state from the Data Layer. It then publishes this state in a message to the `algorithm_input` subject. All algorithms in the system subscribe to this subject. These algorithms then calculate a schedule and return it and their *precedence* value as a message to a reply subject. NATS automatically includes this one-time-use reply subject in the original message. Using a reply subject ensures that this queue will only include messages in response to this particular input. After a set time limit, Switchboard will take the schedule from the reply queue for this input with the highest precedence value and pass this schedule back to the rest of the system.

ACN-Live Proxy

The ACN-Live Proxy is the bridge between ACN-Live and the PowerFlex system. It is a subscriber to the `algorithm_input` channel of the PowerFlex NATS server, so from Switchboard’s perspective, it is just another scheduling algorithm. However, Proxy also hosts its own NATS server, which is accessible by ACN-Live Clients. When Proxy receives a message on the `algorithm_input` subject, it copies that message to the `remote_algorithm` subject on the ACN-Live NATS server.

When the Proxy receives a reply from the researcher’s algorithm, it first checks that the schedule is *safe*, i.e., $r(0) \in \mathcal{S}$. If the schedule is safe, it next checks that it is ϵ -reliable. We use the ASA algorithm with the quick charge objective (see Chapter 6) as the baseline to determine the counterfactual energy delivered $e_i^0(d_i)$.

If the schedule passes both of these tests, the Proxy will package the results and return them to Switchboard. If the schedule fails either of these tests, the Proxy

returns an error to Switchboard so that it will instead use the algorithm with the next highest precedence.

ACN-Live Client

The ACN-Live Client is the interface between a researcher's algorithm code and ACN-Live. To set up the Client, users must provide a valid token, a URL pointing to the ACN-Live NATS server, and a scheduling algorithm that extends `BaseAlgorithm`. The researcher then calls the `Client.run()` method. Behind the scenes, the Client will connect to the NATS cluster and subscribe to the `remote_algorithm` subject. When a message is received on this queue, the Client uses the information in the message to build a `LiveInterface` object, which implements the same interface as the ACN-Sim `Interface` class. The Client then registers this interface with the researcher's algorithm and runs the algorithm. Finally, the Client packages the results and publishes them to the reply channel.

5.3 Using ACN-Live

Testing with ACN-Sim

The first step in using ACN-Live is to develop an algorithm and test that algorithm in ACN-Sim. In Chapter 4, we describe how to implement an algorithm as a subclass of `BaseAlgorithm`. Using ACN-Sim, researchers can quickly evaluate if their algorithm is safe (ACN-Sim will raise warnings if it is not) and reliable (ACN-Sim provides utilities for checking what percentage of user energy demands were met). ACN-Sim also provides a useful testbed for tuning parameters and finding bugs in algorithm implementations.

Once an algorithm has been fully tested in ACN-Sim, it is ready to be considered for ACN-Live.

Running an Algorithm on ACN-Live

To run an algorithm on ACN-Live, researchers must first submit a request to the ACN Research Portal team, including a link to their algorithm code. The team will run independent tests using ACN-Sim to ensure that the algorithm is safe and robust. If the algorithm passes these tests, researchers will be issued a time slot for their field test and a token to authenticate their client against the NATS server.

Gathering Data

During a test, researchers can monitor live data using the PowerFlex web portal, i.e., `caltech.powerflex.com`. After the test, ACN-Live uses tools from ACN-Data to collect the results of the test, including session information and time series. This allows researchers to easily evaluate their results. In addition, researchers can use the session information to re-run this scenario in ACN-Sim. This is helpful, for example, if the researchers are curious about counter-factual scenarios, like what the cost of charging would have been if another algorithm had been used.

In addition, as the test runs, the ACN-Live Client will store a log of all NATS messages it sends and receives. This helps the researcher debug their algorithm and understand surprising behavior that comes up during the test. These messages can then be analyzed and modified to see how it changes the algorithm's output. Alternatively, researchers can run the same messages through a modified algorithm to see if they have fixed a bug or improved behavior.

5.4 Concluding Remarks

ACN-Live will be a unique tool for researchers to evaluate algorithms on real hardware. Without ACN-Live, few researchers would be able to do this, preventing potentially transformational ideas from being deployed in the real world. While this chapter has covered the design and architecture of ACN-Live, the software for the system is currently being developed and tested for deployment in late 2021.

Part III

Algorithms & Applications

Chapter 6

ADAPTIVE SCHEDULING ALGORITHMS

Armed with these tools, we can begin to develop practical algorithms for large-scale, smart EV charging. As we saw in Chapter 1, there is a huge literature on charging algorithms. However, while building out the ACN, it became clear that most existing algorithms could not be applied directly, as they made strong assumptions that did not hold in practice or lacked the flexibility to incorporate practical constraints and objectives.

In particular, we identified a list of properties that are needed for practical smart charging algorithms.

1. Algorithms should incorporate user preferences around their departure time and the amount of energy their vehicle needs.
2. Algorithms should handle multiple levels of unbalanced infrastructure constraints.
3. Algorithms should handle multiple objectives easily.
4. Algorithms should promote desirable properties in the final schedule, such as smooth charging profiles or fairly sharing capacity amongst EVs.
5. Algorithms should be robust against uncertainty in future arrivals and session parameters.
6. Algorithms should handle discrete set-points for pilot signals.
7. Algorithms should be able to reclaim idle capacity if vehicles do not utilize the full pilot signal.

As we examined the literature, no algorithms met all of these requirements. Instead, we developed a new algorithm that has these properties. We term this algorithm the Adaptive Scheduling Algorithm (ASA). This chapter focuses on the design and evaluation of ASA.

In Section 6.1, we describe the design of the Adaptive Scheduling Algorithm. ASA is based on model predictive control (MPC), a popular framework for solving

constrained control problems. Using this framework, we can express limits on the state and action spaces as constraints (Properties 1 and 2). We can also design the objective function to incorporate many operator objectives, such as minimizing cost or charging quickly (Property 3). We introduce a collection of regularizers that promote desirable properties in the final schedule, such as fairness or smoothing (Property 4).

Using MPC also grants us a level of robustness to uncertainty about the future, Property 5. We solve our optimization problem in each time step and only apply the action for the first step of the optimization horizon. Within each optimization, we assume that user inputs are accurate and assume no future arrivals. However, we can update this information in each iteration. While not included in this thesis, methods for further incorporating statistical models into MPC algorithms for EV charging have been proposed [33], and could be incorporated into ASA with only minor modifications.

While it would be possible to directly encode Property 6 into the constraints of the MPC optimization, doing so would make the problem non-convex and potentially intractable for real-time use. Instead, we propose a post-processing heuristic that works well in practice. To give us Property 7, we propose an algorithm called ramp-down to reclaim idle capacity from EVs that are not using their full allocation. The most common reason for this is the vehicle’s acceptance rate decreasing as the battery fills up, as we saw in Fig. 4.3.

In Section 6.2, we use ACN-Data and ACN-Sim to evaluate ASA against baseline algorithms using trace-driven simulations. Using these simulations, we see that ASA can deliver more energy with less infrastructure than baseline algorithms, even when considering practical constraints. We also find that ASA can improve operator profits by 340% compared to unmanaged charging. Finally, in Section 6.3, we show that ASA with a load flattening objective can be used to reduce the strain that large charging systems place on the distribution grid. We find that ASA can eliminate transformer overloads and voltage issues caused by uncontrolled charging.

6.1 Online Scheduling Framework

Model predictive control

The ACN computes charging rates using model predictive control, described in Alg. 1. In line 1 we compute the active EV set \mathcal{V}_k by looking for all EVs currently plugged

Algorithm 1: Adaptive Scheduling Algorithm (ASA)

```

for  $k \in \mathcal{K}$  do
(1)    $\mathcal{V}_k := \{i \in \hat{\mathcal{V}}_k \mid e_i(k) > 0 \text{ AND } d_i(k) > 0\}$ 
(2)   if event fired OR time since last computation  $> P$  then
(3)      $(r_i^*(1), \dots, r_i^*(T), i \in \mathcal{V}_k) := \mathbf{OPT}(\mathcal{V}_k, U_k, \mathcal{R}_k)$ 
(4)      $r_i(k+t) := r_i^*(1+t), t = 0, \dots, T-1$ 
      end
(5)   set the pilot signal of EV  $i$  to  $r_i(k), \forall i \in \mathcal{V}_k$ 
(6)    $e_i(k+1) := e_i(k) - \hat{e}_i(k), \forall i \in \mathcal{V}_k$ 
(7)    $d_i(k+1) := d_i(k) - 1, \forall i \in \mathcal{V}_k$ 
end

```

in which have non-zero remaining energy demand and are not already scheduled to depart. We then check, in line 2, if we should compute a new optimal schedule. This is done whenever an event-fired flag is True or when the time since the last computed schedule exceeds P periods.

If a new schedule is required, we call the optimal scheduling algorithm $\mathbf{OPT}(\mathcal{V}_k, U_k, \mathcal{R}_k)$ in line 3 that takes the form:

$$\max_r U_k(r) \tag{6.1a}$$

$$\text{s.t. } r \in \mathcal{R}_k \tag{6.1b}$$

The set \mathcal{V}_k of active EVs defines the optimization variable $r := (r_i(1), \dots, r_i(T), i \in \mathcal{V}_k)$ for every active EV i over the optimization horizon $\mathcal{T} := \{1, \dots, T\}$. The utility function U_k encodes the problem's objective, while the feasible set \mathcal{R}_k encodes various constraints. We will discuss them in detail in the next two subsections. Note that \mathbf{OPT} does not have a notion of the current time k and returns an optimal solution $r_i^* := (r_i^*(1), \dots, r_i^*(T))$ of (6.1) as a T -dimensional vector for each active EV i . The algorithm then adjusts the indexing and sets the scheduled charging rates of EVs i at time k as $r_i(k+t) := r_i^*(1+t), t = 0, \dots, T-1$ in line 4. At every time k , regardless of if our system produced a new schedule, we set the pilot signal of each EV i to $r_i(k)$ (line 5) and update the system state (lines 6, 7) for the next time period.

We now describe how to design the utility function U_k to achieve desirable features and model various constraints that define the feasible set \mathcal{R}_k for practical systems.

Utility Functions U_k

In general, charging system operators may have many objectives they wish to achieve via smart charging, including charging vehicles as quickly as possible, maximizing their operating profit, utilizing renewable energy sources, or smoothing their total load profile. Operators also have secondary objectives such as fairly distributing available capacity.

To allow operators to specify multiple objectives, our utility function $U_k(r)$ is a weighted sum of utility functions $u_k^v(r)$:

$$U_k(r) := \sum_{v=1}^V \alpha_k^v u_k^v(r)$$

We allow the utility function to change for each computation. Here $u_k^v(r)$, $v = 1, \dots, V$ are a set of utility functions that capture the system operator's objectives and promote desirable properties in the final schedule. Meanwhile, $\alpha_k^v > 0$, $v = 1, \dots, V$ are time-dependent weights used to determine the relative priority of the various components. To simplify notations, we will henceforth drop the subscript k when we discuss the computation at time k .

Charging quickly: One common operator objective is to charge all vehicles as quickly as possible. We can do this by specifying an objective such as

$$u^{QC}(r) := \sum_{t \in \mathcal{T}} \frac{T-t+1}{T} \sum_{i \in \mathcal{V}} r_i(t)$$

where the reward for delivering energy is strictly decreasing in time.

Minimizing cost / maximizing profit: Another common objective for system operators is to maximize their operating profit. Let π be the per-unit revenue from charging, and $\kappa(t)$ be the time-varying cost of one unit of energy. To account for other loads and generation which share a meter with the ACN, we define the net load

$$N(t) := \sum_{i \in \mathcal{V}} r_i(t) + L(t) - G(t)$$

where $L(t)$ denotes the net draw of the other loads while $G(t)$ denotes on-site generation such as PV. Since $L(t)$ and $G(t)$ are unknown for $t > 0$ this formulation relies on a prediction of these functions into the future. There are several methods for load/generation prediction proposed in the literature, but these are outside the scope of this paper. We can express the objective of maximizing profit as

$$u^{EC}(r) := \pi \sum_{\substack{t \in \mathcal{T} \\ i \in \mathcal{V}}} r_i(t) - \sum_{t \in \mathcal{T}} \kappa(t) N(t)$$

This is equivalent to cost minimization when $\pi = 0$.

Minimizing demand charge: In addition to energy costs, utilities often impose a price on the maximum power draw in a billing period called demand charge. Since demand charge is assessed over an entire month, while the optimization horizon is typically < 12 hours, we replace the full demand charge P with a proxy $\hat{P} \leq P$. We also introduce q_0 to be the highest peak so far in the billing period and q' as a prediction of the optimal peak. We can then express the demand charge as

$$u^{DC}(r) := -\hat{P} \cdot \max \left(\max_{t \in \mathcal{T}} N(t), q_0, q' \right)$$

Note that \hat{P} and q' are tunable parameters. We describe the selection of these in Section 6.2.

Minimizing total load variations (load flattening): Another common objective for EV charging operators is to minimize load variations. We can express this objective as

$$u^{LV}(r) := - \sum_{t \in \mathcal{T}} N(t)^2$$

Fairly distributing capacity: The utility functions described so far are not strictly concave in r and hence the optimal solution, r^* , is generally non-unique. We can force a unique optimal solution by including the regularizer:

$$u^{ES}(r) := - \sum_{\substack{t \in \mathcal{T} \\ i \in \mathcal{V}}} r_i(t)^2$$

This regularizer also promotes equal sharing among the EVs, which is desirable for the operator and drivers, and minimizes line losses along the lines which feed each EVSE. This property comes from the fact that all things being equal, this component is maximized when all charging rates are as low as possible. Thus, it is sub-optimal to have one EV charging faster than another if both charging at an equal rate would result in the same optimal value for all other objective components.

Non-completion penalty: A general goal of EV charging systems is to meet users' energy needs by their deadlines. While this can be accomplished by an equality

constraint in \mathcal{R}_k , doing so can lead to infeasibility. Instead, we can use the inequality constraint (6.2c), and add a non-completion penalty of the form:

$$u^{NC_p}(r) := -\sqrt[p]{\sum_{i \in \mathcal{V}} \left| \sum_{t \in \mathcal{T}} r_i(t) - e_i \right|^p}$$

where $p \geq 1$. This is the p-norm of the difference between the energy delivered to each EV and its requested energy. When $p = 1$, this regularizer shows no preference between EVs. For $p > 1$, EVs with higher e_i will be prioritized (given more energy) over those with lower e_i when it is infeasible to meet all energy demands. Note that this regularizer is 0 whenever the energy demands of all EVs are fully met, e.g. $\sum_{t \in \mathcal{T}} r_i(t) = e_i$. Thus, with sufficient weight on this component, (6.2c) will be tight whenever feasible. Likewise, if (6.2c) would have been tight without (7), this regularizer has no effect.

Feasible set \mathcal{R}_k

The feasible set \mathcal{R}_k is defined by a set of equality and inequality constraints that can depend on k , but for notational simplicity, we drop the subscript k . These constraints then take the form:

$$0 \leq r_i(t) \leq \bar{r}_i(t) \quad t \leq d_i, i \in \mathcal{V} \quad (6.2a)$$

$$r_i(t) = 0 \quad t > d_i, i \in \mathcal{V} \quad (6.2b)$$

$$\sum_{t \in \mathcal{T}} r_i(t) \leq e_i \quad i \in \mathcal{V} \quad (6.2c)$$

$$\left| \sum_{i \in \mathcal{V}} A_{li} r_i(t) e^{j\phi_i} \right| \leq c_l(t) \quad t \in \mathcal{T}, l \in \mathcal{L} \quad (6.2d)$$

Constraints (6.2a) ensure that the charging rate in each period is non-negative (we do not consider V2G) and less than its upper bound defined by the EV's BMS and the maximum pilot supported by the EVSE. This is a relaxation of the set of discrete rates allowed by the EVSE and is necessary to keep the scheduling problem convex. We discuss how to recover a feasible discrete solution below. Constraints (6.2b) ensure that an EV does not charge after its departure time. We use constraints (6.2c) to limit the total energy delivered to EV i to at most e_i . To ensure feasibility, we do not require equality (the zero vector is always a feasible solution). This ensures

that **OPT** always returns a feasible schedule, which is important in practice. We can then craft the objective function to ensure this constraint is tight whenever possible.

Quantization of pilot signal

The pilot signal constraints imposed by EVSEs described in Section 2.4 are discrete and intractable in general for large problems. Because of this, we do not include (2.9) in the definition of \mathcal{R}_k , instead relaxing it to (6.2a). However, to account for our non-zero rate constraint, we add the constraint

$$r_i(0) \geq \min(\rho_i(0) \setminus \{0\})$$

to (6.2).¹ We denote the output of this optimization $r^* := (r_i^*(t), \forall i \in \mathcal{V} \forall t \in \mathcal{T})$. For simplicity, we assume that the maximum P between scheduler calls (see Algorithm 1) is set to the length of one period, so only the first charging rate in r^* will be applied.

We then round $r_i^*(0)$ down to the nearest value in ρ_i :

$$\tilde{r}_i(0) \leftarrow \lfloor r_i^*(0) \rfloor_{\rho_i}$$

This rounding may leave unused capacity that our algorithms can reclaim. To reclaim this capacity, we first sort EVs in descending order by the difference between their originally allocated charging rate, $r_i^*(0)$, and the rate after rounding, $\tilde{r}_i(0)$. We then iterate over this queue and increment each EV's charging rate to the next highest value in $\rho_i(t)$, if the resulting current vector $\tilde{r}(0) \in \mathcal{R}_k$ and $\sum_i \tilde{r}_i(0) \leq \sum_i r_i^*(0)$. We continue to loop over this queue until we cannot increment any EV's allocated rate.

Battery tail capacity reclamation

As discussed in Section 2.4, an EV's battery management system will sometimes limit the power draw of the battery as it approaches 100% state-of-charge. When this happens, the difference between the pilot signal and the vehicle's actual charging rate is wasted capacity. To reclaim this capacity, we use a simple algorithm which we call *ramp-down*. Let $r_i^k(0)$ be the pilot signals sent to EV i at time k , $m_i(k)$ be

¹This constraint implicitly assumes that it is feasible to deliver a minimum charging rate to each EV, thus charging infrastructure should be designed with this constraint in mind if the operators want to ensure a minimum charging rate to each EV.

its measured charging current, and $\bar{r}_i^k(0)$ be the upper bound on its charging rate. We define two thresholds, θ_d and θ_u . If $r_i^k(0) - m_i(k) > \theta_d$, we can reclaim some capacity by setting the upper limit on pilot signal of EV i for the next period to be $m_i(k) + \sigma$, where σ is typically around 1 A. To account for the possibility of the EV's BMS only limiting current temporarily, if $\bar{r}_i^k(0) - m_i(k) < \theta_u$, we increment the pilot signal upper bound by σ (clipping at the EV's BMS limit or the EVSE's pilot limit). With this scheme, we can quickly reclaim capacity during the tail region while still allowing EVs to throttle back up if this reclamation was premature. Note that in our current implementation, the upper bound on the pilot signal, $\bar{r}_i^k(t)$ is the same for all t within the same sub-problem k . In more advanced ramp-down schemes, this bound could depend on t or the decision variables $r(t)$.

6.2 Evaluating ASA Against Baseline Algorithms

We can now use ACN-Data and ACN-Sim to evaluate the proposed Adaptive Scheduling Algorithm compared to traditional baseline scheduling algorithms. We do not consider other EV charging approaches proposed in the literature as baselines since they are not applicable under our more realistic setting, for example, unbalanced three-phase infrastructure and discrete pilot signals. The code for these experiments is available on Github [95].

Charging Workloads

For these case studies, we will use data collected from the California Garage ACN between May 1, 2018 - October 1, 2018. During this time, only the subset of 54 EVSEs under transformer t_1 were active. This corresponds to cluster *California_Garage_01* in ACN-Data. Additionally, charging was free during this period, which resulted in high utilization. Statistics on the charging workload from this period are shown in Fig. 6.1 and Table 6.1. During this period, the system served 10,415 charging sessions and delivered 92.78 MWh of energy.

From Table 6.1, we observe a significant difference in system usage between weekdays and weekends. The total energy delivered is much higher on weekdays, but this energy is divided over far more charging sessions leading to lower per-session energy delivery. Also, charging sessions on weekends tend to be shorter than those during the week. This means that our system must handle large numbers of flexible sessions on weekdays and smaller numbers of relatively inflexible sessions on weekends. This behavior precludes simple solutions such as installing large numbers of

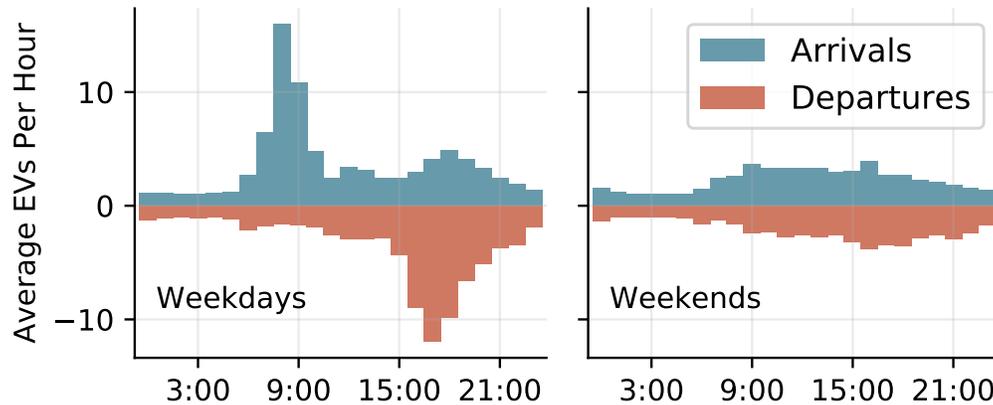


Figure 6.1: Average arrivals and departures per hour for the period May 1, 2018 - October 1, 2018 (54 EVSEs). On weekdays we see a peak in arrivals between 7:00 - 10:00 followed by a peak in departures between 16:00 - 19:00. The Caltech ACN also has a much smaller peak in arrivals beginning around 18:00, which is made up of community members who use the site in the evening, including some patrons of the nearby campus gym. Weekends, however, have a more uniform distribution of arrivals and departures.

Table 6.1: Average Statistics for EV Charging Test Cases Per Day
May 1, 2018 - Oct. 1, 2018

	Mean Daily Sessions	Mean Session Duration (hours)	Mean Session Energy (kWh)	Mean Daily Energy (kWh)	Max Concurrent Sessions
Sun	41.32	3.94	10.05	415.06	18
Mon	71.00	6.14	9.54	677.13	42
Tues	76.73	6.24	8.94	685.79	47
Wed	75.45	6.22	8.75	660.16	44
Thurs	78.50	5.96	8.47	665.21	42
Fri	77.18	6.71	9.04	697.41	43
Sat	43.32	5.01	10.15	439.59	18

level-1 chargers, which would be too slow on weekends and for low laxity weekday sessions, or small numbers of level-2 chargers, which would be insufficient for the number of concurrent sessions on weekdays.

In the physical system, we rely on users to estimate their departure time and energy request. In this section, we will assume that drivers are accurate in their predictions. This allows us to isolate the performance of the algorithm from the accuracy of the user inputs. In Chapter 8, we will discuss how we can use machine learning to improve predictions based on historical data.

Objective Functions

We will consider two common operator objectives within these case studies, charging users as quickly as possible and minimizing costs subject to time-of-use tariffs and demand charge. In 6.3 we will also consider a load flattening objective.

Quick Charge.

We first consider the objective of maximizing total energy delivered when infrastructure is oversubscribed. This is a common use case when electricity prices are static or when user satisfaction is the primary concern. To optimize for this operator objective, we use the Adaptive Scheduling Algorithm (ASA) (Alg. 1) with the utility function

$$U^{QC}(r) := u^{QC}(r) + 10^{-12}u^{ES}(r)$$

Here U^{QC} encourages the system to deliver energy as quickly as possible, which helps free capacity for future arrivals. We include the regularizer $u^{ES}(r)$ to promote equal sharing between similar EVs and force a unique solution. We refer to this algorithm as ASA-QC. We set the weight of the $u^{ES}(r)$ term to be small enough to ensure a strict hierarchy of terms in the objective.

Cost Minimization.

Next, we consider the case where a site host would like to minimize their operating costs. This case study will consider the Southern California Edison TOU EV-4 tariff schedule for separately metered EV charging systems between 20-500 kW, shown in Table 6.2 [96]. In each case, we assume that the charging system operator has a fixed revenue of \$0.30/kWh and only delivers energy when their marginal cost is less than this revenue.

In order to maximize profit, we use the objective

$$U^{PM} := u^{EC} + u^{DC} + 10^{-6}u^{QC} + 10^{-12}u^{ES} \quad (6.3)$$

We denote the ASA algorithm with this objective ASA-PM.

The revenue term π in u^{EC} can have several interpretations. In the most straightforward case, π is simply the price paid by users. However, π can also include

Table 6.2: SCE EV TOU-4 Rate Schedule for EV Charging

Period	Time Range	Summer		Winter	
		Weekday	Weekend	Weekday	Weekend
Off-Peak	23:00 - 8:00	\$0.056 / kWh	\$0.056 / kWh	\$0.061 / kWh	\$0.061 / kWh
Mid-Peak	8:00 - 12:00	\$0.092 / kWh	\$0.056 / kWh	\$0.075 / kWh	\$0.061 / kWh
	18:00 - 23:00				
Peak	12:00 - 18:00	\$0.267 / kWh	\$0.056 / kWh	\$0.087 / kWh	\$0.061 / kWh
Demand Charge	Monthly	\$15.51 / kW			

subsidies by employers, governments, automakers, or carbon credits through programs like the California Low-Carbon Fuel Standard (LCFS). For example, LCFS credits for EV charging have averaged between \$0.13 - \$0.16 / kWh in 2018-2019. In these cases, the algorithm might not meet some energy demands if the marginal price of that energy exceeds π . This is especially important when demand charge is considered since the marginal cost can be extremely high if it causes a spike above the previous monthly peak. Alternatively, π can be set to a very high value (greater than the maximum marginal cost of energy) and act as a non-completion penalty. When this is the case, the algorithm will attempt to minimize costs while meeting all energy demands (when it is feasible to do so).

In u^{DC} , \hat{P} , and q' are tunable parameters. The demand charge proxy \hat{P} controls the trade-off between energy costs and demand charges in the online problem. If \hat{P} is high, i.e., $\hat{P} = P$, the algorithm will only increase its peak when it absolutely must. However, if \hat{P} is too low, e.g., $\hat{P} = 0$, the algorithm will increase its peak significantly even if doing so will only lead to a small decrease in energy costs. We propose the following heuristic, $\hat{P} = P / (D_p - d)$, where D_p be the number of days in the billing period, and d be the index of the current day. This heuristic is based on a simple amortization. At the beginning of the billing period, any increase in the demand charge can be spread over D_p days. The next day it can only be spread over $D_p - 1$ days, and so on. Thus, this heuristic encourages any increases in demand charge to occur early in the billing period, which allows the algorithm to decrease energy costs in the remainder of the billing period by concentrating charging during low-cost times.

For the peak hint, q' , We will consider one version of the algorithm without a peak hint, e.g., $q' = 0$, and one where the peak hint is 75% of the optimal peak calculated using data from the previous month. This percentage is chosen based on maximum historic month-to-month variability in the optimal peak (+11%/-16%).

We also include the quick charge objective as a regularizer, which encourages the scheduling algorithm to front-load charging within a TOU period. To ensure that this regularizer does not lead to a large increase in cost, we use a coefficient of 10^{-6} . This results in a maximum increase in value of $\$0.000050 / \text{kWh}$, three orders of magnitude lower than the minimum cost of energy in Table 6.2.

Impact of Three-Phase Models

As we saw in Section 2.4, unbalance can be a major concern in large-scale charging systems. However, to date, most algorithms proposed in the literature implicitly assume single-phase or balanced three-phase operation.

To see why these assumptions are insufficient for practical systems, we consider two versions of ASA-QC. In the first, ASA-QC only ensures that the total power draw is less than the transformer’s capacity (70 kW), which is sufficient for a single-phase or balanced system. In the second, ASA-QC uses the full three-phase system model that includes individual line constraints. This experiment’s results are shown in Fig. 6.2, where we can see that only considering maximum power draw leads to significant constraint violations in line currents. However, by using an algorithm that considers the full three-phase model, we ensure these line constraints are not violated at the cost of not fully utilizing the 70 kW transformer’s capacity due to unbalance.

This motivates us to consider algorithms that incorporate unbalanced three-phase constraints. These constraints are necessary to ensure safety and can significantly impact the performance of an algorithm. To see this, we will consider the percentage of user energy demands met when infrastructure constraints are binding. We use this metric to evaluate six algorithms over a range of possible transformer capacities based on the real charging workload of the Caltech ACN from September 2018. To demonstrate the effect of infrastructure models, we conduct this experiment with single-phase and three-phase models, as shown in Fig. 6.3. Here we can see that in the single-phase case, EDF, LLF, and ASA-QC all perform near optimally², exceeding the performance of Round Robin and FCFS by up to 8.6%. However, the subplot on the right tells a different story. Here we see that the ASA-QC can match the offline optimal performance as before, while EDF and LLF both underperform.

²Here optimally is defined as the maximum amount of energy that an algorithm with perfect foresight could deliver subject to constraints. It is found by solving (6.1) with perfect knowledge for all EVs in the simulation. We use $U(r) = \sum_{i \in \mathcal{V}_{all}, t \in \mathcal{T}} r_i(t)$.

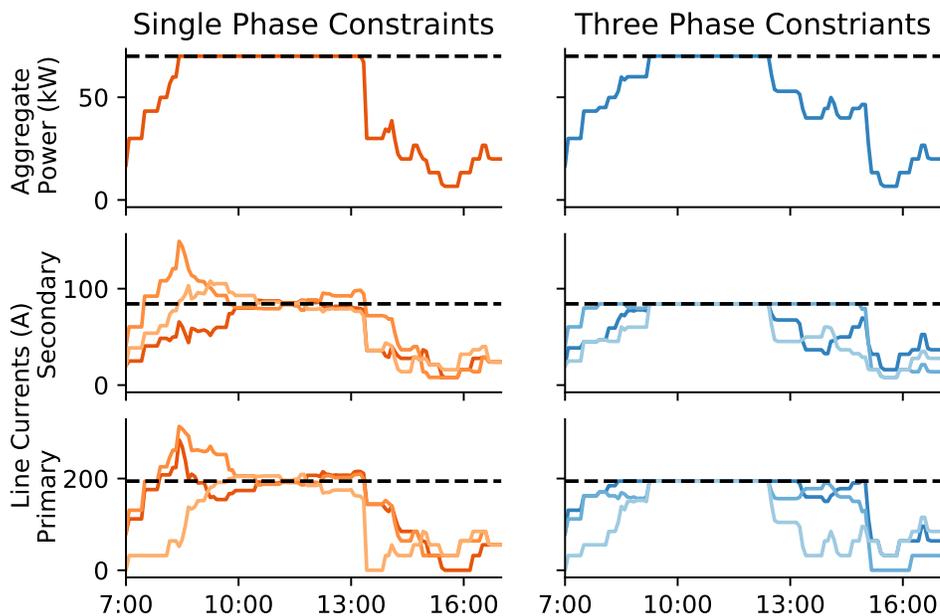


Figure 6.2: Aggregate power draw and line-currents at the primary and secondary side of the transformer when running single-phase and three-phase ASA-QC algorithms on the Caltech ACN with a 70 kW transformer capacity. Shading in the lower plots denotes each phase, while the black dotted line denotes the power/current limit. The experiment is based on data from the Caltech ACN on September 5, 2018, and uses a 5-minute timestep.

In fact, in the highly constrained regime, Round Robin outperforms EDF and LLF despite having less information about the workload. We attribute these results to the importance of phase-balancing in three-phase systems, which has been historically under-appreciated in the managed charging literature.

To understand why ASA-QC performs so much better than the baselines, we must consider what information each algorithm uses. RR uses no information aside from which EVs are currently present and performs the worst. Likewise, EDF uses only information about departure time, while LLF also uses the EV’s energy demand. Only ASA-QC actively optimizes over infrastructure constraints, allowing it to better balance phases (increasing throughput) and prioritize EVs, including current and anticipated congestion. A key feature of the ASA framework is its ability to account for all available information cleanly.³

In addition to comparing algorithms, the curves in Fig. 6.3 can also inform charging

³When even more information is available, i.e., a model of the vehicle’s battery or predictions of future EV arrivals, this information can also be accounted for in the constraint set \mathcal{R} and objective $U(r)$. However, these formulations are outside the scope of this paper.

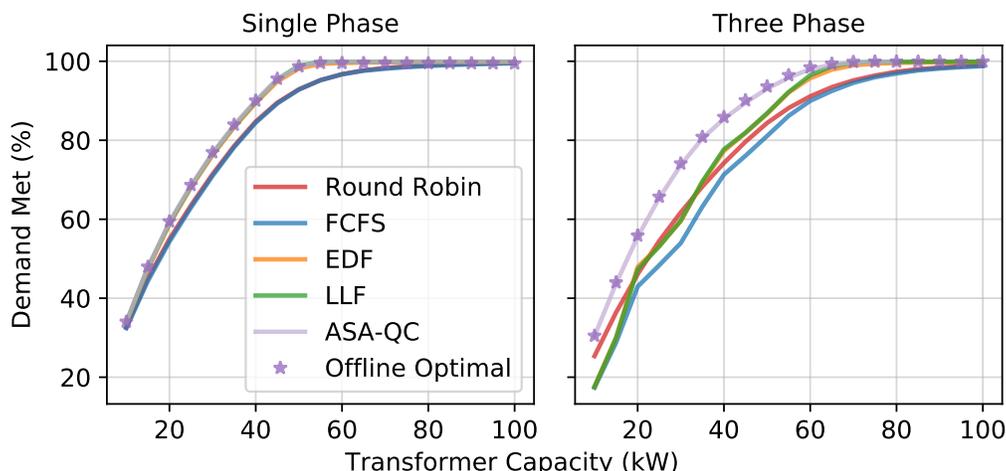


Figure 6.3: Comparison of percentage of energy delivered as a function of transformer capacity for single-phase (left) and three-phase (right) systems. Stars represent the offline optimal, which is an upper bound based on perfect future information. The simulation runs from Sept. 1 through Oct. 1, 2018, with a timestep of 5 minutes. To generate events, we use ACN-Sim’s integration with ACN-Data to get real charging sessions from the Caltech ACN, assuming the ideal battery model. We also use the included Caltech ACN charging network model with ideal EVSEs and use its optional `transformer_cap` argument to limit the infrastructure capacity. In the left plot, ASA-QC, EDF, and LLF are nearly coincident, as are Round Robin and FCFS. Similarly, in the right plot, EDF and LLF overlap in most cases.

systems’ design when accounting for the online algorithm used. For example, we can see that if a host wants to deliver $>99\%$ of charging demand using ASA-QC, a 70 kW transformer would be sufficient, assuming an unbalanced three-phase system. Alternatively, if an existing transformer can only support 40 kW of additional demand, a host could expect to meet approximately 85% of demands without an upgrade.

Impact of Pilot Quantization and Non-Ideal Batteries

In addition to unbalanced three-phase infrastructure, non-ideal batteries and pilot signal quantization can significantly affect the performance of algorithms. To better understanding these effects, we consider each operator objective in the context of the five scenarios in Table 6.3. Here perfect information refers to having access to all EVs’ arrival time, duration, and energy demand in advance, allowing for offline optimization. Continuous EVSEs allow for continuous pilot control between 0 and the EVSE’s upper bound. In contrast, quantized EVSEs only allow a discrete set of values and must keep the charging rate at or above 6 A until the EV is finished charging. Finally, ideal batteries are assumed to follow the pilot signal exactly. In contrast, non-ideal batteries follow the constant current, constant voltage model

Table 6.3: Modeling Assumptions by Scenario

	I	II	III	IV	V
Perfect Information?	✓	✗	✗	✗	✗
Continuous EVSE?	✓	✓	✗	✓	✗
Ideal Battery?	✓	✓	✓	✗	✗

described in [39], where the initial state of charge and battery capacity are fit to maximize tail behavior, and the tail begins at 80% state-of-charge.

For our simulations, we use ACN-Sim, which includes realistic models for each of the scenarios above. In each case, we consider the three-phase infrastructure of the Caltech ACN. We set the length of each time slot to 5 minutes, the maximum time between scheduler calls to 5 minutes, and consider a maximum optimization horizon of 12 hours.

Energy delivery with constrained infrastructure.

We first consider the objective of delivering as much energy as possible within the bounds of our infrastructure. To control congestion in the system, we vary the capacity of transformer t_1 between 20 and 150 kW. For reference, the actual transformer in our system is 150 kW, and a conventional system of this size would require 362 kW of capacity. We then measure the percent of the total energy demand met using ASA-QC as well as three baseline scheduling algorithms; least laxity first (LLF), earliest deadline first (EDF), and round-robin (RR), as implemented in ACN-Sim and are described in [39]. These baseline algorithms are very common in the deadline scheduling literature and have been applied previously to the EV charging domain[26], [29]. In addition, the round robin algorithm is a generalization of the equal sharing algorithm used by many charging providers today. We also consider the maximum energy that could be delivered by solving (6.1) with objective

$$U^{\text{EM_OFF}}(r) := \sum_{\substack{t \in \mathcal{T} \\ i \in \mathcal{V}}} r_i(t)$$

and perfect foreknowledge of future arrivals, i.e., \mathcal{V} includes all EVs, not just those present at time k . We also modify (6.2c) so that EVs cannot charge before their arrival time. We refer to this as the *Optimal* solution.

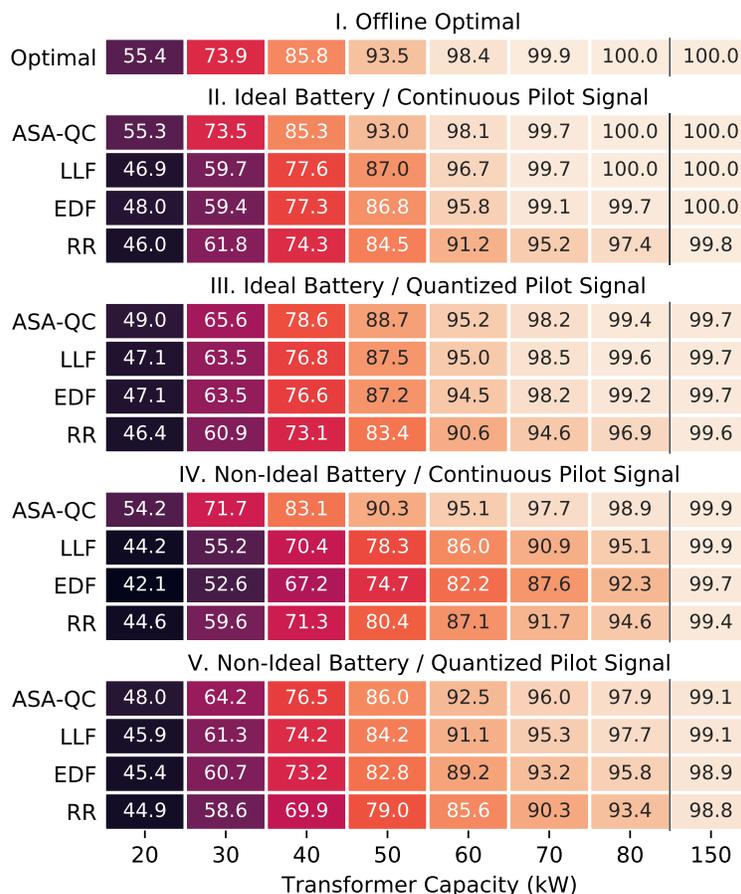


Figure 6.4: Percentage of driver's energy demands that can be met at varying capacities for transformer t_1 for Sept. 2018. Here demand met is defined as the ratio of total energy delivered to total energy requested.

Results from this experiment are shown in Fig. 6.4, from which we observe the following trends.

1. In scenario II, ASA-QC performs near optimally (within 0.4%) and significantly outperforms the baselines (by as much as 14.1% compared to EDF with 30 kW capacity).
2. In almost all cases, ASA-QC performs better than baselines, especially so in highly congested settings. ⁴

⁴For scenarios III and V and transformer capacities less than 68 kW, it may sometimes be infeasible to allocate a minimum of 6 A to each active EV. When this is the case, we allocate 6 A to as many EVs as possible then allocate 0 A to the rest. This allocation is done by first sorting EVs (by laxity for LLF and ASA-QC, deadline for EDF, and arrival time for RR) then allocating 6 A to each EV until the infrastructure constraints are binding.

3. Non-ideal EVSEs (scenarios III and V) have a large negative effect on ASA-QC, which we attribute to rounding the optimal pilots and restriction of the feasible set.
4. Surprisingly, non-ideal EVSEs increase the performance of LLF and EDF for transformer capacities <60 kW. This may be because the minimum current constraint leads to better phase balancing.
5. Non-ideal batteries (scenarios IV and V) have a relatively small effect on the performance of ASA-QC compared to baselines, indicating the robustness of the algorithm.

Profit maximization with TOU tariffs and demand charge.

We can then turn our attention to the problem of cost minimization. For this case, we fix the transformer capacity to 150 kW and consider the previous baselines along with uncontrolled charging, which is the most common type of charging system today. We also consider the optimal profit possible by solving (6.1) with perfect foreknowledge of arrivals and objective:

$$U^{\text{PM_OFF}} := u^{\text{EC}} + u^{\text{DC}}$$

with $\hat{P} = P$, $q' = 0$.

Results of the experiment are shown in Fig. 6.5, from which we observe:

1. Profits from both ASA-PM and ASA-PM w/ Hint are within 3.6% and 1.9% of the optimal, respectively, and far exceed the profits of all baseline algorithms.
2. Uncontrolled, LLF and RR result in *lower* energy costs, but incur *very high* demand charges. These algorithms are not price aware. Instead, low energy costs are a result of drivers arriving during off-peak and mid-peak times. In particular, uncontrolled charging, which does not consider an infrastructure limit, leads to *extremely high* demand charges. On the other hand, both ASA-PM algorithms (and the offline optimal) trade-off higher energy costs for much lower peaks resulting in lower overall costs.
3. Providing a peak hint to ASA-PM increases revenue by allowing more energy demands to be met. In this case, 97.8% vs. 95.6% without peak hints. Accurate hints allow the algorithm to utilize higher capacity earlier in the

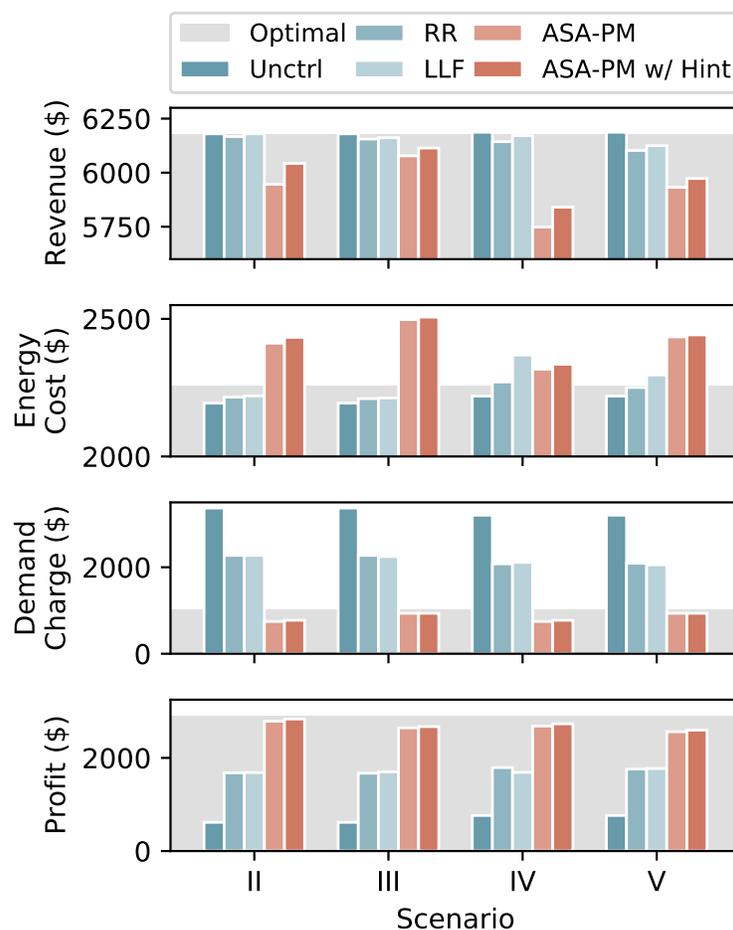


Figure 6.5: Operator profit, costs, and revenue for various scheduling approaches when using SCE's EV TOU-4 tariff, $\pi = \$0.30$, 150 kW transformer capacity, and data from Sept. 2018. In each case, the offline optimal in the ideal setting is shown as a grey background.

billing period, increasing throughput without increasing cost. Even with the peak hint, ASA-PM does not meet 100% of demands even though the offline optimal does. Since ASA-PM does not have knowledge of future arrivals, it must act conservatively in increasing the peak over time. It is, however, important that hints not be too large, as the algorithm can increase the peak as needed, but once a high peak is set, we cannot lower the demand charge.

4. While EVSE quantization and non-ideal batteries each reduce the operator's profit, even in Scenario V, ASA-PM w/ Hint still produces 90% of the optimal profit.
5. Interestingly, revenue increases in scenarios with quantization (III and V). It can be hard to reason exactly why this occurs, though it appears that the

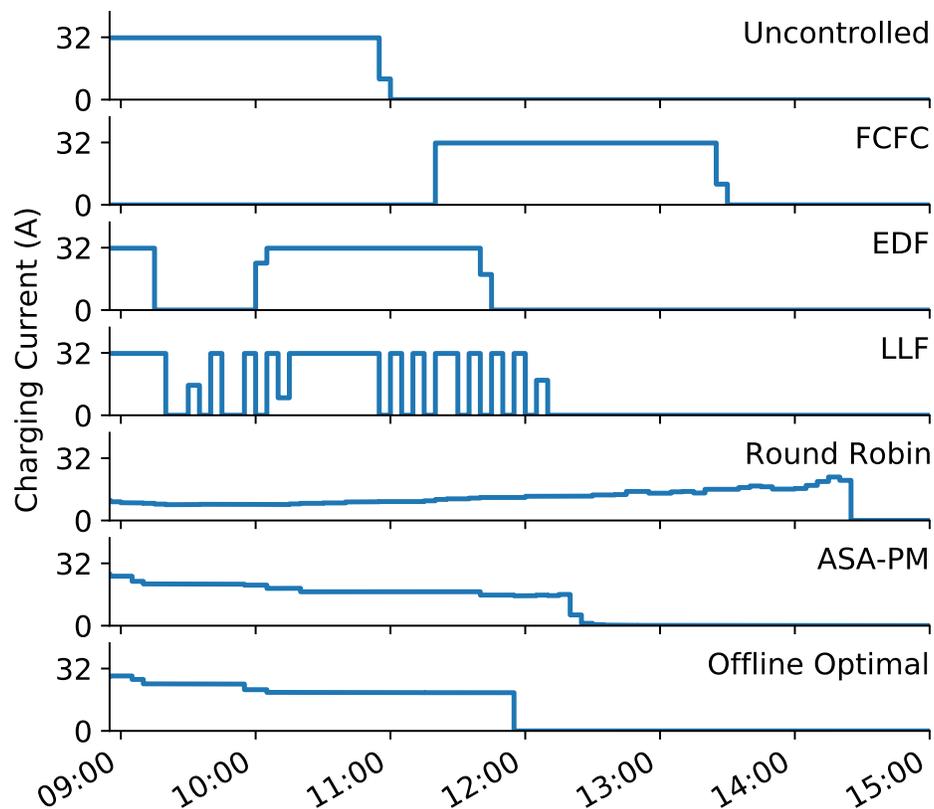


Figure 6.6: Comparison of charging profiles for one EV on September 13, 2018 with a 70 kW transformer capacity.

post-processing step leads to initial conditions for the next solve of **OPT** to produce a higher revenue, higher cost solution.

6. Because we use real tariffs structures, real workloads, and realistic assumptions (Scenario V), we can conclude with reasonable certainty that a charging system operator could expect to net approximately \$2,600 / month using an ACN like system, compared to just \$763 / month in a conventional, uncontrolled system.

Charging Profiles

In addition to aggregate metrics like percentage of energy delivered and total cost, ACN-Sim allows us to examine the charging profiles of individual sessions. Fig. 6.6 shows the charging profiles each algorithm produces for a particular charging session assuming three-phase infrastructure, ideal battery, and continuous pilot signals. Here we can see a qualitative difference between the algorithms. For example, FCFS

behaves very similarly to Uncontrolled charging but is delayed as the EV must wait its turn in the queue. For EDF and LLF, charging can be interrupted when EVs with earlier deadlines arrive or as an EV's laxity evolves over time. Oscillations in the LLF plot result from an increase in laxity as the EV charges, which can decrease its standing in the queue, causing it to stop charging temporarily. These oscillations are generally bad for user experience, preventing LLF from being used widely for smart charging. The smoothed LLF algorithm proposed in [28] adapts the LFF algorithm to prevent these oscillations. Round Robin, ASA-PM, and the offline optimal are quite different. Each EV charges steadily but at a rate below its maximum as congestion in the system necessitates sharing of charging capacity. Here both ASA-PM and the offline optimal use objective (6.3). With this tariff schedule, on-peak rates run from 12 - 6 pm — Offline Optimal finishes charging this user before this, while ASA-PM goes slightly into the on-peak period.

6.3 Load Flattening to Reduce Grid Impacts

Charging systems do not operate in a vacuum. In almost all cases, they draw energy from the power grid. Because of their enormous power and energy requirements, large-scale EV charging systems can have major impacts on the power grid.

In a 2019 study, researchers from Lawrence Berkeley National Laboratory considered 39 distribution feeders (including 19 residential feeders)[17]. They determined that at an adoption rate of one EV per household and with uncontrolled charging, 58% of the residential feeders analyzed exceeded their peak power capacity, 47% had some line overloading, and 16% experienced voltage issues (<0.95 p.u.). Extrapolating from an average peak demand increase of 64% in these feeders, the researchers estimated that 60% of residential feeders in the San Francisco Bay Area would require upgrades to support uncontrolled EV charging loads. This study highlights the need for smart charging approaches. The authors estimate that a relatively simple smart charging approach of shifting demand to between the hours of 6 pm and 6 am while minimizing peak demand could remove the need for reinforcement on all feeders considered. This study demonstrates both the challenges of integrating EV charging into the distribution grid and the opportunities that smart charging approaches like the ACN provides.

The ACN Research Portal allows us to expand studies like this to consider how more advanced smart charging approaches can help alleviate strain on the distribution system, especially for large charging systems like workplaces. To enable studies like

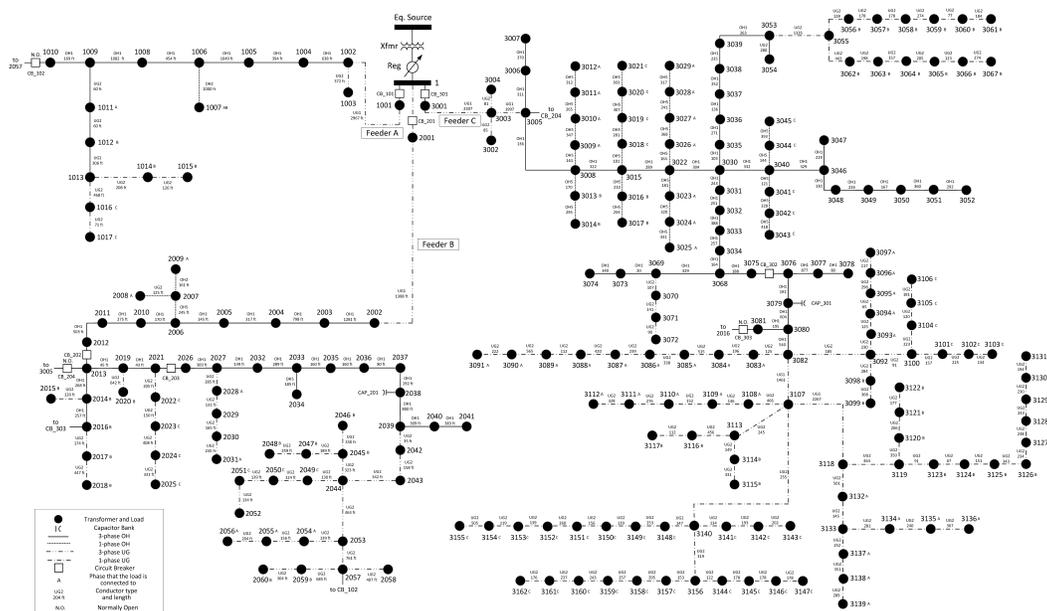


Figure 6.7: 240-node distribution feeder from the Midwest United States [97]. We consider an ACN located at node 2053.

this with the ACN-Sim, we have integrated it with several grid simulation packages, including MATPOWER, PandaPower, and OpenDSS. In each case, we can use ACN-Sim and ACN-Data to provide a realistic load profile, which can then feed into the grid simulation package that evaluates power flows and alerts us to any voltage or overloading issues. The code for these experiments is available on Github [95].

Test Feeder

For this case study, we will consider a 240-node distribution system located in the Midwest United States shown in Fig 6.7 [97]. This distribution system consists of three feeders which share a substation and voltage regulator. This substation serves a total of 1,120 customers. Load data is available with 1-hour resolution at the level of the distribution transformer. We use OpenDSS to model this circuit.

To evaluate if this circuit could support a large-scale EV charging system, let's consider adding an EV charging facility to bus 2053, which has a transformer capacity of 225 kVA. To model this new facility, we will assume it is a replica of the JPL Arroyo Garage ACN included in ACN-Sim, and consider the charging workload from the JPL ACN on September 5, 2019. As we have seen previously, unbalance can play an important role in large-scale charging systems, so we model the ACN as an unbalanced three-phase load in OpenDSS. For load on the feeder, we take smart

meter data from September 5, 2017 (both days were weekdays).

Load Flattening

To minimize the effect of EV charging on the grid, we will seek to flatten the load at the EV site. This prevents large spikes in demand which could cause voltage issues on the distribution feeder. While we only consider local load flattening in this example, ACN-Sim could also be used to investigate global load flattening based on the aggregate load on the feeder.

Load flattening, also known as valley-filling, is a common approach in distributed energy resource literature. In [30] the authors propose a decentralized approach to load flattening for EV charging. Their proposed algorithm is iterative. At each iteration, the utility calculates a time-varying price based only on the aggregate charging rates of all EVs over time and the background load that is assumed to be known in advance. Each EV can then solve an optimization to minimize its costs according to these time-varying prices along with a regularization term which helps ensure convergence. Like our ASA algorithm, model predictive control is used to adapt this algorithm to the online case. Likewise, in [31], the authors propose an online expected load flattening (ELF) algorithm based on model predictive control, which allows for scalable load flattening that is independent of the number of EVs and, in some cases, the length of the optimization horizon. While both of these works provide impressive results for load flattening, neither demonstrate what effect this load flattening will have on the grid.

In this case study, we use ACN-Sim and ACN-Data to consider how load flattening can help decrease the impact of a large charging facility on the distribution feeder. We use the ASA framework with a load flattening objective:

$$U^{\text{LoadFlat}} := u^{LV} + 100u^{NC_1} + 10^{-3}u^{ES} \quad (6.4a)$$

Within u^{LV} , $N(t)$ refers to the bus's net background load after subtracting out on-site generation. The u^{NC_1} term is a non-completion penalty for failing to deliver all the energy requested by EVs. u^{NC_1} has a coefficient of 100 to encourage delivering all the demanded energy before flattening the load. This coefficient is selected empirically to ensure that $> 99.5\%$ of energy demands are met. We add the equal sharing term to ensure a unique solution to the optimization.

Test cases

We consider four cases, a baseline with no EV charging, uncontrolled charging, ASA with a load flattening objective, and ASA with load flattening and on-site solar.

Because none of these algorithms use direct feedback from the grid simulator, we first run the ACN simulation for the full 24-hour horizon, then use this power draw as an input to OpenDSS. The results of these experiments are shown in Fig. 6.8. Uncontrolled charging results in an unacceptable minimum voltage of under 0.93 p.u. and overloads the transformer at bus 2053. This indicates that the grid as designed could not support uncontrolled charging at this scale at bus 2053.

To prevent voltage issues, we can schedule charging during periods of low background load by using MPC with objective (6.4a). We provide the actual building load as an input to the algorithm and ensure that the total load is constrained to be below the transformer’s capacity. From Fig. 6.8, we can see that this improves the minimum system voltage to 0.965 p.u., which matches the system-wide minimum from the baseline.

Since many EV charging systems are co-located with solar PV, we consider adding a 270 kW DC (225 kW AC) PV array at bus 2053. The solar data was generated from NREL’s SAM tool for Des Moines, Iowa, in a typical meteorological year (TMY) for Sept. 5. We use the same MPC algorithm but now set the background load to the net load after subtracting solar. We see in Fig. 6.8 this roughly recovers the same grid-wide minimal voltage as before we added an ACN, indicating that smart charging and solar PV could enable widespread adoption of EV charging without adverse grid impacts. These case studies assume perfect knowledge of background load and generation, as forecasting methods are beyond this study’s scope. However, no knowledge of future EV arrivals is used.

6.4 Concluding Remarks

Through building the ACN, we have identified practical challenges, including unbalanced three-phase infrastructure, quantization of pilot signals, and non-ideal battery behavior, which require us to rethink classical scheduling approaches. To meet these challenges, we propose ASA, a flexible model predictive control based algorithm along with pre- and post-processing heuristics, which can be easily configured to meet different operator objectives. We propose a collection of such objectives, including regularizers to promote desirable properties in the final schedule.

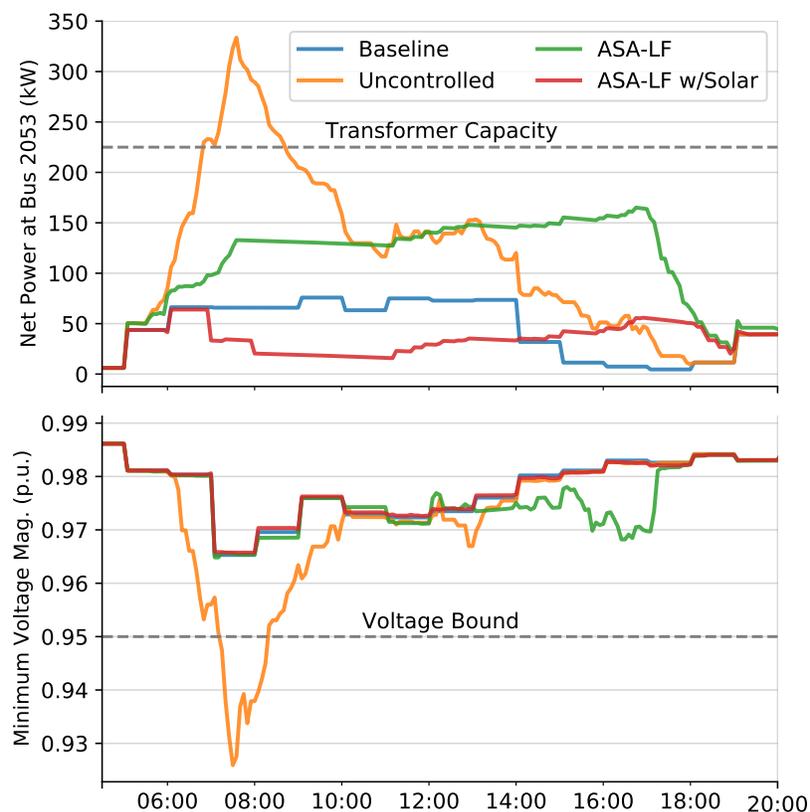


Figure 6.8: Comparing the effect of charging scenarios on net power draw at Bus 2053 (top) and minimum system voltage (bottom). EV data is taken from September 6, 2018, at the JPL ACN. Background load is taken from smart meter data from September 6, 2017. Solar PV production is estimated using NREL’s SAM tool for a 225 kW AC PV array in Des Moines, Iowa.

Using real workload data collected from the Caltech ACN, we find that unbalanced three-phase infrastructure significantly affects the absolute and relative performance of algorithms. We also find that ASA offers significant improvements in energy delivered with constrained infrastructure when continuous pilots are allowed and performs comparably to baselines when pilots are restricted to a discrete set of values. We also note that by changing the objective function, we can easily modify ASA to maximize operator profit. Using real data from Sept. 2018, we achieve \$2,835 in profit (98.1% of offline optimal) in an idealized setting and \$2,600 (90% of offline optimal) when considering non-ideal batteries and EVSEs. Compared to uncontrolled charging systems, our simulations show that with ASA, we can increase an EV charging system operator’s profit by 3.4 times (see Section 6.2).

Published Works

This chapter contains text and figures originally published in:

Z. J. Lee, G. Lee, T. Lee, C. Jin, R. Lee, Z. Low, D. Chang, C. Ortega, and S. H. Low, “Adaptive Charging Networks: A Framework for Smart Electric Vehicle Charging,” *IEEE Transactions on Smart Grid*, 2021. DOI: 10.1109/TSG.2021.3074437.

Z. J. Lee, S. Sharma, D. Johansson, and S. H. Low, “ACN-Sim: An Open-Source Simulator for Data-Driven Electric Vehicle Charging Research,” arXiv:2012.02809 [cs, eess], Dec. 2020, arXiv: 2012.02809. [Online]. Available: <http://arxiv.org/abs/2012.02809>

Chapter 7

PRICING EV CHARGING SERVICES

In Chapter 6, we developed the Adaptive Scheduling Algorithm (ASA), which allocates available power capacity to electric vehicles in order to meet operator objectives while respecting physical constraints. In this chapter, we examine the dual problem of allocating *costs* among users of a charging system. This task is complicated by commercial electricity tariffs, which include both time-varying energy costs (\$/kWh) and demand charge (\$/kW). Time-of-use (TOU) energy costs incentivize shifting energy use to off-peak periods, while demand charges incentivize consumers to smooth their demand profile, reducing infrastructure costs. These demand charges can make up a significant portion of a charging facility's total electricity bill, e.g., up to 90% in the case of DC fast charging [98]. In Section 7.1, we show that demand charge can be up to 75% of the total electricity cost of uncontrolled level-2 charging and up to 49% even when EVs are scheduled optimally to reduce costs. Currently, most workplace and public-use charging facilities are either free or charge a flat rate based on time or energy delivered. As EV adoption grows, the current practice is not sustainable.

In this chapter, we propose a novel method to price charging services, which attributes the operator's total cost to each user based on her contribution to the social cost. Our method is based on three ideas. First, the system operator's primary objective is to meet EVs' energy requests by their departure times without overloading the charging infrastructure and at minimal cost (energy cost plus demand charge). Second, since users do not directly control when their EV is charged in a managed charging environment, the price the user pays should be based on the lowest cost charging schedule the operator could have used, rather than the actual charging schedule used. This gives systems operators flexibility by decoupling charging decisions from pricing decisions while holding them accountable to provide low-cost charging to their users. It also rewards users for providing useful flexibility, even if the operator chose not to utilize that flexibility. Third, to properly assign prices to cover costs, billing should take place at the end of the utility company's billing period, when demand charges and the full impacts of congestion are known.

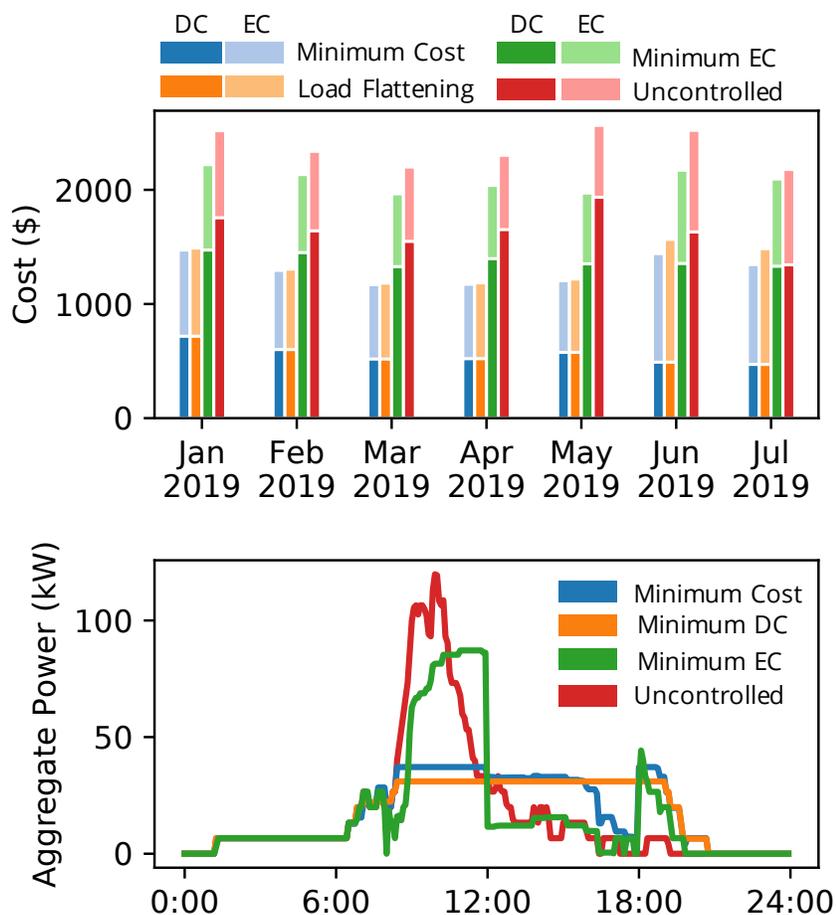


Figure 7.1: (top) Total electricity costs, broken down by demand charge (DC) and energy cost (EC), for the Caltech ACN when using SCE's TOU-EV-4 tariff schedule for various scheduling strategies. Note that all schedules here are calculated offline with perfect future information. All algorithms, except uncontrolled, respect the infrastructure constraints of California Garage 01. (bottom) Example of aggregate power draw for each approach for May 1, 2019.

7.1 Demand Charge

For many commercial customers, demand charge can be a significant portion of their electricity bill. This is especially true for large-scale charging systems at sites where demand tends to be synchronized, such as workplaces. With uncontrolled charging, this would lead to large peaks in demand followed by long periods of low utilization, which leads to extremely high demand charges relative to the amount of energy delivered. This can be seen in Fig 7.1. High demand charges from uncontrolled charging must be passed on to drivers in the form of higher per-unit prices. These higher prices could dissuade customer adoption of EVs.

Managed charging can help reduce demand charges. However, there is an inherent

trade-off between minimizing demand charge and minimizing time-of-use energy costs. This trade-off occurs because demand charge is minimized when load is flat across the day, while energy cost is minimized when load is concentrated in low-cost periods. To see this, consider three possible schedules for EV charging, each with its own objective: 1) minimize energy cost only, 2) minimize demand charge only (load flattening), and 3) minimize total cost. To find each schedule, we solve an offline convex optimization problem with the appropriate objective as well as constraints and data collected from the California Garage 01 ACN. We consider the Southern California Edison EV TOU-4 tariff described in Table 6.2. The total cost, broken down by energy cost and demand charge, of each optimal schedule as well as uncontrolled charging, is shown in Fig 7.1.

The results exhibit two distinctive features. First, uncontrolled charging and energy-cost minimization result in much higher peaks than demand-charge minimization and total-cost minimization. As a consequence, uncontrolled charging (energy-cost minimization) results in up to 335% (258%) increase in demand charge relative to the minimum cost schedule.¹ Second, even in the cases where we minimize the total cost or flatten load, the resulting demand charge is still between 31-49% of the total cost. These results suggest:

- Because demand charges are significant, charging system operators cannot ignore them and must pass these costs on to their users. However, because demand charges are assessed over the whole month, care must be taken to properly attribute these costs to individual charging sessions in a fair and principled way.
- Since managed charging offers the potential to reduce overall costs significantly, the pricing scheme should reward drivers who provide flexibility to the system if that flexibility can be used to reduce overall costs.

Our goal is to design a pricing scheme and an online scheduling algorithm with these properties.

7.2 Pricing rule

In this section, we present the basic design of our pricing method. While we calculate prices offline with perfect information such as would be available at the end of a

¹Note that uncontrolled charging has a higher peak as it is not subject to electrical infrastructure constraints.

billing period, we explain in Section 7.3 how to combine this pricing scheme with online scheduling algorithms so that it can be used in practice.

Basic pricing design

Consider the problem of pricing EV charging service over an entire month in an offline setting. We divide the month into T control periods indexed by $t = 1, \dots, T$. For our experiments we will consider periods of length 5 minutes. Suppose there are N EVs requiring charging service throughout the month. We will abuse notation and use T and N also to denote the sets $T := \{1, \dots, T\}$ and $N := \{1, \dots, N\}$ respectively. Let

- EV $i = 1, \dots, N$ be specified by $(a_i, d_i, e_i, \bar{r}_i(t))$ where $a_i \in T$: is its arrival time, $d_i \in T$ is its departure time, e_i is its energy request. For simplicity, we express energy in kWp defined as the energy delivered by charging at 1 kW for 1 period, i.e., 1/12 kWh for 5 min period. $\bar{r}_i(t)$ is a possibly time-varying upper bound on the charging rate (in kW). Here $\bar{r}_i(t)$ is assumed known and in practice can be a limit imposed by the charger (EVSE) serving EV i , the car's battery management system, some other algorithm, or a combination of these. Note that while in the rest of this thesis the units for r and \bar{r} are in amps instead of kW, we assume nominal voltage in the system, so there is a linear mapping between the two.
- $\kappa(t)$ be the possibly time-varying electricity prices (in \$/kWh) at time $t \in T$ that the operator pays the local utility company for energy. For simplicity, we will implicitly convert $\kappa(t)$ into \$/kWp.
- P be the demand charge defined by the utility (\$/kW).
- $c_l(t)$ be the possibly time-varying capacities of bottlenecks $l = 1, \dots, L$, at time t .
- A_{li} be the coefficient which relates the charging rate of EV i to the aggregate current that is bound by bottleneck l . In the simplest case this can be 1 if EV i is constrained by bottleneck l , 0 otherwise. In more complex three-phase systems, this could be the linearized constraints proposed in Chapter 2. For simulations in this paper we use the latter.

We assume $e_i > 0$, $\bar{r}_i(t) > 0$, $\kappa(t) > 0$, $P > 0$, $A_{li} \geq 0$, $c_l(t) > 0$. Given these parameters, the operator will determine the charging rates $r := (r_i(t), i \in N, t \in T)$ for every EV i at time t .

To provide the EV charging service, the operator needs to pay for both energy and demand charge (among other expenses). These costs are a function of the charging rates r :

$$C(r) := \sum_t \kappa(t) \sum_i r_i(t) + P \max_t \sum_i r_i(t) \quad (7.1a)$$

Hence the operator is interested in solving the following minimum-cost charging problem:

$$\begin{aligned} C^{\min} &:= \min_{r_i \in R_i} C(r) \\ \text{s. t.} & \sum_t r_i(t) = e_i, \quad \forall i \end{aligned} \quad (7.1b)$$

$$\sum_i A_{li} r_i(t) \leq c_l(t), \quad \forall l, \forall t \quad (7.1c)$$

$$r_i(t) \leq \bar{r}_i(t), \quad \forall i, \forall t \quad (7.1d)$$

where

$$R_i := \{r_i \in \mathbb{R}^T : r_i(t) \geq 0; r_i(t) = 0 \text{ for } t < a_i \text{ or } t > d_i\}$$

Here (7.1b) ensures every EV's energy request e_i is met before its departure time d_i , (7.1c), (7.1d) ensures that the capacity limits of the network $c_l(t)$ and charger $\bar{r}_i(t)$ are respected. We assume problem (7.1) is feasible. An optimal solution specifies a schedule that meets all EV energy demands safely and at minimum cost to the operator.

Introduce the auxiliary variable q that represents the peak demand and convert the problem into the equivalent form:

$$C^{\min} := \min_{r_i \in R_i, q \geq 0} \sum_t \kappa(t) \sum_i r_i(t) + Pq \quad (7.2a)$$

$$\text{s. t.} \quad \sum_t r_i(t) = e_i \quad \forall i \quad (7.2b)$$

$$\sum_i A_{li} r_i(t) \leq c_l(t) \quad \forall l, \forall t \quad (7.2c)$$

$$r_i(t) \leq \bar{r}_i(t) \quad \forall i, \forall t \quad (7.2d)$$

$$q \geq \sum_i r_i(t) \quad \forall t \quad (7.2e)$$

Let $\alpha := (\alpha_i, \forall i)$, $\beta := (\beta_{lt}, \forall l, \forall t)$, $\gamma := (\gamma_{it}, \forall i, \forall t)$, $\delta := (\delta_t, \forall t)$ be the Lagrange multipliers for (7.2b), (7.2c), (7.2d) (7.2e) respectively. The dual of the optimization problem (7.2) is

$$\max_{\substack{\alpha, \beta \geq 0 \\ \delta \geq 0, \gamma \geq 0}} \sum_i e_i \alpha_i - \sum_{t,l} c_l(t) \beta_{lt} - \sum_{t,i} \bar{r}_i(t) \gamma_{it} \quad (7.3a)$$

$$\text{s. t. } \kappa(t) + \sum_l A_{li} \beta_{lt} + \gamma_{it} + \delta_t \geq \alpha_i \quad \forall i, \forall t \quad (7.3b)$$

$$P \geq \sum_t \delta_t \quad (7.3c)$$

Pricing rule. Let (r^*, q^*) and $(\alpha^*, \beta^*, \gamma^*, \delta^*)$ be an optimal primal-dual solution to the minimum-cost charging problem (7.2), (7.3). LP duality implies the following observations at optimality.

1. *Network congestion price β_{lt}^* .* We interpret β_{lt}^* as the congestion price at bottleneck l at time t . From (7.2c), this congestion price is zero, i.e., $\beta_{lt}^* = 0$, if bottleneck l is not congested at time t , i.e., $\sum_i r_i^*(t) < c_l(t)$.
2. *Charger congestion price γ_{it}^* .* We interpret γ_{it}^* as the congestion price at charger i at time t . From (7.2d), this congestion price is zero, i.e., $\gamma_{it}^* = 0$, if EV i is charged at lower than the peak rate allowed by the charger, i.e., $r_i^*(t) < \bar{r}_i(t)$.
3. *DC price δ_t^* .* We interpret δ_t^* as the demand charge price at time t . From (7.2e), the price is nonzero, i.e., $\delta_t^* > 0$, only if the total charging rate at time t is at its peak (see Theorem 1), i.e., $\sum_i r_i(t) = \max_\tau \sum_i r_i^*(\tau)$.

For each EV i at each time t , define a *composite price* $\pi_i^*(t)$:

$$\pi_i^*(t) := \underbrace{\kappa(t)}_{\text{energy}} + \underbrace{\sum_l A_{li} \beta_{lt}^*}_{\text{network congestion}} + \underbrace{\gamma_{it}^*}_{\text{charger congestion}} + \underbrace{\delta_t^*}_{\text{demand charge}} \quad (7.4)$$

This EV-specific time-varying price $\pi_i^*(t)$ incorporates the energy price $\kappa(t)$ that the operator pays the utility, the congestion prices β_{lt}^* at all bottlenecks used by EV i , the congestion price γ_{it}^* at the charger, and the demand-charge price δ_t^* . It captures the social cost that EV i is responsible for at time t . Since $\kappa(t) > 0$, $A_{li} \geq 0$ by assumption, the composite prices $\pi_i^*(t) > 0$ for all i, t .

Given the primal-dual solution pairs, the *pricing rule* is

- EV i pays $\pi_i^*(t)r_i^*(t)$ at each time $t \in [a_i, d_i]$ it charges.
- Total payment for EV i 's session is

$$\Pi_i^* = \sum_t \pi_i^*(t)r_i^*(t) \quad (7.5)$$

Clearly the payment $\Pi_i^* > 0$ since $\pi_i^*(t) > 0$. This payment covers energy cost, congestion rents, and demand charge for which EV i is responsible. Having defined the above costs and pricing rules, we present Theorem 1 on the consequences of these costs and pricing rules.

Theorem 1. *Suppose EV i charges at the optimal rates $r_i^* := ((r_i^*(t), t \in [a_i, d_i]))$ and pays Π_i^* given in (7.5). Then*

1. *Decomposition of DC price P . P is decomposed into DC price δ_t^* at each time t :*

$$P = \sum_t \delta_t^*$$

Moreover the DC price $\delta_t^ > 0$ only if $\sum_i r_i^*(t) = \max_\tau \sum_i r_i^*(\tau)$, i.e., only if the total charging rate at time t hits the its peak. Thus $Pq^* = \sum_t \delta_t^* \sum_i r_i^*(t)$.*

2. *Equivalent session price α_i^* . EV i 's total payment satisfies:*

$$\Pi_i^* := \alpha_i^* \cdot e_i$$

i.e., the total payment of EV i is equivalent to charging i only a time-invariant price α_i^ per unit of energy. Moreover $\alpha_i^* > 0$.*

3. *Nonnegative operator surplus. The total payment by all EVs exceeds the total electricity cost (energy + demand charge) that the operator pays the utility:*

$$\sum_i \Pi_i^* \geq C^{min}$$

Proof.

1. Since $e_i > 0$ for all i , we must have $q^* = \max_t \sum_i r_i^*(t) > 0$ and hence $P = \sum_t \delta_t^*$ in (7.3c).

2. The constraint (7.3b) and complementary slackness imply that, for all $t = 1, \dots, T$,

$$\pi_i^*(t) \geq \alpha_i^* \quad \text{with} \quad \pi_i^*(t) = \alpha_i^* \quad \text{if} \quad r_i^*(t) > 0$$

This implies (using (7.2b)):

$$\Pi_i^* := \sum_t \pi_i^*(t) r_i^*(t) = \sum_t \alpha_i^* r_i^*(t) = \alpha_i^* \cdot e_i$$

As noted above $\Pi_i^* > 0$ for all i . Hence $\alpha_i^* > 0$ since $e_i > 0$ by assumption.

3. Assertion 2 implies

$$\begin{aligned} \sum_i \Pi_i^* &= \sum_i e_i \alpha_i^* \\ &\geq \sum_i e_i \alpha_i^* - \sum_{t,l} c_l(t) \beta_{lt}^* - \sum_{t,i} \bar{r}_i(t) \gamma_{it}^* \\ &= D(\alpha^*, \beta^*, \gamma^*, \delta^*) = C^{\min} \end{aligned}$$

where $D(\alpha^*, \beta^*, \gamma^*, \delta^*)$ is the optimal dual objective value and the last equality follows from strong duality.

This completes the proof of the theorem. \square

Remark 1.

1. *The equivalent session price $\alpha_i^* > 0$ is the Lagrange multiplier associated with the (equality) energy constraint (7.2b). This EV-specific time-invariant energy price takes into account of energy, congestions, and demand charge. It lower bounds the composite prices at all times, i.e., $\pi_i^*(t) \geq \alpha_i^*$, but it alone determines the total payment $\Pi_i^* = \alpha_i^* e_i$. Hence, instead of charging EV i at each time t at the time-varying price $\pi_i^*(t)$, we can instead charge i a session price α_i^* based only on energy e_i delivered. It is in this sense that α_i^* is an equivalent session price for EV i .*
2. *Property (7.6) states that EV i pays a nonzero amount at time t , i.e., $r_i^*(t) > 0$, only if the composite price is at its lower bound, i.e., only if $\pi_i^*(t) = \alpha_i^*$ at time t .*
3. *From assertion 3 of Theorem 1 the operator surplus*

$$\sum_i \Pi_i^* - C^{\min} = \sum_{t,l} c_l(t) \beta_{lt}^* + \sum_{t,i} \bar{r}_i(t) \gamma_{it}^*$$

is a measure of how congested the bottlenecks and the chargers are (congestion rents). The higher the surplus is, the more congested the system is, and the surplus is zero if and only if no bottleneck nor charger is ever congested ($\beta_{lt}^* = 0$ and $\gamma_{it}^* = 0$ for all l, t). The demand charge price δ_i^* does not directly affect the site host surplus.

Separation of pricing and control. Note that under the pricing rule (7.5), EV i pays an amount $\sum_t \pi_i^*(t) r_i^*(t)$ for its service, even if it is *not* charged at rates $r_i^*(t)$ at time t . Indeed charging rates in practice are often determined through other means, e.g., using an online scheduling algorithm that does not have perfect future information or even solving a different optimization problem that has a different objective function and a different set of constraints. However, by using this price structure, operators are incentivized to schedule EVs at as low a cost as possible so that the revenue provided by users is enough to cover their costs. If, however, the operator chooses to determine charging schedules in some other way, the user is indifferent so long as their energy demand is fully met.

Pricing with onsite solar

Onsite solar generation can be used to reduce both the environmental footprint and overall cost of an EV charging system. However fairly distributing this savings and incentivizing drivers to provide enough flexibility so that the system can charge their vehicles using solar generation can be challenging. We can easily modify our pricing scheme to account for solar generation by introducing two additional variables $r_i^g(t)$ and $r_i^s(t)$ such that $r_i(t) = r_i^g(t) + r_i^s(t)$. We can interpret $r_i^g(t)$ to be the portion of EV i 's charging rate which was delivered from the grid while $r_i^s(t)$ is the portion delivered by onsite solar generation. We can then formulate the optimization as:

$$\min_{\substack{r_i^g \in R_i \\ r_i^s \in R_i \\ q \geq 0}} \sum_t \kappa(t) \sum_i r_i^g(t) + \sum_t p_t^s \sum_i r_i^s(t) + Pq \quad (7.6a)$$

$$\text{s. t.} \quad \sum_t (r_i^g(t) + r_i^s(t)) = e_i \quad \forall i \quad (7.6b)$$

$$\sum_i A_{li} (r_i^g(t) + r_i^s(t)) \leq c_l(t) \quad \forall l, \forall t \quad (7.6c)$$

$$r_i^g(t) + r_i^s(t) \leq \bar{r}_i(t) \quad \forall i, \forall t \quad (7.6d)$$

$$q \geq \sum_i r_i^g(t) \quad \forall t \quad (7.6e)$$

$$S(t) \geq \sum_i r_i^s(t) \quad \forall t \quad (7.6f)$$

where $S(t)$ is given and specifies the total solar generation available for EV charging at time t . This could be the total production of the PV array or the excess after all other loads are met. Here we assume that energy generated by the onsite solar PV has a price, $p_t^s \geq 0$ (such as from a power purchase agreement). We can then formulate the dual problem, introducing the additional dual variable $\epsilon := (\epsilon_t, \forall t)$ for (7.6f).

$$\begin{aligned} \max_{\substack{\alpha \\ \beta \geq 0 \\ \delta \geq 0 \\ \gamma \geq 0 \\ \epsilon \geq 0}} \quad & \sum_i e_i \alpha_i - \sum_{t,l} c_l(t) \beta_{lt} - \sum_{t,i} \bar{r}_i(t) \gamma_{it} - \sum_t \epsilon_t S(t) \end{aligned} \quad (7.7a)$$

$$\text{s.t.} \quad \kappa(t) + \sum_l A_{li} \beta_{lt} + \gamma_{it} + \delta_t \geq \alpha_i \quad \forall i, \forall t \quad (7.7b)$$

$$p_t^s + \sum_l A_{li} \beta_{lt} + \gamma_{it} + \epsilon_t \geq \alpha_i \quad \forall i, \forall t \quad (7.7c)$$

$$P \geq \sum_t \delta_t \quad (7.7d)$$

Let (r^{g*}, r^{s*}, q^*) and $(\alpha^*, \beta^*, \gamma^*, \delta^*, \epsilon^*)$ be the optimal primal/dual solution to (7.6)/(7.7). The charging rate for each EV is $r_i^*(t) := r^{g*}(t) + r^{s*}(t)$.² We define a composite price for each component of the charging current:

$$\pi_i^{g*}(t) := \underbrace{\kappa(t)}_{\text{energy}} + \underbrace{\delta_t^*}_{\text{demand charge}} + \underbrace{\sum_l A_{li} \beta_{lt}^*}_{\text{network congestion}} + \underbrace{\gamma_{it}^*}_{\text{charger congestion}} \quad (7.8)$$

$$\pi_i^{s*}(t) := \underbrace{\kappa^s(t)}_{\text{solar price}} + \underbrace{\epsilon_t^*}_{\text{solar congestion}} + \underbrace{\sum_l A_{li} \beta_{lt}^*}_{\text{network congestion}} + \underbrace{\gamma_{it}^*}_{\text{charger congestion}} \quad (7.9)$$

We then define the total price paid by driver i as

$$\Pi_i^{sol*} = \sum_t \pi_i^{g*}(t) r_i^{g*}(t) + \pi_i^{s*}(t) r_i^{s*}(t) \quad (7.10)$$

Theorem 1 and its implications extend directly to the case with onsite solar.

²It is important to note that the decomposition into r^{g*}, r^{s*} is purely for accounting purposes, as we cannot control the source of power to each EV.

Theorem 2. Suppose EV i charges at the optimal rates $r_i^* := r_i^{g*} + r_i^{s*} = ((r_i^*(t), t \in [a_i, d_i])$ and pays Π_i^{sol*} given in (7.10). Then

1. Decomposition of DC price P . Suppose $\exists(i, t)$ s.t. $r_i^{g*}(t) > 0$. P is decomposed into DC price δ_t^* at each time t :

$$P = \sum_t \delta_t^*$$

Moreover the DC price $\delta_t^* > 0$ only if $\sum_i r_i^{g*}(t) = \max_\tau \sum_i r_i^{g*}(\tau)$, i.e., only if the total charging rate from the grid at time t hits the peak. Thus $Pq^* = \sum_t \delta_t^* \sum_i r_i^{g*}(t)$.

2. Equivalent session price α_i^* . EV i 's total payment satisfies:

$$\Pi_i^{sol*} := \alpha_i^* \cdot e_i$$

i.e., the total payment of EV i is equivalent to charging i only a time-invariant price α_i^* per unit of energy. Moreover $\alpha_i \geq 0$.

3. Nonnegative operator surplus. The total payment by all EVs exceeds the total electricity cost (energy + demand charge + solar value) that the operator pays:

$$\sum_i \Pi_i^{sol*} \geq C_{sol}^{min}$$

4. Savings with onsite solar. The total cost to the operator with onsite solar is not greater than the total cost without it.

$$C^{min} \geq C_{sol}^{min}$$

Proof.

1. Since by assumption $\exists(i, t)$ s.t. $r_i^{g*}(t) > 0$ we must have $q^* = \max_t \sum_i r_i^{g*}(t) > 0$ and hence $P = \sum_t \delta_t^*$.
2. The constraints (7.7b) and (7.7c) and complementary slackness imply:

$$\begin{aligned} \pi_i^{g*} &\geq \alpha_i^* && \text{with } \pi_i^{g*} = \alpha_i^* \text{ if } r_i^{g*}(t) > 0 \\ \pi_i^{s*} &\geq \alpha_i^* && \text{with } \pi_i^{s*} = \alpha_i^* \text{ if } r_i^{s*}(t) > 0 \end{aligned}$$

This implies (using (7.6b)):

$$\begin{aligned}\Pi_i^{sol*} &:= \sum_t \pi_i^{g*}(t) r_i^{g*}(t) + \pi_i^{s*}(t) r_i^{s*}(t) \\ &= \sum_t \alpha_i^* (r_i^{g*}(t) + r_i^{s*}(t)) = \alpha_i^* \cdot e_i\end{aligned}$$

3. Assertion 2 implies

$$\begin{aligned}\sum_i \Pi_i^{sol*} &= \sum_i e_i \alpha_i^* \\ &\geq \sum_i e_i \alpha_i^* - \sum_{t,l} c_l(t) \beta_{lt}^* \\ &\quad - \sum_{t,i} \bar{r}_i(t) \gamma_{it}^* - \sum_t \epsilon_t \mathcal{S}(t) \\ &= D_{sol}(\alpha^*, \beta^*, \gamma^*, \delta^*, \epsilon^*) = C_{sol}^{min}\end{aligned}$$

where $D_{sol}(\alpha^*, \beta^*, \gamma^*, \delta^*)$ is the optimal dual objective value and the last equality follows from strong duality.

4. Let (r^*, q^*) be an optimal solution of (7.2) yielding an optimal value C^{min} . By inspection, $(r_i^g := r_i^*, r_i^s := 0, q^*)$ is a feasible point for (7.6) which achieves C^{min} , implying $C_{sol}^{min} \leq C^{min}$.

This completes the proof of the theorem. □

7.3 Online scheduling

While the prices are computed in an offline setting with perfect information at the end of each month based on users' collective behavior, charging decisions must be made online with limited information. For example, the operator does not know when future EVs will arrive nor what their energy demands will be. Instead, EVs arrive at random times, but when an EV arrives, it informs the operator, e.g., using a mobile app, its energy demand e_i and departure time $d_i \in T$. We assume in this paper that e_i and d_i provided by the drivers are accurate. In practice, they are not but can be learned from historical data for workplace charging; see Chapter 8.

Online scheduling algorithm

The goal then of the operator at time k is to approximate the offline minimum cost solution with only the information available at that time. Since in the online

setting we cannot optimize over the entire billing period, we instead employ the ASA algorithm proposed in Chapter 6.

In order to maximize profit, we use the objective:

$$U^{\text{CM}} := u^{\text{EC}} + u^{\text{DC}} + 10^{-4}u^{\text{QC}} + 20u^{\text{NC}_1}$$

Recall the u^{EC} is the energy cost. Since we use a non-completion penalty, we set $\pi = 0$. u^{DC} is the demand charge component. We set the demand charge proxy, $\hat{P} = P/(D_p - d)$, and the peak estimate q' to 75% of the optimal peak of the previous month. For discussion on these parameters, see Section 6.2. We also include u^{QC} as a regularizer which promotes charging quickly. Finally, u^{NC_1} , is a non-completion penalty based on the 1-norm. We assign a high enough weight to this term to ensure that all energy demands are met if it is feasible to do so. Because we consider a linearized constraints set, we also replace 6.2d with 7.1d.

Prices in the online setting

While scheduling is done online, we still calculate prices offline at the end of the month using (7.2) and (7.3) as in Section 7.2. Note that in this case, e_i in (7.3) is the actual energy delivered, which may differ from the user's energy request. This pricing scheme is similar to other services such as electricity, water, and phone bills, which are calculated based on usage at the end of the month rather than as the service is used. However, we note that this scheme is more volatile than most other bills faced by consumers owing to the challenges of demand charge. In future work, we hope to provide predictions and bounds of the cost faced by consumers when they input their parameters.

7.4 Simulations

We use ACN-Data and ACN-Sim to explore several questions about the proposed pricing scheme and online scheduling algorithm. The code and data for these experiments are available at https://github.com/zach401/pricing_ev_charging_service.

- Assuming the pricing scheme proposed in Section 7.2, how much surplus would the operator receive if they were able to schedule charging offline in order to minimize costs (solving (7.2))?

- How much surplus would the operator receive if they instead schedule according to the online algorithm proposed in Section 7.3 with the same pricing scheme?
- Does the proposed pricing scheme result in reasonable prices on a per session and per user basis?

In summary, our results are:

- On average, when charging vehicles according to the offline optimal charging schedule r^* , the Caltech ACN would receive \$101 in surplus each month, while JPL would receive a surplus of \$324.
- As expected, in the offline case, revenue from energy costs ($\sum_{t,i} \kappa(t)r_i^*(t)$) matches the actual energy cost and revenue from demand charges ($\sum_{t,i} \delta_i^* r_i^*(t)$) matches the actual demand charge.
- When using the online algorithm proposed in Section 7.3, costs for operating the Caltech ACN increase by an average of 9.2%, while costs for JPL increased by 6.5%.
- On average, with the online algorithm, Caltech has a surplus of -\$18 per month, while JPL has a surplus of \$92.
- We find that the basic pricing scheme described in Section 7.2 can lead to high prices for some sessions, particularly those involved in setting the peak demand for the month. The maximum price for any one session was \$1.62 / kWh at Caltech and \$1.53 / kWh at JPL out of 12,049 and 15,509 sessions, respectively.

The absolute value of the surpluses being small is a desirable feature since the design objective is not profit maximization. It suggests our pricing design is approximately revenue adequate, and sites are sized appropriately.

Simulation setup

For each simulation, we used data collected from Jan 1, 2019 - Aug 1, 2019 at the Caltech and JPL ACNs. We adopt the Southern California Edison TOU rate schedule for separately metered EV charging systems between 20-500 kW, shown

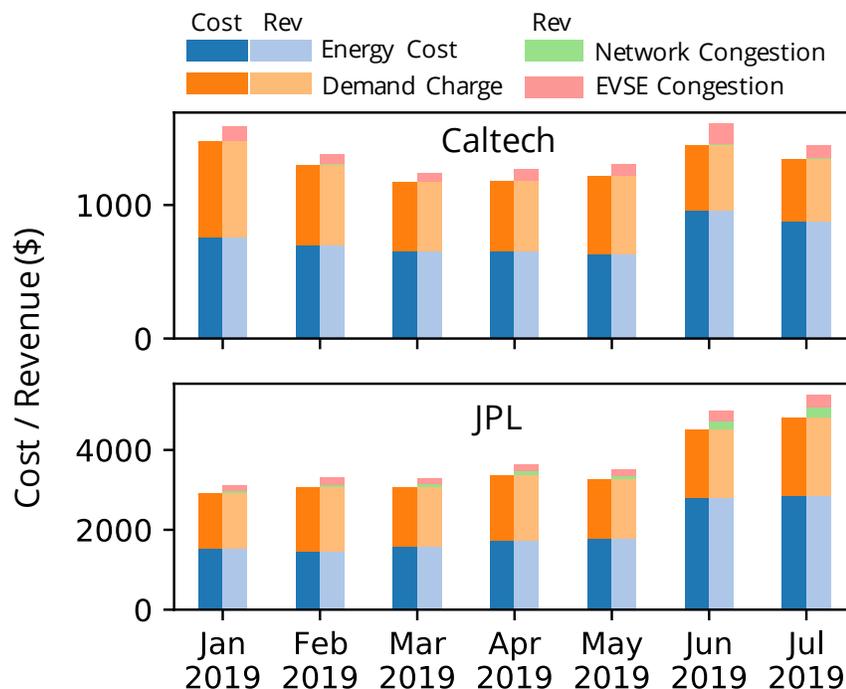


Figure 7.2: Comparison of the total cost of providing charging services in the offline case (shown in darker colors to the left) to the total revenue of the operator (in lighter colors to the right).

in Table 6.2 [96]. Here, Summer runs Jun 1 - Oct 1. While this rate structure was recently replaced with the TOU-EV-8 structure [20], which temporarily removed demand charges for EV charging, this exception is slated to be removed in 5 years. So it is still relevant to investigate pricing schemes under the old tariff structure.

Operator surplus offline

To investigate the upper bound on the surplus an operator can expect, we solve (7.2) and (7.3) for each month and each site. We then calculate the actual cost of electricity for each scenario, as well as the revenue paid to the operator. The results of this experiment are shown in Fig. 7.2. In each case, we split the total cost and total revenue into its constituent components.

From this plot, we can see that the revenue for energy and demand charge perfectly match their respective costs. Thus in the offline case, the operator passes on these costs to users, as we expected. All surplus comes from congestion rents on the network and EVSEs. Note that there is very little rent for network congestion for Caltech, as the charging network has been over-provisioned for future growth. On the other-hand, JPL experiences significant network congestion in the summer

Table 7.1: Offline Operator Surplus by Month

	Jan	Feb	March	April	May	June	July
Caltech	\$114	\$82	\$64	\$93	\$95	\$165	\$99
JPL	\$190	\$267	\$215	\$284	\$250	\$472	\$594

Table 7.2: Online Operator Surplus by Month

	Jan	Feb	March	April	May	June	July
Caltech	-\$13	-\$3	-\$10	-\$105	-\$38	\$94	-\$54
JPL	\$-7	\$68	\$194.84	-\$56	-\$93	\$187	\$353

months. Note that the magnitude of congestion rents are higher in the summer when there is higher price variability. This reflects how congestion rents allow (7.3b) to hold with equality despite variability in $\kappa(t)$ and δ_t . Table 7.1 shows a more detailed breakdown of the surplus for each site. In total, Caltech had a surplus of \$711 for these seven months, while JPL had a surplus of \$2,270. Since in this context, the surplus is a result of congestion; this indicates that JPL should invest these surpluses in increasing the capacity of bottleneck links.

Operator surplus online

Since in practice operators cannot apply the offline optimal schedule that requires future information, they must instead use an online algorithm like the one proposed in Section 7.3. We can then compare the cost of this online algorithm with the revenues generated based on the offline pricing scheme, as shown in Fig. 7.3. We find that this online algorithm results in an average increase of 9.2% and 6.5% in total costs at Caltech and JPL, respectively.

In some cases, such as JPL in July, the surplus is enough to cover this increase in cost. However, in other cases, such as Caltech in July, the costs are higher than the revenue generated, and operator surplus is negative. Table 7.2 shows a breakdown of the operator surplus in each month for each site. From this table, we can see that over these seven months, Caltech lost \$129 while JPL made \$647. This shows that non-negative operator surplus cannot be guaranteed in the online setting with this pricing scheme (even though these surplus and loss are small).

If this possibility of negative surplus is troubling to operators, a simple solution is to distribute any negative surplus as an energy surcharge to be covered by each

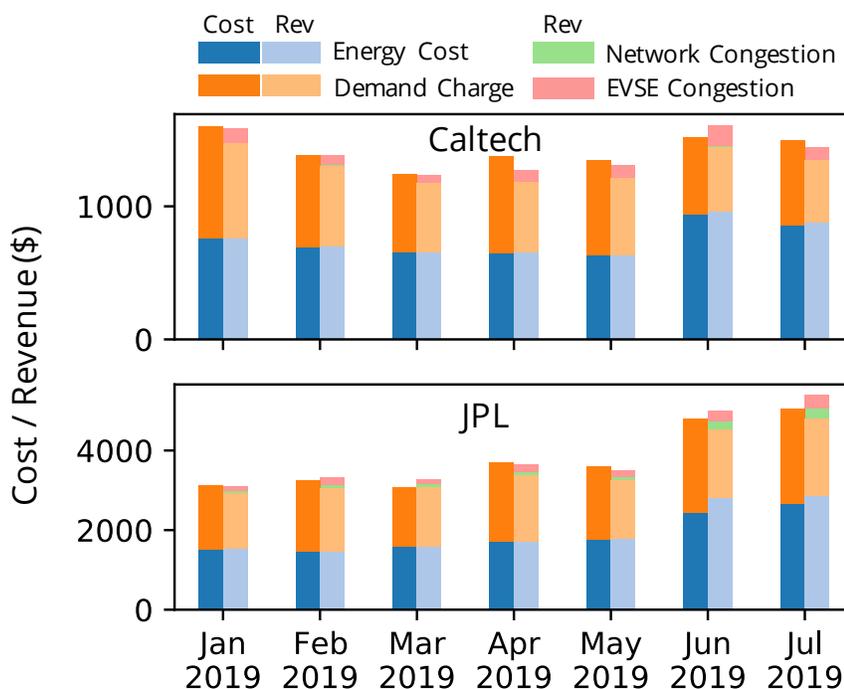


Figure 7.3: Comparison of the total cost of providing charging services in the online case (shown in darker colors to the left) to the total revenue of the operator (in lighter colors to the right).

Table 7.3: Energy surcharges to ensure non-negative surplus (\$/kWh)

	Jan	Feb	March	April	May	June	July
Caltech	0.0013	0.0004	0.0011	0.012	0.0045	-	0.0082
JPL	0.0003	-	-	0.0025	0.004	-	-

user in proportion to the energy they consumed.³ The resulting energy surcharges for each month are shown in Table 7.3. From this table, we can see that only a modest ($< \$0.015$) surcharge on each kWh delivered is necessary to ensure a non-negative surplus. Note that even with this surcharge, the proposed pricing scheme still communicates price signals which align individuals' incentives with those of the group. This is in contrast to a plan which divides all costs by the total energy delivered and provides a flat price to all users.

Distribution of prices

The left panels of Fig 7.4 show the distribution of session prices α_i . From the figure, we see that most sessions ($>95\%$) have α_i very near to or below the maximum energy

³This could also be done with surpluses if the operator does not wish to make a profit or reinvest the surpluses in reducing congestion.

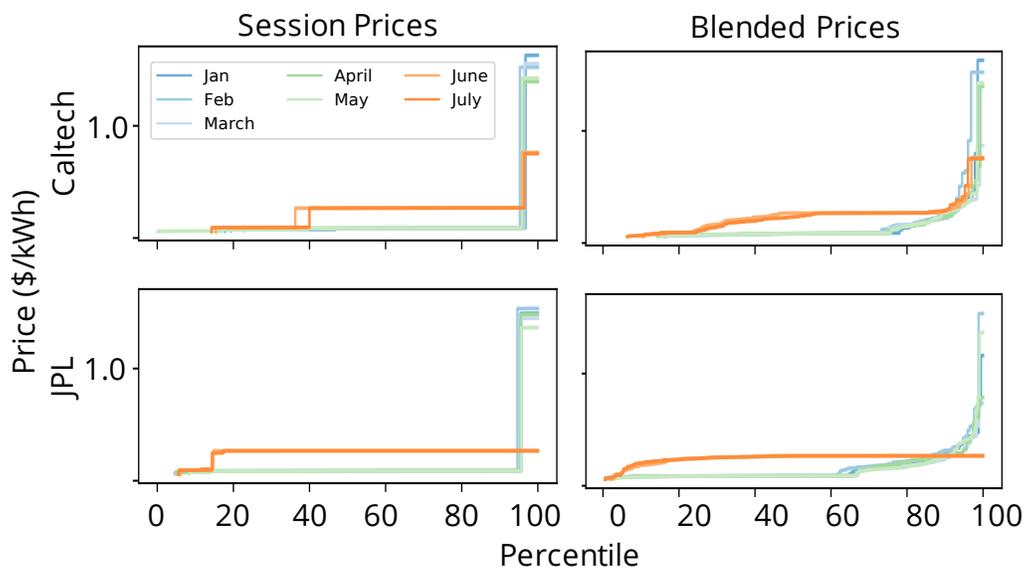


Figure 7.4: Distribution of prices per session (left) and blended per user (right) for each month.

cost for the period (\$0.087 for winter months and \$0.267 for summer months). We also note that in winter months, a small percentage of the total charging sessions have very high prices. Upon close examination, we find that these sessions occur on the same day and are responsible for setting the demand charge for the month. In summer months (June and July), we see that this effect is much less pronounced at Caltech and non-existent at JPL. The likely reason for this is that the higher variation in prices during the summer months causes the scheduling algorithms to concentrate charging during low-cost periods. This means that δ_i^* is non-zero in more periods, spreading the demand charge among more charging sessions.

Since the proposed pricing scheme involves charging users at the end of the month, users are more likely to care about their blended price, defined as the total price paid by the user divided by their total energy received, rather than the price paid for any individual session. The distribution of these blended prices is shown in the right panels of Fig 7.4. From the figure, we see that these blended prices tend to smooth out the price distribution. In addition, we can see that even though some users may experience one or more high-cost sessions throughout the month, for most users these are offset by other lower-cost sessions, lowering their blended price.

7.5 Concluding Remarks

In this chapter, we propose a pricing scheme that assigns a per-session price α_i^* to each charging session that captures the session's effect on energy cost, demand

charge, and system congestion. This scheme has several desirable properties, including guaranteed non-negative operator surplus in the offline setting, and an explicit decomposition to prices on each cost component (energy, demand, and congestion).

We also propose an online scheduling algorithm based on model predictive control, which uses historical information and a demand charge proxy to manage the trade-off between energy costs and demand charge. Using data collected from large-scale charging facilities at Caltech and JPL, we demonstrate that the proposed online scheduling algorithm approximates the offline optimal reasonably well, e.g., the online optimal cost is higher than the offline optimal cost by 9.2% and 6.5% at Caltech and JPL respectively. In the case of JPL, congestion rents are enough to cover this increase in costs, while at Caltech, this results in a negligible average loss of \$18 per month.

While in this work, we consider pricing only at the end of each billing period (1 month), in the future, we hope to build on this scheme to develop online variants. For example, if we can predict session prices when an EV arrives and provide real-time feedback to the driver as they select their energy request and departure time, we can incentivize users to provide more flexibility to the system. Likewise, if final prices can be set when a user leaves, we no longer will need to bill the user at the end of the month. We also plan to address the case where users can provide willingness to pay bids to the system that bound their session price α_i .

Published Works

This chapter contains text and figures originally published in:

Z. J. Lee, J. Z. Pang, and S. H. Low, "Pricing EV charging service with demand charge," *Electric Power Systems Research*, vol. 189, Dec. 2020, issn: 03787796. doi: 10.1016/j.epsr.2020.106694. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0378779620304971>

Chapter 8

DATA-DRIVEN MODELING

In the previous two chapters, we have seen how we can use ACN-Data and ACN-Sim to evaluate algorithms for control and pricing. In both cases, we used real workloads from ACN-Data to evaluate our algorithms. However, there are cases when *statistical models* learned from data can be advantageous over simply using real workloads, such as when we want to evaluate a system before it is built or predict how a system will scale as charging demands increase.

This chapter will demonstrate how we use Gaussian Mixture Models (GMMs) to accurately model charging workloads. Using these models, we first demonstrate, in Section 8.2, that we can predict users' departure time and energy delivered better than the user's own estimates through the mobile app. Then, in Section 8.3, we show that we can use the learned GMM, along with ACN-Sim, to evaluate infrastructure options for a site host. Next, in Section 8.4, we show how we can use our learned models and stochastic optimization to optimally size on-site solar generation to power a smart charging system. Finally, in Section 8.5, we zoom out and use these models to evaluate the potential benefit of smart EV charging to smooth the net demand "duck curve" in California.

8.1 Modeling Charging Workloads

Approaches to Modelling EV Charging Workloads

There are several approaches to modeling EV charging workloads. In [33] and [99], the authors use Kernel Density Estimation (KDE) to approximate the distribution of arrival times, departure times, and energy delivered. While non-parametric models like KDE are very popular, they lack the interpretability of parametric models.

Alternatively, researchers often assume that EVs arrive according to a Poisson process. This model is common in queueing theory and has been applied to EV charging problems [34], [100], [101]. In [67] the authors use ACN-Data to estimate the arrival rate. However, in its basic form, the Poisson process does not account for time-dependent arrivals and departures (though time-varying variants exist). In addition, Poisson processes must be combined with other models to predict workload

features like energy delivered.

In this work, we will instead consider a Gaussian Mixture Model (GMM) [102], [103]). We use the GMM to *jointly* model a tuple (a_i, d_i, e_i) in \mathbb{R}^3 where a_i denotes the arrival time, d_i denotes the duration and e_i is the total energy delivered. We can then draw from this distribution or condition the distribution on a vehicle's arrival time to estimate the duration and energy delivered.

Intuition Behind the Model.

In this context, the Gaussian Mixture Model lends itself to a natural interpretation. We assume that most EV drivers have a finite set of normal routines. For example, a driver might normally plan to arrive at work each morning at 8 AM and remain for 9 hours, leaving at 5 pm. This driver also likely follows a similar route to work, so her energy needs are similar day-to-day. However, this driver's actual arrival time, duration, and energy request will be corrupted by noise such as traffic, so the actual values of (a_i, d_i, e_i) can be seen as a noisy version of the driver's routine. It is also natural to assume that these variables are correlated, i.e., more traffic might mean a later arrival time and higher energy needs. This motivates us to consider modeling the driver's routines as a multivariate Gaussian distribution.

Drivers may also have several routines. For example, a driver may drop their kids off at school some mornings but not others. We can model this by considering a mixture of multivariate Gaussians.

So far, we have considered the model on an individual level. However, we also assume that drivers may share similar routines, i.e., commuting to work from similar areas. This motivates us to consider population-level models with far fewer components (Gaussians) than the number of drivers in the population.

Problem Formulation

Consider a dataset X consisting of N charging sessions. The data for each session $i = 1, \dots, N$, is represented by a triple $x_i = (a_i, d_i, e_i)$ in \mathbb{R}^3 . The data point X_i (we use capital letters for random variables) are independently and identically distributed (i.i.d.) according to some unknown distribution. Since we assume that drivers have finitely many behavior profiles, let K be the number of typical profiles

denoted by μ_1, \dots, μ_K .¹ Each data point X_i can be regarded as a corrupted version of a typical profile with a certain probability. Define a latent variable $Y_i \equiv k$ if and only if X_i is corrupted from μ_k . Moreover, by the i.i.d. assumption, each incoming EV has an identical probability π_k taking μ_k , i.e., $\pi_k := \mathbb{P}(Y_i = k)$ for $i = 1, \dots, N$, $k = 1, \dots, K$. Conditioned on $Y_i = k$, the difference $X_i - \mu_k$ that the profile X_i deviates from the typical profile μ_k can be regarded as Gaussian noise. In this manner, assuming $Y_i = k$, we let $X_i \sim \mathcal{N}(\mu_k, \Sigma_k)$ be a Gaussian random variable with mean μ_k and covariance matrix Σ_k . To estimate the underlying distribution and approximate it as a mixture of Gaussians, it suffices to estimate the parameters $\theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K$. The probability density of observing a data point x can then be approximated using the learned GMM as

$$p(x|\theta) = \sum_{k=1}^K \pi_k \frac{\exp\left(-\|x - \mu_k\|_{\Sigma_k^{-1}}^2 / 2\right)}{\sqrt{(2\pi)^3 \det(\Sigma_k)}}$$

Population and Individual-level GMMs

We train GMMs based on a training dataset X_{Train} and predict the charging duration and energy delivered for drivers in a set \mathcal{U} . The results are tested on a corresponding testing dataset X_{Test} .

The training data from ACN-Data can be divided into two parts: user-claimed data X_C and unclaimed data X_U .

This motivates us to study two different approaches. The first approach generates a population-level GMM (P-GMM) based on the overall training data $X_{\text{Train}} = X_C \cup X_U$. However, users can have distinctive charging behaviors. To achieve better prediction accuracy, we take advantage of the user-claimed data and predict the charging duration and energy delivered for each user. In the second approach, we partition the claimed data into a collection of smaller datasets consisting of the charging information of each user in \mathcal{U} . We write $X_C = \bigcup_{j \in \mathcal{U}} X_j$. We can then train individual-level GMMs (I-GMM) for each user $j \in \mathcal{U}$ by fine-tuning the weights of the components of the P-GMM with data from each of the users to arrive at a final model for each of them.

¹We assume the number K of components is known. In our experiments in Section 8.2, grid search [104] and cross-validation are used to find the best number of components.

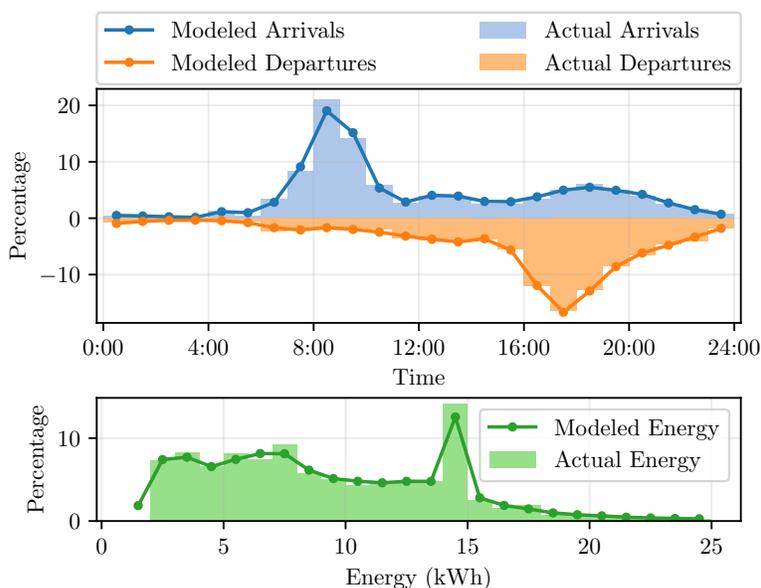


Figure 8.1: Comparison of model distributions with actual data for California_Garage_01 during the training period.

Distribution Learned by P-GMM

To evaluate how well our learned population-level GMM fits the underlying distribution, we gather 100,000 samples from a P-GMM trained on data from the California Garage 01 cluster collected prior to Sep. 1, 2018. We then plot in Figure 8.1 the distribution of these samples along with the empirical distribution from our training set. We choose to plot departure time instead of duration directly as this demonstrates that our model has learned not only the distribution of session duration but also the correlation between arrival time and duration. We see that in all cases, our learned distribution matches the empirical distribution well.

8.2 Predicting User Behavior

This section uses the GMM that we have learned from the ACN-Data dataset to predict a user's departure time and the associated energy consumption based on their known arrival time. Despite recent advances in arrival time based prediction via kernel density estimation [33], [99], [105], simple empirical predictions are commonly used in practical EV charging systems. For example, the ACNs from which this data was collected using user inputs directly in the scheduling problem. Other charging systems simply take the average of the past behavior as a prediction. Our data, however, shows that user input can be quite unreliable, partially because

of a lack of incentives for users to provide accurate predictions. We demonstrate that the predictions can be more precise using simple probabilistic models.

Calculating Arrival Time-based Predictions

Let \mathcal{U} denote the set of users. Suppose a convergent solution $\theta^{(j)} = (\pi_k^{(j)}, \mu_k^{(j)}, \Sigma_k^{(j)})_{k=1}^K$ is obtained for user $j \in \mathcal{U}$ where $\mu_k^{(j)} := (a_k^{(j)}, d_k^{(j)}, e_k^{(j)})$ and the user's arrival time is known *a priori* as $\alpha^{(j)}$. For the sake of completeness, we present the following formulas used for predicting the duration $\delta^{(j)}$ and energy to be delivered $\varepsilon^{(j)}$ as conditional Gaussians of the user $j \in \mathcal{U}$:

$$\delta^{(j)} = \sum_{k=1}^K \bar{\pi}_k^{(j)} \left(d_k^{(j)} + (\alpha^{(j)} - a_k^{(j)}) \frac{\Sigma_k^{(j)}(1, 2)}{\Sigma_k^{(j)}(1, 1)} \right) \quad (8.1)$$

$$\varepsilon^{(j)} = \sum_{k=1}^K \bar{\pi}_k^{(j)} \left(e_k^{(j)} + (\alpha^{(j)} - a_k^{(j)}) \frac{\Sigma_k^{(j)}(1, 3)}{\Sigma_k^{(j)}(1, 1)} \right) \quad (8.2)$$

where $\Sigma_k^{(j)}(1, 1)$, $\Sigma_k^{(j)}(1, 2)$ and $\Sigma_k^{(j)}(1, 3)$ are the first, second and third entries in the first column (or row) of the covariance matrix $\Sigma_k^{(j)}$ respectively. Denoting by $p(\cdot|\mu, \sigma^2)$ the probability density for a normal distribution with mean μ and variance σ^2 , the modified weights conditioned on arrival time in (8.1) and (8.2) above are

$$\bar{\pi}_k := \frac{p\left(\alpha^{(j)}|a_k^{(j)}, \Sigma_k^{(j)}(1, 1)\right)}{\sum_{k=1}^K p\left(\alpha^{(j)}|a_k^{(j)}, \Sigma_k^{(j)}(1, 1)\right)}$$

Error Metrics

We consider both absolute error and percentage error when evaluating duration and energy predictions.

Mean absolute error.

Recall that \mathcal{U} is the set of all users in a testing dataset $\mathcal{X}_{\text{Test}}$. Let \mathcal{A}_j denote the set of charging sessions for user $j \in \mathcal{U}$. The Mean Absolute Error (MAE) is defined in (8.3) to assess the overall deviation of the duration and energy consumption. For a testing dataset $\mathcal{X}_{\text{Test}} = \{(a_{i,j}, d_{i,j}, e_{i,j})\}_{j \in \mathcal{U}, i \in \mathcal{A}_j}$, the corresponding MAEs for

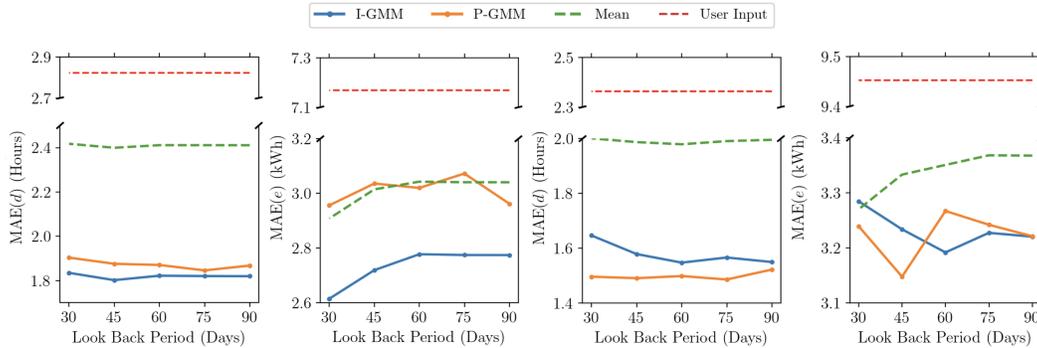


Figure 8.2: Prediction errors for Caltech (left two columns) and JPL (right two columns) for training dataset sizes ranging from 30 days to 90 days in the past. As a benchmark, we consider simply taking the mean of each user’s prior behavior. For comparison, we also include the errors of user inputs. The results are measured by the mean absolute error (MAE) defined in (8.3).

duration and energy are represented by $\text{MAE}(d)$ and $\text{MAE}(e)$ with

$$\text{MAE}(x) := \sum_{j \in \mathcal{U}} \frac{1}{|\mathcal{U}|} \sum_{i \in \mathcal{A}_j} \frac{1}{|\mathcal{A}_j|} |x_{i,j} - \hat{x}_{i,j}| \quad (8.3)$$

where $\hat{x}_{i,j}$ is the estimate of $x_{i,j}$ and $x = d$ or e .

Symmetric mean absolute percentage error.

The Symmetric Mean Absolute Percentage Error (SMAPE) in (8.4) is commonly used (for example, see [99]) to avoid skewing the overall error by the data points wherein the duration or energy consumption is small. The corresponding SMAPEs for duration and energy are represented by $\text{SMAPE}(d)$ and $\text{SMAPE}(e)$ with

$$\text{SMAPE}(x) := \sum_{j \in \mathcal{U}} \frac{1}{|\mathcal{U}|} \sum_{i \in \mathcal{A}_j} \frac{1}{|\mathcal{A}_j|} \left| \frac{x_{i,j} - \hat{x}_{i,j}}{x_{i,j} + \hat{x}_{i,j}} \right| \times 100\% \quad (8.4)$$

Results and Discussion

Experimental setup.

In Figure 8.2, we report $\text{MAE}(d)$ and $\text{MAE}(e)$ for I-GMM and P-GMM on the Caltech dataset as a function of the look back period that defines the length of the training set. Users with more than 20 sessions during Nov. 1, 2018 and Jan. 1, 2019 are included in \mathcal{U} and tested. Note that the size of the training data may not be proportional to the length of periods since there are fewer claimed sessions early in the dataset. The 30-day testing data is collected from Dec. 1, 2018 to Jan. 1, 2019.

We study the behavior of prediction accuracy with different training data sizes by training the GMMs with data collected from five time intervals ending on Nov. 30, 2018 and starting on Sep. 1, 2018, Sep. 15, 2018, Oct. 1, 2018, Oct. 15, 2018 and Nov. 1, 2018 respectively. The GMM components are initialized using k-means clustering as implemented by the Scikit Learn GMM package [104]. Since it is not deterministic, we repeat this initialization 25 times and keep the model with the highest log-likelihood on the training dataset. Grid search and cross validation [104] are used to find the best number of components for each GMM.

Observations.

As observed from Figure 8.2, for the JPL dataset with testing data obtained from Dec. 1, 2018 to Jan. 1, 2019, the 60-day training data gives the best overall performance. This coincides with our intuition that user behavior changes over time and that there is a trade-off between data quality and size. The Caltech dataset also displays this trade-off, with the 30-day training set performing best. This is likely because there was a transition from free to paid charging on Nov. 1, 2018, which meant that data had very different properties before that date.

Hence, for the JPL dataset, we fix the training data as the one collected from Oct. 1, 2018 to Dec. 1, 2018 and show the scatterings of SMAPEs for each session in the testing data (from Dec. 1, 2018 to Jan. 1, 2019) in Figure 8.3. The SMAPEs are concentrated on small values with a few outliers, and high-quality duration prediction has a positive correlation with high-quality energy prediction. As a comparison, user input SMAPEs, shown as Xs, are much worse.

Table 8.1 shows the average SMAPEs for the various methods tested. We show the results using the 30 and 60-day training data for Caltech and JPL, respectively. For reference, we also calculate the error of two additional ways to predict user parameters: 1) we use the mean of the training data X_j as our prediction for each user, and 2) we treat the user input data directly as the prediction. Note that to account for stochasticity in the GMM training process, the results in Figure 8.2 and Table 8.1 are obtained via 50 Monte Carlo simulations.

Implications.

EV users need incentives to provide more accurate predictions. As shown in Figures 8.2, Figure 8.3, and Table 8.1, user input data conspicuously gives the worst

Table 8.1: SMAPEs for Caltech and JPL datasets.

Caltech	I-GMM	P-GMM	Mean	User Input
SMAPE(d)%	15.8543	16.6313	20.4432	25.8093
SMAPE(e)%	14.4273	17.2927	15.9275	27.5523

JPL	I-GMM	P-GMM	Mean	User Input
SMAPE(d)%	12.2500	12.5079	15.8985	18.5994
SMAPE(e)%	12.7318	13.6863	13.3014	26.8769

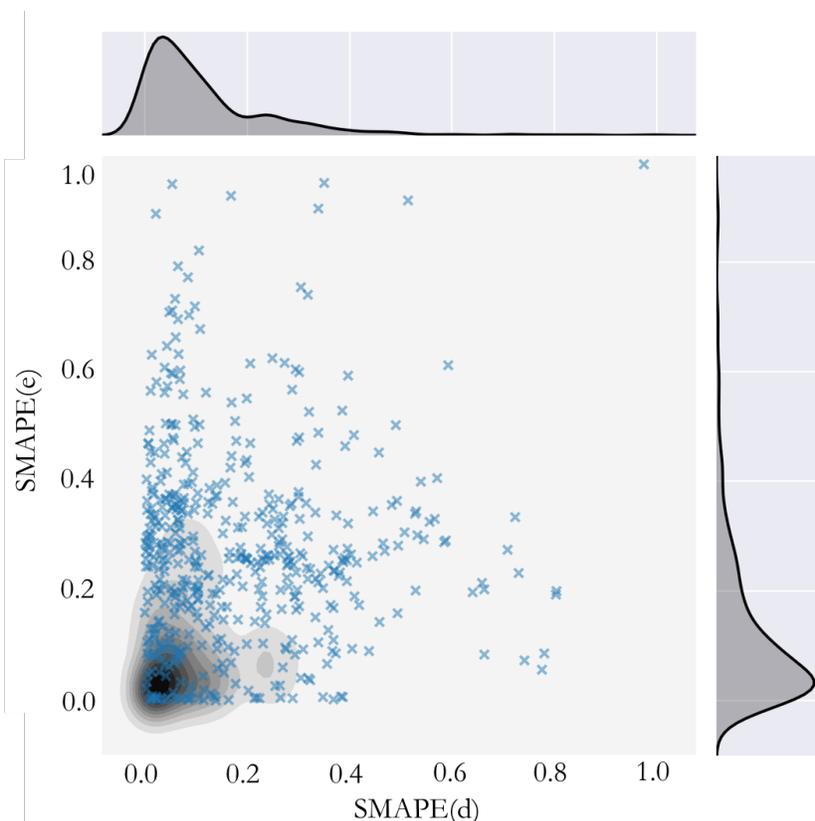


Figure 8.3: Correlation between SMAPE(d) and SMAPE(e) and their marginal distributions for the JPL dataset. Kernel density estimation is used to approximate the joint distribution of the SMAPEs for predicted duration and energy, which is shown as grey shading. The blue crosses represent the corresponding user input SMAPEs (for I-GMM) with respect to each charging session in the testing data set X_{Test} .

overall prediction. However, in some commercial EV charging companies, e.g., PowerFlex, user input data is used as the direct input for the scheduling and pricing algorithms. Therefore, significant improvements can be made in the future by leveraging tools from statistics and machine learning to better predict user behaviors, e.g., using GMMs. In addition, we find that when predicting user behavior, there is a trade-off between training data quantity and quality caused by changing user

behavior over time which must be considered.

8.3 Evaluating Charging System Designs with Data

In previous chapters, we claimed that smart charging could significantly reduce the capital and operating costs of charging systems. While these are general claims, bolstered by experiments like those shown in Chapter 6, we can use ACN-Data and ACN-Sim to quantify these benefits for a particular use case. By enabling this type of analysis, we can help site hosts understand the real savings they can expect using smart charging systems like the ACN in a data-driven approach, rather than using intuition alone.

For this example, we consider a site host who wants to install an EV charging solution at an office building. The host estimates that the system will charge approximately 100 EVs per day. The code for this analysis is available on Github [95].

Infrastructure Options

There are several infrastructure configurations we can consider to meet this demand.

102 Level-1 EVSEs.

One option would be to install enough Level-1 plugs so that each expected driver could plug in and charge during the workday. This was once common advice, as Level-1 outlets provide (and thus require) only 1.9 kW per plug. This means that sites can install far more uncontrolled Level-1 plugs with the same infrastructure capacity than uncontrolled Level-2 chargers, which draw 6.7 kW. For this case, we will consider 102 Level-1 EVSEs (34 per phase), which would require a 200 kW transformer.

102 Uncontrolled Level-2 EVSEs.

While Level-1 chargers are sufficient for many charging needs at long dwell time locations, Level-2 chargers have become the standard for workplace charging. If the site host wants to provide each driver with a Level-2 EVSE, her power needs will increase significantly. 102 uncontrolled Level-2 EVSEs would require a 680 kW transformer.

30 Uncontrolled Level-2 EVSEs.

Because infrastructure upgrades can be prohibitively expensive, especially if a utility service upgrade is necessary, the site host can also consider installing Level-2 EVSEs with the same 200 kW transformer as in the first case. However, without smart charging, this site host will only be able to install 30 Level-2 EVSEs without overloading the transformer.

102 Load Managed Level-2 EVSEs.

An alternative approach is to install 102 Level-2 EVSEs with the 200 kW transformer. Under normal circumstances, this could overload the transformer. However, using a load management system like the ACN, we can use smart charging algorithms like ASA to ensure that the aggregate load of the site is below the transformer capacity. For this example, we will consider ASA-PM, as described in Chapter 6. Because a primary goal, in this case, is to meet all energy demands, we set the value of energy $\pi = 1,000$.

Expected Usage Patterns

We assume that the office will have a usage pattern similar to that of JPL.² As such, we train a Gaussian Mixture Model based on the data collected from weekday usage at JPL. We assume the site will not allow usage on weekends. We then use ACN-Sim's `GaussianMixtureEvents` tool to create a queue of events from this generative model, assuming 100 arrivals on weekdays and 0 on weekends. We also create models of the charging networks described in each proposal. Since EVs are generated, we use the `StochasticNetwork`, which randomly assigns EVs to EVSEs when they arrive.

Evaluation.

We evaluate the scenarios on four criteria:

1. Transformer capacity required
2. Percentage of total energy requested that was delivered

²JPL users pay a fixed price of \$0.10 / kWh, so we will assume that this site will charge a similar (subsidized) price.

Table 8.2: Infrastructure Solution Evaluation (100 EV / Day)

EVSEs	EVSE Level	Algorithm	Capacity (kW)	Swaps / Month	Demand Met	Cost (\$/kWh)
102	Level 1	Unctrl	200	0	75.4%	0.278
102	Level 2	Unctrl	680	0	99.9%	0.351
30	Level 2	Unctrl	200	1,103.5	99.6%	0.256
102	Level 2	ASA-PM	200	0	99.8%	0.234

3. Number of times drivers need to swap spaces to allow others to charge after they finish
4. Operating cost of the system (including energy cost and demand charge)

To evaluate costs, we will use the Southern California Edison (SCE) time-of-use (TOU) rate schedule [96] for separately metered EV charging systems between 20-500 kW, which is shown in Table 6.2 (SCE considers September a summer month).

We repeat these experiments for ten months of generated data, with mean results shown in Table 8.2. Note that the standard deviation between months was less than 3.5% for each metric in each case.

From Table 8.2, we can see that while installing 100 level-1 EVSEs might be the simplest solution, these slow chargers can only meet 75.4% of demand because they cannot support users with large energy needs and short deadlines. However, the alternative of installing a 680 kW transformer and associated service upgrade would be cost-prohibitive for most sites, and installing only 30 level-2 EVSEs requires over 1,100 swaps per month, leading to lost productivity and poor user experience. In this case, the smart charging solution with model predictive control has clear advantages in both capital cost (only requiring a 200 kW transformer), user satisfaction (no swaps are necessary while nearly all user demands are met), and operating costs (having the lowest cost per kWh). This illustrates the benefits of smart charging systems

Example 4. *Space swapping is an important but often overlooked cost of under-sized charging systems. Consider a technology company where workers' time can conservatively be valued at over \$100 / hour. If each swap involved two works and takes 10 minutes, then 1,100 swaps would cost the company \$36,666 / month (1,100 swaps \times 1/6 hrs \times 2 drivers \times \$100 / hr).*

Table 8.3: Infrastructure Solution Evaluation (200 EV / Day)

EVSEs	EVSE Level	Algorithm	Capacity (kW)	Swaps / Month	Demand Met	Cost (\$/kWh)
102	Level 1	Unctrl	200	1,174.5	73.2%	0.244
102	Level 2	Unctrl	680	1,081.5	99.8%	0.327
30	Level 2	Unctrl	200	2,973.9	91.6%	0.233
102	Level 2	ASA-PM	200	1,441.9	87.1%	0.223
201	Level 2	ASA-PM	200	0	98.4%	0.227

Designing for System Expansion.

The benefit of smart charging approaches is amplified as EV adoption grows, and charging infrastructure must scale accordingly. In this scenario, we consider how the system will scale to 200 charging sessions per day. The results are shown in Table 8.3. Intuitively the systems designed for 100 EVs per day require far more swaps with increased demand, and similarly, the percent of demand met decreases. This is also true for the smart charging (ASA-PM) case. However, while scaling the number of EVSEs in traditional uncontrolled charging systems would require a corresponding scaling of the transformer capacity to ensure safety, the smart charging approach allows us to add new EVSEs without increasing the transformer capacity. To enable scalability, we can leave an open space beside each of the originals and install a second EVSE using the same cable. We then use the charging algorithm to ensure the capacity of this cable is not exceeded. In this experiment, we assume the cable was sized for a single EVSE (32 A). However, if the scale-out was planned, the site could have installed a larger cable initially. Thus, we can easily scale the number of EVSEs without increasing transformer or interconnection capacity.

Interestingly, as the number of EVs served by the system increases, the effective cost per kWh decreases for all systems. This indicates the economies of scale, which are associated with demand charge. With more usage, it is possible to spread the demand charge over more energy delivered, decreasing the price per kWh.

8.4 Optimal Sizing of On-site Solar for Workplace Charging

In a recent survey of U.S. drivers [106], 80% of respondents list environmental benefits as their primary reason for purchasing an EV. One way to increase EVs' environmental benefits is to transition from overnight charging, when the generation mix is relatively dirty [107], [108], to mid-day charging at the workplace using on-site solar generation. In addition to these environmental benefits, charging EVs

using solar energy can often be cheaper than purchasing energy from the grid. Accurately understanding this synergy between workplace EV charging and on-site solar generation requires knowledge of user behavior patterns, which determine how well the production of on-site generation can meet EV charging demand.

While many studies have addressed scheduling EV charging to utilize on-site solar generation, it is generally assumed that solar capacity is fixed [57], [60], [109], [110]. In this section, we use ACN-Data and our learned distributions, to address the separate problem of how historical data can be used to optimally size on-site solar generation to minimize the cost of workplace EV charging.

Optimally sizing on-site generation.

Assuming that at operation time EVs will be charged optimally, we can formulate the problem of finding the optimal solar capacity, ζ , of an on-site solar installation as minimizing the following cost function, given a set \mathcal{V} of EVs:

$$U^{sol*}(\zeta, \mathcal{V}) := \mathbf{OPT}(\mathcal{V}, U^{sol}, \hat{\mathcal{R}}) \quad (8.5)$$

where the total charging cost, U_{sol} is defined as

$$U^{sol} := -\kappa_s \sum_{t \in \mathcal{T}} \zeta s(t) - \sum_{t \in \mathcal{T}} \kappa_e(t) L(\zeta, t) - \Delta \cdot \max_{t \in \mathcal{T}} L(\zeta, t) \quad (8.6)$$

$$L(\zeta, t) := \left[\sum_{i \in \mathcal{V}} r_i(t) - \kappa s(t) \right]^+ \quad (8.7)$$

and $\hat{\mathcal{R}}$ is defined in Section 6.1 with the slight modification that the energy constraint (6.2c) is replaced with equality. In (8.6), $s(t)$ is a solar generation profile found by normalizing the time series of solar production over a period by the capacity of the system which generated it. κ_s is the levelized cost of energy (LCOE) for the solar array, which accounts for capital and operating costs. LCOE is calculated assuming that all solar generation is utilized, and therefore we are charged for all solar production regardless of if it is used to charge EVs or not.³ $\kappa_e(t)$ is the possibly time-varying electricity cost from the grid, and Δ is the demand charge levied by the utility based on peak usage throughout a billing period. Here $[a]^+ := \max\{a, 0\}$.

³In many cases, excess solar generation could be used by other loads or sold back to the grid. However, in this example, we consider the simple case where the PV system is connected only to the EVSEs and no net metering is offered.

Given that EV charging will be scheduled optimally according to (8.5), the optimal solar capacity to install is then

$$\zeta^* := \arg \max_{\zeta} \mathbb{E}_{\mathcal{V}}[U^{sol^*}(\zeta, \mathcal{V})]$$

Hence the optimal solar capacity ζ^* minimizes the expected total cost where the expectation is taken over the random set \mathcal{V} of EV arrivals. To compute ζ^* we estimate the expected cost using the empirical mean of scenarios $\{\mathcal{V}_1, \dots, \mathcal{V}_S\}$ sampled from the learned distribution:

$$\mathbb{E}_{\mathcal{V}}[U^{sol^*}(\zeta, \mathcal{V})] \approx \frac{1}{S} \sum_{j=1}^S U^{sol^*}(\zeta, \mathcal{V}_j) \quad (8.8)$$

Computing optimal solar capacity

We first compute the optimal solar capacity using our proposed method for a single month, then evaluate its performance based on actual data.

Consider California Garage 01 at Caltech during September. We will use the time-of-use (TOU) rate schedule shown in Table 6.2 to determine savings. We use an LCOE for solar of \$0.08/kWh, which corresponds to NREL’s SunShot 2020 goal for commercial PV systems, which was met in 2017 [111]. To produce a realistic solar profile $s(t)$, we use NREL’s SAM tool and the National Solar Radiation Database (NSRD) to estimate solar output for a typical meteorological year at Caltech [112]. For this study, we set the length of each discrete time interval in the optimization to be 15 min.

We generate 100 EV charging scenarios using the learned GMMs, each 1 month long. To account for the differences between weekdays and weekends, we train a separate GMM for each, using data from California Garage 01 from May 1 - September 1, 2018. Since the GMMs do not model the number of arrivals each day, we fit a separate Gaussian to predict the number of arrivals using data from the previous month. Once again, we have one model for weekdays and one for weekends. We then generate scenarios by first taking a draw from the appropriate Gaussian to estimate the number of arrivals on the given day and then gathering that many samples from the corresponding GMM. We repeat this procedure until we have accumulated 100 months’ worth of generated data. The optimal solar capacity ζ^* that minimizes the average total cost across these 100 EV scenarios is then computed from (8.5)– (8.8). The result is denoted “Planning” in Table 8.4.

Table 8.4: Evaluating Planned Solar Capacity for Caltech

	Data Source	Solar Capacity (kW)	Percent Solar	Total Cost	Savings
Planning	Synthetic	76 kW	57.5%	\$2,156	-
Evaluation	Real Sept	76 kW	50.0%	\$2,447	\$1,092
Optimal	Real Sept	81 kW	52.3%	\$2,444	\$1,095

When the optimal solar capacity $\zeta^* = 76$ kW is installed, *on average* we expect 57.5% of EV demand to be met by on-site solar generation and the resulting total cost to be \$2,156 a month.

We can then evaluate how this ζ^* chosen at planning time performs using real charging sessions collected from the Caltech ACN in Sept. 2018 and solar data from Sept. 2017 provided through the NSRD, i.e., we evaluate (8.5) –(8.6) with $\zeta^* = 76$ kW. The result is denoted “Evaluation” in Table 8.4. Compared with no on-site solar ($\zeta = 0$ in (8.5) –(8.6)), solar generation could have saved Caltech \$1,092 for that month.

To appreciate how well this performs, suppose we optimize the solar capacity ζ for the real September EV data from California Garage 01 and NSRD solar data from Sept. 2017. To do this, we find the optimal ζ by minimizing (8.5) over ζ with \mathcal{V} being the real data. We then evaluate the performance of this optimally sized array using (8.5) –(8.6). The result, denoted by “Optimal” in Table 8.4, represents a lower bound on the total cost of charging with on-site solar. Compared with the performance of $\zeta^* = 76$ kW on the real September data, the solar capacity and solar coverage are slightly higher. However, the total cost and savings differ by only \$3 for that month. This suggests that the capacity determined at planning time performs very well on the real EV data.⁴

Optimal capacity over a year

The previous example only covers a single month to illustrate our methodology. We now estimate the optimal solar capacity over an entire year. To do so, we first generate a set of scenarios $\{\mathcal{V}_1, \dots, \mathcal{V}_{240}\}$ where each scenario is 1 month long using the procedure outlined in Section 8.4. This is equivalent to 20 years of simulated data. Since solar generation and time-of-use prices depend on the month, we adjust $s(t)$ and $\kappa_e(t)$ for each scenario based on its corresponding month. We then solve

⁴The small change in cost (0.12%) for a relatively large change in capacity (6.2%) is indicative of the shallow slope in the cost function near the minimum, which is helpful for robustness.

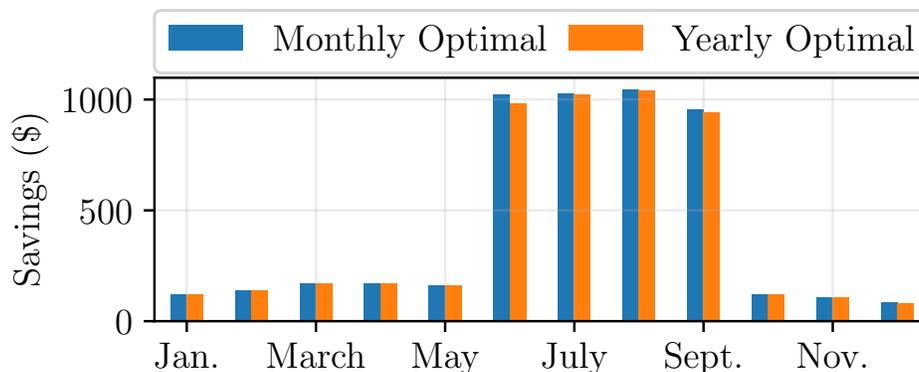


Figure 8.4: Savings achieved versus no on-site solar for each month of the year. We consider two scenarios, one where the solar capacity is allocated optimally for each month and a second where the solar capacity is fixed based on an optimization over the full year.

(8.5)–(8.8) to find the optimal solar capacity over the year and estimate the expected savings. The result for Caltech is denoted by “Yearly Optimal” in Figure 8.4. It shows that most savings are concentrated during the summer months when TOU rates and solar production are high. Over the year, we expect a total savings of \$5,043, \$3,984 of which is from June through September. For reference, we compute the optimal ζ^* for each month individually (as in Section 8.4) and estimate the corresponding savings. The result, denoted by “Monthly Optimal” in Figure 8.4, represents an upper bound on expected savings because it is not practical to change the solar installation month-to-month. This plot shows that the yearly optimal solar capacity ζ^* achieves an expected savings close to this upper bound.

Sensitivity to LCOE

We next illustrate the sensitivity of these benefits to the LCOE of solar, which will continue to fall in the coming years with a goal of \$0.04 / kWh for commercial PV systems by 2030 [111]. We first generate a 20-year collection of scenarios for JPL as we did for Caltech. For each LCOE we solve (8.5)–(8.8) to find the optimal ζ^* over the year for both Caltech and JPL. The expected benefits are shown in Figure 8.5.

These results confirm that as solar prices decrease, there is an increase in the optimal solar capacity, percent of charging demand met by solar, and operators’ savings for both sites. At very low solar costs, we can meet a very high percentage of total charging demand using solar alone, especially at JPL, where users’ schedules align well with solar production. This results in substantial cost reductions for site operators and a significant reduction in the environmental impact of EV charging.

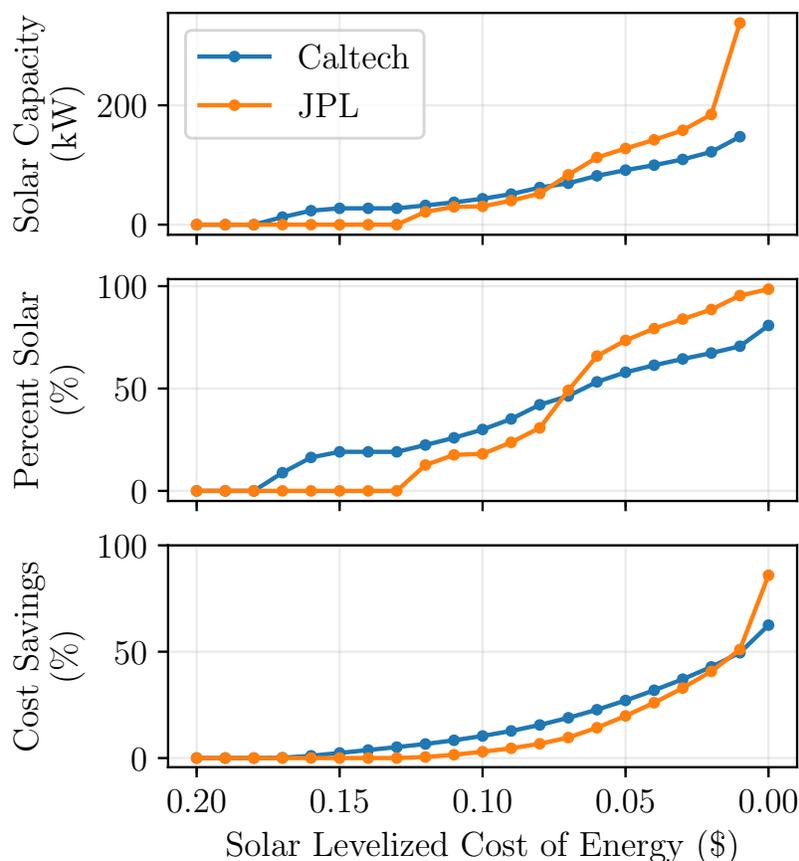


Figure 8.5: Effect of solar levelized cost of energy and site on optimal solar capacity, percent of demand met by solar energy, and cost savings over the no solar case.

Thus as the LCOE for solar decreases, we expect that on-site generation will play an important role in reducing the cost and environmental footprint of workplace EV charging.

The optimal capacities ζ^* at the two sites differ due to differences in their usage profiles. While ζ^* becomes nonzero at Caltech at an LCOE of \$0.17 / kWh, it remains zero at JPL until the cost drops below \$0.12 / kWh. We suspect this is because JPL users tend to arrive earlier than Caltech users, allowing them to be scheduled to avoid on-peak rates during the summer, which reduces the marginal benefit of on-site solar when LCOE is high. In addition, since JPL has almost zero utilization on weekends and receives no compensation for solar generation during that time, it makes less sense to install solar there when LCOE is high.

8.5 Smoothing the Duck Curve

The "duck curve" is a phenomenon in the net demand curve of an area with large amounts of solar energy. As solar generation ramps up in the morning, net demand dips rapidly, with a trough in the early afternoon. The sun setting in the evening then coincides with increased electricity demand as people arrive home from work, leading to an even faster increase in net demand. The phenomenon was first identified by modelers at the National Renewable Energy Laboratory (NREL) in 2008 and subsequently named by the California Independent System Operator (CAISO) for its resemblance to the arched back of a duck, as seen in Fig. 8.6.

The duck curve represents an important challenge for grid operators as more solar generation is integrated into the grid. One key challenge is to ensure that dispatchable generators are flexible enough to follow this net demand curve. This flexibility is expressed as a ramping requirement. CAISO estimates that on a typical spring day, the system will need to be capable of adding 13 GW of generation in under 3 hours [113]. These ramping requirements are challenging for many traditional thermal generators, which are normally used to provide baseload power. In addition to high ramping requirements, the duck curve also depicts the risk of overgeneration. Overgeneration can cause wholesale prices to collapse, sometimes even becoming negative, and require the curtailment of renewable power. Combined, these can severely impact the economics of operating both renewable energy projects and traditional power plants. Smart EV charging has the potential to help with both of these issues, especially when controlled at scale. To accurately estimate the potential of workplace EV charging to alleviate ramping and overgeneration concerns, we need access to realistic data about usage patterns. In this section, we use ACN-Data for this purpose.

Problem Formulation

We formulate the problem of minimizing ramping as an offline optimization problem.

$$\mathbf{OPT}(\mathcal{V}, U_{\text{ramp}}, \hat{\mathcal{R}}) \quad (8.9)$$

where we denote the objective by

$$U_{\text{ramp}} := - \sum_{t \in \mathcal{T}} (N(t) - N(t-1))^2 \quad (8.10)$$

Recall that

$$N(t) := \sum_{i \in \mathcal{V}} r_i(t) + L(t) - G(t) \quad (8.11)$$

where here $L(t) - G(t)$ represents the total net demand of the entire grid, rather than just a single charging facility. For notational simplicity we will refer to $L(t) - G(t)$ as $D(t)$ in this section. For this example, we will only consider the flexibility of EV charging. While we acknowledge that many other flexible loads such as water heaters, appliances, pool pumps, etc. are currently included in $D(t)$ and could be used to aid in smoothing the Duck Curve, we focus our attention on the contribution of electric vehicles and thus treat these loads as fixed.

Qualitative results

To demonstrate the potential of workplace charging to smooth the Duck Curve, we consider a net demand curve for Dec. 11, 2018, from CAISO [114]. We consider three levels of EV penetration in California based on the current number of EVs in California (350,000) and the state's goals for 2025 (1.5 million) and 2030 (5 million). For this case study, we make the optimistic assumption that all of these vehicles would be available for workplace charging. While this assumption is unrealistic, it bounds the potential benefit from above at each level of penetration. Once again, we set the length of each discrete time interval in the optimization to be 15 min.

To reduce the computational burden in solving (8.9) for millions of EVs, we use a representative sample of n EVs drawn from our learned distribution and scale down the net demand curve $\tilde{D}(t)$ from CAISO by the ratio of n to the desired number of EVs, denoted by N . Define $D(t) := (n/N)\tilde{D}(t)$. We then solve (8.9) for $D(t)$ and this representative sample. Finally, we scale the optimal net demand curve, $N^*(t)$, by N/n to arrive at a final curve in the original units. For this experiment, $n = 1,000$.

Figure 8.6 plots the resulting optimal net demand curve $N^*(t)$, for both the Caltech data and the JPL data. Even with only 350,000 EVs, we see a non-trivial smoothing of the net demand curve. With 1.5 million EVs under control, we see a significant filling of the "belly" of the duck as well as a reduction in the morning and afternoon ramping requirements. By the time we reach 5 million EVs under control, we can see an almost complete smoothing of the duck in the JPL case. However, we note that for the Caltech distribution, 5 million EVs lead to a noticeable increase in peak demand. This is because we use the distribution for free charging, which includes

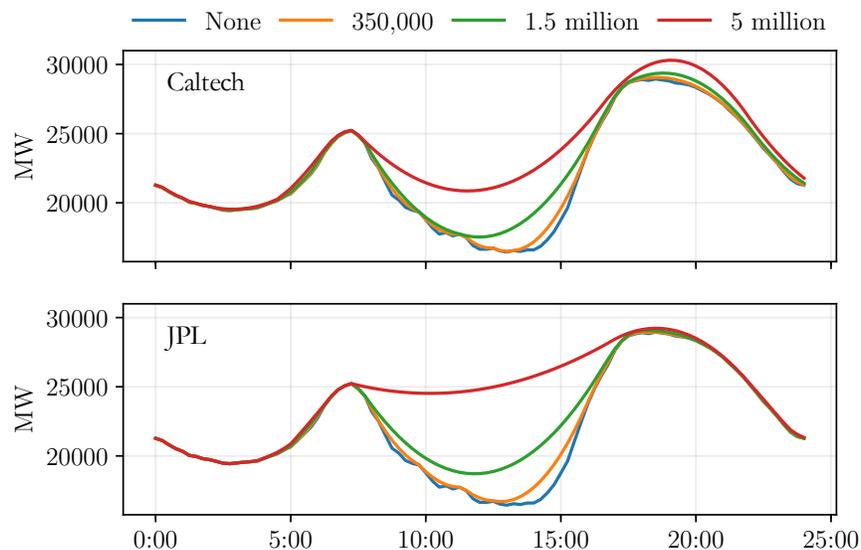


Figure 8.6: Net demand curves after optimal smoothing. EV penetration levels are based on the current population of EVs in California and California’s goals for 2025 and 2030. Upper panel: the Caltech data. Lower panel: the JPL data.

a significant number of short sessions that begin around 5-7 pm, thus requiring us to charge these EVs during the peak of background demand. This demonstrates the benefits of concentrating EV charging during normal working hours, for which the JPL distribution is representative.

Quantitative results

To examine quantitatively how much we can smooth the Duck Curve, we vary the number of EVs under control from 10,000 to 10 million for each distribution. We optimally schedule each group of EVs using (8.9) and measure the resulting maximum up and down ramps and the peak demand. The results are shown in Figure 8.6. Surprisingly, we find that with as few as 2 million EVs under control, we can cut up and down ramping requirements by nearly 50% with only a 0.6% increase in peak demand when using the JPL distribution. This is encouraging as JPL is closest to what we expect for workplace charging.

8.6 Concluding Remarks

This chapter shows several examples of how the ACN Research Portal enables *data-driven* research. Using ACN-Data, we can train statistical models like those described in Section 8.1. Armed with these models, we can predict user behavior,

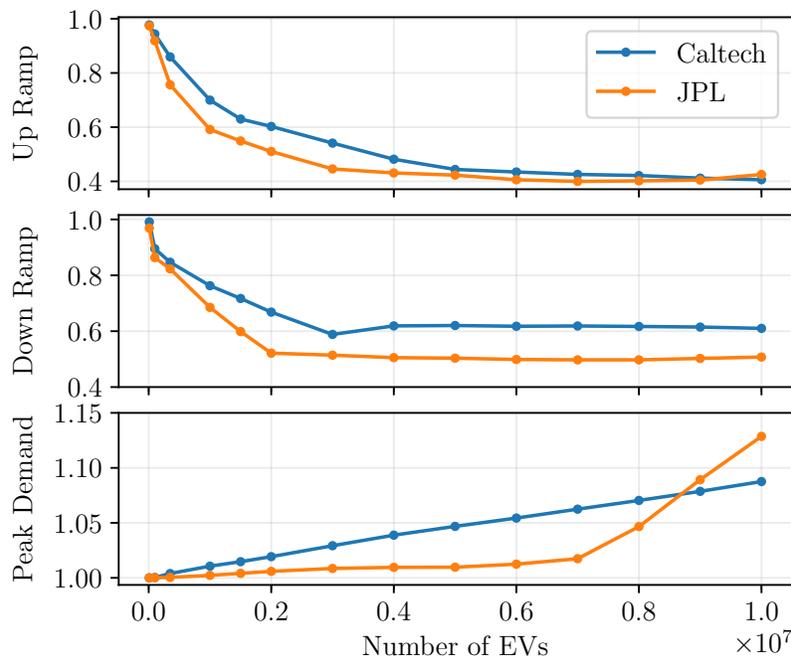


Figure 8.7: 15 minute maximum ramping rates and peak demand relative to the baseline without EVs.

design and evaluate charging systems, and model large-scale grid impacts.

In Section 8.2, we found that our predictive models were more accurate than users' estimates of their departure time and energy needs. However, our model is still far from accurate. There is a need for even more advanced algorithms for predicting user behavior, or alternatively, incentive structures to encourage users to provide more accurate estimates. In either case, we have seen that ACN-Data and ACN-Sim can help researchers by providing data on which to train new predictive models and a platform to evaluate incentive schemes, such as the pricing models proposed in Chapter 7.

In Section 8.3, we demonstrated that the ACN Research Portal can help engineers design and evaluate charging systems. By enabling this approach, we can quantify the benefits of smart EV charging to stakeholders, i.e., site hosts, policymakers, and drivers, and help engineers design more efficient charging systems. For example, we have shown that traditional approaches for uncontrolled EV charging are sub-optimal compared to smart EV charging systems like the ACN in terms of infrastructure required, swaps required, demand met, and cost. While this may seem intuitive, quantifying these benefits for a site host or policymaker can make a huge difference in their decision-making process. The ACN Research Portal allows us to do this

with relative ease and confidence because our simulations are based are backed by real data. Moreover, using the ACN Research Portal, we have shown that a smart charging system can be designed to scale gracefully.

In Section 8.4, we have shown that the ACN Research Portal can go beyond demonstrating benefits to actually making recommendations. Using the statistical models developed in Section 8.1 and stochastic optimization, we can optimally size PV generation in conjunction with smart EV charging. By installing solar and optimizing EV charging with it, site hosts can expect a savings of upwards of \$4,000-\$5,000 / year, even after accounting for the capital costs of the PV array.

Finally, in Section 8.5, we have seen that by controlling smart charging at scale, we can not only minimize the negative impacts of EV charging on the grid, but we can also provide services to help the grid operate more efficiently and integrate more renewable power.

The examples in this chapter are only a sampling of the data-driven work enabled by the ACN Research Portal. We hope that the ACN Research Portal will be a resource to researchers, utilities, companies, and policymakers as they seek to accelerate transportation electrification and unlock the potential of EVs to develop a cleaner, more efficient, and more resilient grid.

Published Works

This chapter contains text and figures originally published in:

Z. J. Lee, T. Li, and S. H. Low, “ACN-Data: Analysis and Applications of an Open EV Charging Dataset,” in Proceedings of the Tenth ACM International Conference on Future Energy Systems, 2019, pp. 139–149, isbn: 978-1-4503-6671-7. doi: 10.1145/3307772.3328313. [Online]. Available: <http://doi.acm.org/10.1145/3307772.3328313>

Z. J. Lee, S. Sharma, D. Johansson, and S. H. Low, “ACN-Sim: An Open-Source Simulator for Data-Driven Electric Vehicle Charging Research,” arXiv:2012.02809 [cs, eess], Dec. 2020, arXiv: 2012.02809. [Online]. Available: <http://arxiv.org/abs/2012.02809>

Part IV

Impact and Future Directions

Chapter 9

CONCLUSIONS

To meet the infrastructure demands of a growing EV population will require us to significantly increase the speed of *research* and *deployment*. However, despite this critical need, there is still a significant gap between the research happening at institutions worldwide and what can be deployed in practice. We refer to this as the *research gap*. To meet our global environmental goals, design a robust power grid, and avoid wasting billions of dollars in unnecessary infrastructure upgrades, we must bridge this gap as soon as possible.

The goal of this thesis has been to develop systems, tools, and algorithms to bridge this gap and accelerate the pace and deployment of smart charging research.

9.1 The Research Gap

To accomplish this goal, we have identified *four* categories of disconnects between existing approaches and what is needed in practice.

Algorithms proposed in the literature cannot be deployed in practice.

We find that many algorithms proposed in the literature are based on models and assumptions that preclude them from practice. While some of these assumptions are minor (such as assuming that a pilot signal can be modeled as a continuous variable), others are more serious, such as ignoring the maximum charging rate of the EV. Other examples include not accounting for infrastructure constraints, especially in unbalanced systems, which we have shown are critical to ensure safety.

Researchers lack access to real data to use data-driven techniques like machine learning and trace-based simulation.

While the field of data science and machine learning has exploded in popularity in recent years, relatively little work has been applied to EV charging. We suspect that this is, in large part, because of a lack of access to real data around EV charging.

Moreover, to make analysis tractable, researchers often have to make unrealistic assumptions about charging workloads. While this enables theoretical analysis to

prove the properties of an algorithm, algorithms should also be tested on realistic workloads to determine if they are suitable for deployment.

Researchers cannot easily share code and experiments for bench-marking.

In many fields, it is customary to benchmark algorithms with a common dataset and environment. However, in most cases with EV charging research, each paper proposes its own models and slightly different problem formulation. Moreover, even if algorithms use similar models, it can be difficult to benchmark algorithms since each study uses a custom simulation environment and different data sets. If researchers want to test against a previous algorithm, they are usually left to implement the algorithm from scratch or port code to work with their custom simulator.

Researchers cannot field test algorithms.

Even for those algorithms which could be deployed in practice, only a fraction will ever control a real vehicle. This is because pilots and field tests are costly and time-consuming. Since stakeholders, like utilities, funding agencies, and investors, often require a pilot before moving forward, this means that many promising approaches die on the vine.

9.2 Tools to Bridge the Gap

In this thesis, we have developed a collection of tools and data, called the Adaptive Charging Network Research Portal, to begin to bridge this gap.

Adaptive Charging Network

The Adaptive Charging Network (ACN) is our system for smart electric vehicle charging. The first ACN was deployed at Caltech in 2016. This system has since been commercialized through PowerFlex Systems and deployed at over 200 sites. Using the ACN, we have collected real data and gained insights into what assumptions hold in practical systems.

Moreover, our scheduling approach, the Adaptive Scheduling Algorithm (ASA), has been deployed on the ACN for over four years, proving that smart charging algorithms can be used in practice.

ACN-Data

ACN-Data is our public dataset taken from ACNs at Caltech, JPL, and an office building in Northern California. The dataset includes over 80,000 charging sessions and 3-years of high-fidelity time series of the system state and control signals. This dataset enables data-driven techniques to be applied to EV charging problems and trace-driven simulations through ACN-Sim.

ACN-Sim

ACN-Sim is our open-source simulation environment which combines real data taken from ACN-Data with models based on our experience operating real charging systems. It enables realistic, trace-based simulations so that researchers can evaluate how their algorithms would perform in a real system. In addition, ACN-Sim provides a common platform for EV scheduling research so that researchers can easily share their experiments and algorithms.

ACN-Live

Finally, ACN-Live is our platform for enabling any researcher to field-test their algorithms. It abstracts the charging system so that researchers can implement their algorithm once and run on both ACN-Sim and a real ACN at Caltech. When ACN-Live is deployed, it will allow researchers who would otherwise never have access to a physical testbed to demonstrate their approach to stakeholders and gather data from a real system.

9.3 Key Lessons

Using these tools, we have uncovered key lessons about smart charging systems.

Unbalance plays a major role in real charging systems.

We have seen that large-scale charging systems can be highly unbalanced (Section 2.4). Because of this, charging algorithms must account for this unbalance or risk overloading infrastructure. This constraint has been overlooked in the EV charging literature. We find, in Section 6.2, that including unbalance in our model can significantly influence our conclusions. In particular, we find that assuming a single-phase or balanced-three-phase model will underestimate the infrastructure requirements for an unbalanced system. In addition, comparisons between algorithms

made on a balanced model do not transfer to an unbalanced one. For example, we find that EDF, LLF, and ASA are near-optimal in the single-phase case. However, in the three-phase case, ASA is still near-optimal, but EDF and LLF underperform.

Current hardware limitations around pilot signal levels can have a major effect on throughput.

We have seen that current charging hardware often supports only a discrete set of setpoints for the pilot signal. In addition, the J1772 standard does not allow for charging rates below 6 A[49], and many vehicles on the market today will not allow charging to pause (meaning the pilot goes to 0 then returns to a non-zero rate).

We find that these hardware limitations can be significant when evaluating algorithms. In Section 6.2, we found that these realistic constraints can reduce throughput by up to 11% for ASA. However, the limitations have a minimal effect on LLF, EDF, and RR. These baseline algorithms support a natural discretization, while ASA requires a post-processing step. Surprisingly, in some highly congested cases, this discretization and minimum charging rate actually improve the performance of EDF and LLF. We believe this is because these constraints lead to a more balanced solution, which increases throughput. While hardware will likely improve to make these constraints less relevant, the 6 A minimum will likely continue.

Model predictive control approaches can approximate offline optimal results in many cases.

Somewhat surprisingly, in the ideal model (without discretization or non-ideal battery behavior), we find that ASA-QC can match the throughput of the offline optimal algorithm. In the task of profit maximization, we find that ASA-PM can achieve 3.6% of the offline optimal. Providing the algorithm with a hint on the optimal peak allows us to close this gap to just 1.9%. Even in more realistic cases, ASA performs with 10% of the optimal in both cases. Note that the offline optimal is calculated for the ideal model.

With smart charging, charging systems can simultaneously reduce capital and operating costs.

We find that assuming realistic charging workloads from ACN-Data, smart charging can significantly reduce capital and operating costs for large-scale charging systems.

In Section 8.3, we find that an ACN with 100 level-2 EVSES running the ASA-PM algorithm would be able to cut its infrastructure needs by 70% while also reducing its cost per kWh by 33%, compared to a conventional unmanaged charging system. We also note that *swapping costs* are an underappreciated cost of under-sizing charging systems. When the number of EV drivers exceeds the number of ports, drivers often interrupt their day to move their vehicle so that someone else can use the charging station. We calculate that the soft cost of lost time could exceed the total cost of electricity by over 450% (\$36,666 / month in swapping cost vs. \$8,122 in energy costs).

Without proper incentives, users will not accurately estimate their departure time nor session energy.

We find that in our dataset, user inputs are not an accurate prediction of the user's session duration nor the energy delivered during their charging session. As we see in Section 8.2, even a simple prediction model is a better predictor than the user's input through the mobile app. We expect one reason for this is that users do not currently have an incentive to provide accurate estimates to the system. The design of these incentive structures is an area of future work, as are more advanced methods for predicting user behavior.

Demand charges play an important role in the overall economics of a charging site.

We find that demand charges can be a significant portion of the overall operating costs of a charging system. Even if smart charging is used to minimize cost, these demand charges can be 31-49% of the overall cost. For uncontrolled charging, the percentage is even higher. How to pass those costs on to users fairly and transparently is an open problem. We propose a promising direction in answering this question in Chapter 7, but acknowledge that there is still work to be done before this pricing scheme could be deployed in practice.

With proper scheduling and forecasting, on-site solar can significantly reduce the overall costs of a charging system.

We find that on-site solar generation can lead to significant reductions in overall costs. Even when accounting for the capital cost of solar through the levelized cost

of energy (LCOE), we find that site hosts could expect over \$5,000 in savings at an LCOE of \$0.08 / kWh, which is achievable today.

With proper scheduling, smart EV charging can avoid distribution grid upgrades, especially when paired with on-site solar.

We find that while uncontrolled charging can lead to spikes in demand which can overload components on the distribution grid and violate voltage bounds. Using ASA with a load flattening objective, we demonstrate that we can mitigate these issues and operate a charging system without upgrades to the distribution network. Moreover, when the site is assumed to have on-site solar generation, we can return the minimum voltage level in the network to its baseline levels.

9.4 Suggestions for System Designers

Based on these lessons, we offer the following advice to charging system designers:

- When designing a charging system, designers should ensure that phases are as balanced as possible. This includes ensuring equal numbers of EVSEs on each phase and that EVSEs assigned to each phase are distributed spatially rather than grouped.
- To unlock the potential of smart charging with oversubscribe infrastructure, careful records of system layout must be taken so that we can form the A matrix. Any mistakes in these records could lead to safety-critical errors. Methods for validating and learning the A matrix from data are an area of future work.
- We find that with algorithms like ASA, 3-4x oversubscription ratios are feasible. However, this is highly dependent on the underlying workload. ACN-Sim can be used to evaluate the feasibility of a design based on the expected workload and infrastructure constraints.
- When using a smart charging system like the ACN, the number of charging ports and the infrastructure capacity are only loosely coupled. Designers should determine port capacity based on the number of vehicles served each day to avoid swapping costs. They can then determine the infrastructure capacity based on the expected energy requests and laxity of sessions at the

site. Even if infrastructure limits, such as grid interconnects, prevent meeting 100% of anticipated demands, installing a port for every vehicle is still best. Algorithms can then determine how to allocate the scarce capacity optimally.

- ACN-Sim and ACN-Data can be used to anticipate costs, predict savings over uncontrolled charging, size infrastructure/solar/storage, and evaluate how the system will scale. See Chapter 8 for examples of these use cases.

9.5 Suggestions for Public Policy

- We find that using smart charging systems, like the ACN, the installation cost per port can be dramatically less than current incentive programs allow. For example, in the Southern California Edison Charge Ready program, the average cost per level-2 port installed was \$14,246 [115]. The average cost per site was \$278,886 (with 20 ports per site being the average). Of this \$278,886, the average infrastructure costs were \$256,328, or 92% of total project costs. In contrast, Adaptive Charging Networks can reduce infrastructure requirements by up to 75%. This results in a 69% reduction in average cost per port, which has been verified in practice by our collaborators at PowerFlex. On the macro-level, achieving a roll-out of 500,000 level-2 charging ports would cost \$7.12 billion using conventional uncontrolled charging, or just \$2.35 billion with load management, a savings of \$4.77 billion.
- We find that smart charging can reduce or remove the need for grid upgrades to support large-scale EV charging. This benefit is magnified when on-site solar generation is considered. Incentives for on-site solar and workplace charging could significantly reduce cost, greenhouse gas emissions, and grid impacts of EV charging.

9.6 Future Directions

In this work, we have begun to bridge the gap between smart charging techniques and the real world. The tools we have created have enabled new lines of inquiry, influenced policy, and spawned a successful company. However, the examples in this thesis are still only a small subset of what is possible with the ACN Research Portal. Directions of future work include:

Expansion of ACN-Sim

Currently, ACN-Sim provides a realistic simulation environment for evaluating algorithms for controlling level-2 charging. However, we have received feedback from users who want to use the simulator to evaluate co-optimization with DC Fast Chargers and other distributed energy resources like stationary storage. We plan to integrate these into future releases of ACN-Sim.

In addition, we plan to pursue tighter integration between ACN-Sim and transportation and grid simulators so that researchers can evaluate how these systems interact through co-simulation.

Another interesting area of research in the ACN-Sim project is how to model strategic user behavior. Our simulations have assumed that users are non-strategic and will not change their behavior in response to charging algorithms or prices. This is likely not the case in practice. Instead, we should model users as strategic agents. This could open up entirely new lines of research where smart charging algorithms must account for this strategic behavior in users. We can also utilize ACN-Live to evaluate if our user models accurately represent the behavior we see in practice.

Deployment of ACN-Live

In this thesis, we have described the design of ACN-Live. Deployment of ACN-Live has been delayed due, in part, to issues caused by the COVID-19 pandemic. We anticipate beginning internal tests of ACN-Live by late 2021. Once ACN-Live is operational, it will turn the Caltech ACN into a first-of-its-kind shared testbed, enabling researchers to evaluate their algorithms on real hardware with minimal overhead.

Incentive Structures to Solicit Accurate Estimations

We have observed that user inputs in the current version of the PowerFlex app are highly inaccurate. We suspect that one reason for this is that users are not incentivized to provide truthful information. The design of pricing and/or control methods which incentivize truthfulness (or penalize inaccuracy) could be an interesting and potentially impactful line of research. ACN-Live will allow us to test these incentive methods on real users to gauge their effectiveness.

Pricing Mechanisms for Fairly Dividing Costs

In Chapter 7, we introduced a novel method for fairly distributing costs in charging systems with time-of-use tariffs and demand charge. We acknowledge that there are problems with the current algorithm which would limit its use in practice. For example, the current algorithm does not reveal prices to users until the end of the month. This means that the pricing scheme cannot effectively influence behavior. This is also a poor user experience, as we find that prices for some sessions can be extremely high.

We are currently exploring two lines of research to alleviate these issues. In the first, we utilize a similar duality approach but use stochastic optimization and machine learning to predict prices when the user arrives. In the second, we allow users to bid into the system with maximum prices they are willing to pay for a certain amount of energy. These bids then become constraints to the optimization.

Load Forecasting with ACN-Data and ACN-Sim.

In many applications, it is helpful to forecast aggregate load into the future. Traditional black-box and statistical approaches rely on access to a historical dataset of time series. However, with ACN-Data and ACN-Sim, we can first define a parameterized workload model, which could be learned from data or designed by an expert. Using ACN-Sim, we can simulate what the aggregate load would be for a given workload and algorithm.

This method offers two key benefits over traditional time-series forecasting methods. First, it allows the analyst to tune the workload. For example, the analyst might want to change the number of vehicles expected per day. To do this with a traditional time series forecast, the analyst might resort to scaling the entire time series, which might violate constraints or result in an unrealistic profile for a given algorithm. Second, this method would allow the analyst to evaluate the effect of different smart charging algorithms on the aggregate load. With this approach, an analyst could use other similar sites to generate a statistical workload model, even if those sites have very different infrastructure constraints or are running a different control algorithm.

BIBLIOGRAPHY

- [1] A. Madrigal, *Powering the Dream: The History and Promise of Green Technology*. Cambridge, MA: Da Capo Press, 2011, ISBN: 978-0-306-81885-1.
- [2] G. Mom, *The Electric Vehicle: Technology and Expectations in the Automobile Age*, eng. Baltimore: Johns Hopkins University Press, 2004, ISBN: 978-0-8018-7138-2.
- [3] C. E. Palmer, “Price Classification of Motor Cars for 1917,” *Scientific American*, vol. 116, no. 1, pp. 30–32, 1917, Publisher: Scientific American, a division of Nature America, Inc., ISSN: 00368733, 19467087.
- [4] J. Larminie and J. Lowry, *Electric Vehicle Technology Explained*. John Wiley & Sons, 2012.
- [5] B. C. Johnson, “Environmental Products that Drive Organizational Change: General Motor’s Electric Vehicle (EV1),” *Corporate Environmental Strategy*, vol. 6, no. 2, pp. 140–150, Jan. 1999, ISSN: 1066-7938. DOI: 10.1016/S1066-7938(00)80024-X.
- [6] J. H. Wesseling, J. C. M. Farla, and M. P. Hekkert, “Exploring car manufacturers’ responses to technology-forcing regulation: The case of California’s ZEV mandate,” *Environmental Innovation and Societal Transitions*, vol. 16, pp. 87–105, Sep. 2015, ISSN: 2210-4224. DOI: 10.1016/j.eist.2015.03.001.
- [7] K. Rajashekara, “History of electric vehicles in General Motors,” *IEEE Transactions on Industry Applications*, vol. 30, no. 4, pp. 897–904, Jul. 1994, ISSN: 1939-9367. DOI: 10.1109/28.297905.
- [8] M. Shnayerson, *The Car That Could: The Inside Story of GM’s Revolutionary Electric Vehicle*, English, 1st edition. New York: Random House, Aug. 1996, ISBN: 978-0-679-42105-4.
- [9] I. E. Agency, *Global EV Outlook 2020: Entering the Decade of Electric Drive?* Jun. 2020.
- [10] M. Woodward, J. Hamilton, B. Walton, J. Ringrow, G. Alberts, S. Fullerton-Smith, and E. Day, “Electric vehicles: Setting a course for 2030,” Deloitte NSE, Tech. Rep., Jul. 2020.
- [11] United States Department of Energy Office of Energy Efficiency and Renewable Energy, *Charging at Home*. [Online]. Available: <https://www.energy.gov/eere/electricvehicles/charging-home> (visited on 04/29/2021).

- [12] R. J. Javid and A. Nejat, “A comprehensive model of regional electric vehicle adoption and penetration,” *Transport Policy*, vol. 54, pp. 30–42, Feb. 2017, ISSN: 0967-070X. DOI: 10.1016/j.tranpol.2016.11.003.
- [13] U. C. Bureau, *data.census.gov*, generated by Zachary J. Lee using data.census.gov. [Online]. Available: <https://data.census.gov/cedsci/table?q=Housing&tid=ACSDP1Y2019.DP04> (visited on 04/29/2021).
- [14] Pecan Street Inc., *Dataport*, 2019. [Online]. Available: <https://dataport.cloud>.
- [15] Z. J. Lee, T. Li, and S. H. Low, “ACN-Data: Analysis and Applications of an Open EV Charging Dataset,” in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, ser. e-Energy '19, 2019, pp. 139–149, ISBN: 978-1-4503-6671-7. DOI: 10.1145/3307772.3328313.
- [16] J. Kwac, J. Flora, and R. Rajagopal, “Household Energy Consumption Segmentation Using Hourly Data,” *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 420–430, Jan. 2014, ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2013.2278477.
- [17] J. Coignard, P. MacDougall, F. Stadtmueller, and E. Vrettos, “Will Electric Vehicles Drive Distribution Grid Upgrades?: The Case of California,” *IEEE Electrific. Mag.*, vol. 7, no. 2, pp. 46–56, Jun. 2019, ISSN: 2325-5897, 2325-5889. DOI: 10.1109/MELE.2019.2908794.
- [18] J. Cross and R. Hartshorn, “My Electric Avenue: Integrating Electric Vehicles into the Electrical Networks,” in *6th Hybrid and Electric Vehicles Conference (HEVC 2016)*, Institution of Engineering and Technology, 2016, pp. 1–6, ISBN: 978-1-78561-294-7. DOI: 10.1049/cp.2016.0972.
- [19] C. Nelder and E. Rogers, *Reducing EV Charging Infrastructure Costs*. Jan. 2020. DOI: 10.13140/RG.2.2.26490.03525.
- [20] S. C. Edison, *Schedule TOU-EV-8 GENERAL SERVICE TIME-OF-USE, ELECTRIC VEHICLE CHARGING, DEMAND METERED*, Mar. 2019. [Online]. Available: https://library.sce.com/content/dam/sce-doclub/public/regulatory/tariff/electric/schedules/general-service-&-industrial-rates/ELECTRIC_SCHEDULES_TOU-EV-8.pdf (visited on 04/21/2021).
- [21] C. o. Pasadena, *City of Pasadena Water and Power Rates Card*, Jun. 2020. [Online]. Available: https://ww5.cityofpasadena.net/water-and-power/wp-content/uploads/sites/54/2020/07/Summary-Rates-2020_07.pdf (visited on 04/29/2021).
- [22] Z. J. Lee, J. Z. Pang, and S. H. Low, “Pricing EV charging service with demand charge,” in *Electric Power Systems Research*, vol. 189, p. 106694, Dec. 2020, ISSN: 03787796. DOI: 10.1016/j.epsr.2020.106694.

- [23] Q. Wang, X. Liu, J. Du, and F. Kong, "Smart Charging for Electric Vehicles: A Survey From the Algorithmic Perspective," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1500–1517, 2016, ISSN: 1553-877X. DOI: 10.1109/COMST.2016.2518628.
- [24] J. C. Mukherjee and A. Gupta, "A Review of Charge Scheduling of Electric Vehicles in Smart Grid," *IEEE Systems Journal*, vol. 9, no. 4, pp. 1541–1553, Dec. 2015, ISSN: 1932-8184, 1937-9234, 2373-7816. DOI: 10.1109/JSYST.2014.2356559.
- [25] C.-Y. Chung, J. Chynoweth, C.-C. Chu, and R. Gadh, "Master-Slave Control Scheme in Electric Vehicle Smart Charging Infrastructure," en, *The Scientific World Journal*, vol. 2014, pp. 1–14, 2014, ISSN: 2356-6140, 1537-744X. DOI: 10.1155/2014/462312.
- [26] M. Zeballos, A. Ferragut, and F. Paganini, "Proportional Fairness for EV Charging in Overload," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6792–6801, Nov. 2019, ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2019.2911231.
- [27] J. A. Stankovic, M. Spuri, K. Ramamritham, and G. Buttazzo, *Deadline Scheduling for Real-Time Systems EDF and Related Algorithms*. 1998, ISBN: 978-1-4615-5535-3.
- [28] Y. Nakahira, N. Chen, L. Chen, and S. H. Low, "Smoothed Least-laxity-first Algorithm for EV Charging," en, in *Proceedings of the Eighth International Conference on Future Energy Systems*, 2017, pp. 242–251, ISBN: 978-1-4503-5036-5. DOI: 10.1145/3077839.3077864.
- [29] Y. Xu, F. Pan, and L. Tong, "Dynamic Scheduling for Charging Electric Vehicles: A Priority Rule," *IEEE Trans. Automat. Contr.*, vol. 61, no. 12, pp. 4094–4099, Dec. 2016, ISSN: 0018-9286, 1558-2523. DOI: 10.1109/TAC.2016.2541305.
- [30] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 940–951, May 2013, ISSN: 0885-8950. DOI: 10.1109/TPWRS.2012.2210288.
- [31] W. Tang and Y. J. Zhang, "A Model Predictive Control Approach for Low-Complexity Electric Vehicle Charging Scheduling: Optimality and Scalability," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1050–1063, Mar. 2017, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2016.2585202.
- [32] G. Zhang, S. T. Tan, and G. G. Wang, "Real-Time Smart Charging of Electric Vehicles for Demand Charge Reduction at Non-Residential Sites," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4027–4037, 2018. DOI: 10.1109/TSG.2016.2647620.

- [33] B. Wang, Y. Wang, H. Nazaripouya, C. Qiu, C.-c. Chu, and R. Gadh, "Predictive Scheduling Framework for Electric Vehicles Considering Uncertainties of User Behaviors," *IEEE Internet of Things Journal*, vol. 4, pp. 52–63, 2017, ISSN: 2327-4662. DOI: 10.1109/JIOT.2016.2617314.
- [34] L. Zhang and Y. Li, "Optimal Management for Parking-Lot Electric Vehicle Charging by Two-Stage Approximate Dynamic Programming," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1722–1730, Jul. 2017, ISSN: 1949-3061. DOI: 10.1109/TSG.2015.2505298.
- [35] Y. Guo, J. Xiong, S. Xu, and W. Su, "Two-Stage Economic Operation of Microgrid-Like Electric Vehicle Parking Deck," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1703–1712, May 2016, ISSN: 1949-3061. DOI: 10.1109/TSG.2015.2424912.
- [36] Z. Xu, W. Su, Z. Hu, Y. Song, and H. Zhang, "A Hierarchical Framework for Coordinated Charging of Plug-In Electric Vehicles in China," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 428–438, Jan. 2016, ISSN: 1949-3061. DOI: 10.1109/TSG.2014.2387436.
- [37] L. Gan, A. Wierman, U. Topcu, N. Chen, and S. H. Low, "Real-time Deferrable Load Control: Handling the Uncertainties of Renewable Generation," in *Proceedings of the Fourth International Conference on Future Energy Systems*, ser. e-Energy '13, New York, NY, USA: ACM, 2013, pp. 113–124, ISBN: 978-1-4503-2052-8. DOI: 10.1145/2487166.2487179.
- [38] J. de Hoog, T. Alpcan, M. Brazil, D. A. Thomas, and I. Mareels, "Optimal Charging of Electric Vehicles Taking Distribution Network Constraints Into Account," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 365–375, Jan. 2015, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2014.2318293.
- [39] Z. J. Lee, S. Sharma, D. Johansson, and S. H. Low, "ACN-Sim: An Open-Source Simulator for Data-Driven Electric Vehicle Charging Research," *arXiv:2012.02809 [cs, eess]*, Dec. 2020, arXiv: 2012.02809.
- [40] T. Bohn, C. Cortes, and H. Glenn, "Local automatic load control for electric vehicle smart charging systems extensible via OCPP using compact submeters," *IEEE*, Jun. 2017, pp. 724–731, ISBN: 978-1-5090-3953-1. DOI: 10.1109/ITEC.2017.7993359.
- [41] T. Bohn and H. Glenn, "A real world technology testbed for electric vehicle smart charging systems and PEV-EVSE interoperability evaluation," in *2016 IEEE Energy Conversion Congress and Exposition (ECCE)*, IEEE, Sep. 2016, pp. 1–8, ISBN: 978-1-5090-0737-0. DOI: 10.1109/ECCE.2016.7854765.
- [42] J. Chynoweth, Ching-Yen Chung, C. Qiu, P. Chu, and R. Gadh, "Smart electric vehicle charging infrastructure overview," in *ISGT 2014*, Washing-

- ton, DC, USA: IEEE, Feb. 2014, pp. 1–5, ISBN: 978-1-4799-3653-3. DOI: 10.1109/ISGT.2014.6816440.
- [43] J. Quiros-Tortos, L. Ochoa, and T. Butler, “How Electric Vehicles and the Grid Work Together: Lessons Learned from One of the Largest Electric Vehicle Trials in the World,” *IEEE Power and Energy Mag.*, vol. 16, no. 6, pp. 64–76, Nov. 2018, ISSN: 1540-7977, 1558-4216. DOI: 10.1109/MPE.2018.2863060.
- [44] J. Quiros-Tortos, L. F. Ochoa, S. W. Alnaser, and T. Butler, “Control of EV Charging Points for Thermal and Voltage Management of LV Networks,” *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 3028–3039, Jul. 2016, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2015.2468062.
- [45] J. Quiros-Tortos, A. N. Espinosa, L. F. Ochoa, and T. Butler, “Statistical Representation of EV Charging: Real Data Analysis and Applications,” in *2018 Power Systems Computation Conference (PSCC)*, Dublin, Ireland: IEEE, Jun. 2018, pp. 1–7, ISBN: 978-1-910963-10-4. DOI: 10.23919/PSCC.2018.8442988.
- [46] P. B. Andersen, S. Hashemi Toghroljerdi, T. M. Sørensen, B. E. Christensen, J. C. M. L. Høj, and A. Zecchino, *The Parker Project: Final Report*, English. Technical University of Denmark, 2019.
- [47] O. Frendo, N. Gaertner, and H. Stuckenschmidt, “Open source algorithm for smart charging of electric vehicle fleets,” *IEEE Trans. Ind. Inf.*, pp. 1–1, 2020, ISSN: 1551-3203, 1941-0050. DOI: 10.1109/TII.2020.3038144.
- [48] ———, “Real-Time Smart Charging Based on Precomputed Schedules,” *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6921–6932, Nov. 2019, ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2019.2914274.
- [49] S. International, *SAE Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler J1772_201710*, 2017. [Online]. Available: https://www.sae.org/standards/content/j1772_201710/.
- [50] N. A. Chaturvedi, K. Reinhardt, J. Christensen, J. Ahmed, and A. Kojic, “Algorithms for Advanced Battery-Management Systems,” *IEEE Control Syst.*, vol. 30, no. 3, pp. 49–68, Jun. 2010, ISSN: 1066-033X, 1941-000X. DOI: 10.1109/MCS.2010.936293.
- [51] H. E. Perez, X. Hu, S. Dey, and S. J. Moura, “Optimal Charging of Li-Ion Batteries With Coupled Electro-Thermal-Aging Dynamics,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 7761–7770, Sep. 2017, ISSN: 0018-9545, 1939-9359. DOI: 10.1109/TVT.2017.2676044.
- [52] F. Kazhamiaka, S. Keshav, C. Rosenberg, and K.-H. Pettinger, “Simple Spec-Based Modeling of Lithium-Ion Batteries,” *IEEE Transactions on Energy Conversion*, vol. 33, no. 4, pp. 1757–1765, Dec. 2018, ISSN: 1558-0059. DOI: 10.1109/TEC.2018.2838441.

- [53] F. Kazhamiaka, C. Rosenberg, and S. Keshav, “Tractable lithium-ion storage models for optimizing energy systems,” *Energy Informatics*, vol. 2, no. 1, p. 4, May 2019, ISSN: 2520-8942. DOI: 10.1186/s42162-019-0070-6.
- [54] O. Frendo, J. Graf, N. Gaertner, and H. Stuckenschmidt, “Data-driven smart charging for heterogeneous electric vehicle fleets,” *Energy and AI*, Aug. 2020, ISSN: 26665468. DOI: 10.1016/j.egyai.2020.100007.
- [55] K. Clement-Nyns, E. Haesen, and J. Driesen, “The Impact of Charging Plug-In Hybrid Electric Vehicles on a Residential Distribution Grid,” *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 371–380, Feb. 2010, ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2009.2036481.
- [56] N. Chen, C. W. Tan, and T. Q. S. Quek, “Electric Vehicle Charging in Smart Grid: Optimality and Valley-Filling Algorithms,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1073–1083, Dec. 2014, ISSN: 1932-4553, 1941-0484. DOI: 10.1109/JSTSP.2014.2334275.
- [57] D. Wu, H. Zeng, C. Lu, and B. Boulet, “Two-Stage Energy Management for Office Buildings With Workplace EV Charging and Renewable Energy,” *IEEE Transactions on Transportation Electrification*, vol. 3, no. 1, pp. 225–237, Mar. 2017, ISSN: 2332-7782. DOI: 10.1109/TTE.2017.2659626.
- [58] A. Ramanujam, P. Sankaranarayanan, A. Vasan, R. Jayaprakash, V. Sarangan, and A. Sivasubramaniam, “Quantifying The Impact of Electric Vehicles On The Electric Grid: A Simulation Based Case-Study,” in *Proceedings of the Eighth International Conference on Future Energy Systems - e-Energy '17*, Shatin, Hong Kong: ACM Press, 2017, pp. 228–233, ISBN: 978-1-4503-5036-5. DOI: 10.1145/3077839.3077854.
- [59] J. Rivera, C. Goebel, and H.-A. Jacobsen, “Distributed Convex Optimization for Electric Vehicle Aggregators,” *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1852–1863, Jul. 2017, ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2015.2509030.
- [60] A. Schuller, C. M. Flath, and S. Gottwalt, “Quantifying load flexibility of electric vehicles for renewable energy integration,” *Applied Energy*, vol. 151, pp. 335–344, Aug. 2015, ISSN: 03062619. DOI: 10.1016/j.apenergy.2015.04.004.
- [61] J. Coignard, S. Saxena, J. Greenblatt, and D. Wang, “Clean vehicles as an enabler for a clean electricity grid,” *Environ. Res. Lett.*, vol. 13, no. 5, p. 054031, May 2018, ISSN: 1748-9326. DOI: 10.1088/1748-9326/aabe97.
- [62] Z. J. Lee, G. Lee, T. Lee, C. Jin, R. Lee, Z. Low, D. Chang, C. Ortega, and S. H. Low, “Adaptive charging networks: A framework for smart electric vehicle charging,” *IEEE Transactions on Smart Grid*, 2021. DOI: 10.1109/TSG.2021.3074437.

- [63] C. Sun, T. Li, S. H. Low, and V. O. K. Li, “Classification of electric vehicle charging time series with selective clustering,” *Electric Power Systems Research*, vol. 189, p. 106695, 2020, ISSN: 0378-7796. DOI: <https://doi.org/10.1016/j.epsr.2020.106695>.
- [64] T. Li, S. H. Low, and A. Wierman, “Real-time Flexibility Feedback for Closed-loop Aggregator and System Operator Coordination,” in *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, Virtual Event Australia: ACM, Jun. 2020, pp. 279–292, ISBN: 978-1-4503-8009-6. DOI: [10.1145/3396851.3397725](https://doi.org/10.1145/3396851.3397725).
- [65] P. Meenakumar, M. Aunedi, and G. Strbac, “Optimal Business Case for Provision of Grid Services through EVs with V2G Capabilities,” in *2020 Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, Monte-Carlo, Monaco: IEEE, Sep. 2020, pp. 1–10, ISBN: 978-1-72815-641-5. DOI: [10.1109/EVER48776.2020.9242538](https://doi.org/10.1109/EVER48776.2020.9242538).
- [66] J. Schlund, M. Pruckner, and R. German, “FlexAbility - Modeling and Maximizing the Bidirectional Flexibility Availability of Unidirectional Charging of Large Pools of Electric Vehicles,” in *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, Virtual Event Australia: ACM, Jun. 2020, pp. 121–132, ISBN: 978-1-4503-8009-6. DOI: [10.1145/3396851.3397697](https://doi.org/10.1145/3396851.3397697).
- [67] C.-A. Lin, K. Shang, and P. Sun, “Wait Time Based Pricing for Queues with Customer-Chosen Service Times,” *SSRN Journal*, 2020, ISSN: 1556-5068. DOI: [10.2139/ssrn.3709573](https://doi.org/10.2139/ssrn.3709573).
- [68] S. Correa, L. Jiao, A. Jakubenas, R. Moyano, J. O. Iglesias, and J. Taneja, “Who’s in Charge Here?: Scheduling EV Charging on Dynamic Grids via Online Auctions with Soft Deadlines,” in *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, Virtual Event Japan: ACM, Nov. 2020, pp. 41–49, ISBN: 978-1-4503-8061-4. DOI: [10.1145/3408308.3427619](https://doi.org/10.1145/3408308.3427619).
- [69] A. Al Zishan, M. M. Haji, and O. Ardakanian, “Adaptive Control of Plug-in Electric Vehicle Charging with Reinforcement Learning,” in *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, Virtual Event Australia: ACM, Jun. 2020, pp. 116–120, ISBN: 978-1-4503-8009-6. DOI: [10.1145/3396851.3397706](https://doi.org/10.1145/3396851.3397706).
- [70] Y. Amara-Ouali, Y. Goude, P. Massart, J.-M. Poggi, and H. Yan, “A review of electric vehicle load open data and models,” Nov. 2020, [Online]. Available: <https://hal.inria.fr/hal-03028375>.
- [71] L. Markram, *Electric Vehicle Charging Stations: Energy Consumption & Savings*, <https://boulder.colorado.gov/open-data/electric-vehicle-charging-stations/>, Accessed: 2020-02-01.

- [72] *Electric Vehicle Charging Station Usage (July 2011 - Dec 2017)*, <https://data.cityofpaloalto.org/dataviews/244892/electric-vehicle-charging-station-usage-july-2011-dec-2017/>, Accessed: 2020-02-01.
- [73] T. Bourdas, *Belib Paris Network of 22 kW AC-DC Accelerated Charging Stations for Electric Vehicles*, <https://www.data.gouv.fr/en/datasets/belib-reseau-parisien-de-bornes-de-recharges-accelerees-22-kw-ac-dc-pour-vehicules-electriques/>, Accessed: 2020-02-01.
- [74] *Transactions of the Charges of SAP Labs France Executive Electric Vehicles*, <https://opendata.reseaux-energies.fr/explore/dataset/conso-ve-sap0/information>, Accessed: 2020-02-01.
- [75] *Electric Vehicle Charging Sessions Dundee*, <https://data.dundeecity.gov.uk/dataset/ev-charging-data>, Accessed: 2020-02-01.
- [76] *ElaadNL Open Datasets for Electric Mobility Research*, https://platform.elaad.io/analyses/ElaadNL_opendata.php, Accessed: 2020-02-01.
- [77] Z. J. Lee, S. Sharma, and D. Johansson, *ACN Portal*, <https://github.com/zach401/acnportal>, Accessed: 2020-01-19.
- [78] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, “MAT-POWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education,” *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb. 2011, ISSN: 0885-8950. DOI: 10.1109/TPWRS.2010.2051168.
- [79] L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, “Pandapower—An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems,” *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018, ISSN: 0885-8950. DOI: 10.1109/TPWRS.2018.2829021.
- [80] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, “PowerModels.JL: An Open-Source Framework for Exploring Power Flow Formulations,” in *2018 Power Systems Computation Conference (PSCC)*, Dublin, Ireland: IEEE, Jun. 2018, pp. 1–8, ISBN: 978-1-910963-10-4. DOI: 10.23919/PSCC.2018.8442948.
- [81] D. Montenegro, R. Dugan, R. Henry, T. McDermott, and W. Sunderm, *OpenDSS - EPRI Distribution System Simulator*. [Online]. Available: <https://sourceforge.net/projects/electricdss/>.
- [82] D. P. Chassin, J. C. Fuller, and N. Djilali, “GridLAB-D: An Agent-Based Simulation Framework for Smart Grids,” 2014.

- [83] S. Saxena, J. MacDonald, and S. Moura, “Charging ahead on the transition to electric vehicles with standard 120V wall outlets,” *Applied Energy*, vol. 157, pp. 720–728, Nov. 2015, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2015.05.005.
- [84] S. Saxena, C. Le Floch, J. MacDonald, and S. Moura, “Quantifying EV battery end-of-life through analysis of travel needs with vehicle powertrain models,” *Journal of Power Sources*, vol. 282, pp. 265–276, May 2015, ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2015.01.072.
- [85] E. S. Rigas, S. Karapostolakis, N. Bassiliades, and S. D. Ramchurn, “EVLlib-Sim: A tool for the simulation of electric vehicles’ charging stations using the EVLib library,” *Simulation Modelling Practice and Theory*, vol. 87, pp. 99–119, Sep. 2018, ISSN: 1569190X. DOI: 10.1016/j.simpat.2018.06.007.
- [86] T. Morstyn, K. A. Collett, A. Vijay, M. Deakin, S. Wheeler, S. M. Bhagavathy, F. Fele, and M. D. McCulloch, “OPEN: An open-source platform for developing smart local energy system applications,” en, *Applied Energy*, vol. 275, p. 115397, Oct. 2020, ISSN: 03062619. DOI: 10.1016/j.apenergy.2020.115397.
- [87] N. Blair, N. DiOrio, J. Freeman, P. Gilman, S. Janzou, T. Neises, and M. Wagner, “System Advisor Model (SAM) General Description (version 2017.9.5),” *National Renewable Energy Laboratory: Golden, CO, USA*, 2018.
- [88] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement Learning: A Survey,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, Jan. 1996. DOI: 10.1613/jair.301.
- [89] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “OpenAI Gym,” *arXiv preprint arXiv:1606.01540*, 2016.
- [90] Z. J. Lee and S. Sharma, *Adacharge*, <https://github.com/caltech-netlab/adacharge>, Nov. 2020.
- [91] S. Diamond and S. Boyd, “CVXPY: A Python-embedded modeling language for convex optimization,” *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.
- [92] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, “A Rewriting System for Convex Optimization Problems,” *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.
- [93] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. B. Hamrick, J. Grout, S. Corlay, *et al.*, “Jupyter Notebooks—A publishing format for reproducible computational workflows.” in *ELPUB*, 2016, pp. 87–90.
- [94] *NATS – Cloud Native, Open Source, High-performance Messaging*. [Online]. Available: <https://nats.io/> (visited on 05/01/2021).

- [95] Z. J. Lee and S. Sharma, *ACN Research Portal Examples*, <https://github.com/caltech-netlab/acnportal-experiments>.
- [96] S. C. Edison, *Schedule TOU-EV-4 GENERAL SERVICE TIME-OF-USE ELECTRIC VEHICLE CHARGING - DEMAND METERED*, Aug. 2017. [Online]. Available: https://library.sce.com/content/dam/sce-doctlib/public/regulatory/historical/electric/2017/schedules/general-service-&-industrial-rates/ELECTRIC_SCHEDULES_TOU-EV-4_2017.pdf (visited on 04/21/2021).
- [97] F. Bu, Y. Yuan, Z. Wang, K. Dehghanpour, and A. Kimber, "A Time-Series Distribution Test System Based on Real Utility Data," in *2019 North American Power Symposium (NAPS)*, 2019, pp. 1–6. doi: 10.1109/NAPS46351.2019.8999982.
- [98] G. Fitzgerald and C. Nelder, *EVGO Fleet and Tariff Analysis Phase 1: California*, Mar. 2017. [Online]. Available: https://rmi.org/wp-content/uploads/2017/04/eLab_EVgo_Fleet_and_Tariff_Analysis_2017.pdf.
- [99] Y.-W. Chung, B. Khaki, C. Chu, and R. Gadh, "Electric vehicle user behavior prediction using hybrid kernel density estimator," in *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 2018, pp. 1–6. doi: 10.1109/PMAPS.2018.8440360.
- [100] R. Carvalho, L. Buzna, R. Gibbens, and F. Kelly, "Critical behaviour in charging of electric vehicles," *New Journal of Physics*, vol. 17, no. 9, Sep. 2015, ISSN: 1367-2630. doi: 10.1088/1367-2630/17/9/095001.
- [101] S. Chen, T. Mount, and L. Tong, "Optimizing Operations for Large Scale Charging of Electric Vehicles," in *2013 46th Hawaii International Conference on System Sciences*, Jan. 2013, pp. 2319–2326. doi: 10.1109/HICSS.2013.435.
- [102] B. G. Lindsay, "Mixture models: Theory, geometry and applications," in *NSF-CBMS regional conference series in probability and statistics*, JSTOR, 1995.
- [103] E. Eirola and A. Lendasse, "Gaussian mixture models for time series modelling, forecasting, and interpolation," in *International Symposium on Intelligent Data Analysis*, Springer, 2013, pp. 162–173.
- [104] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [105] Z. Chen, Z. Zhang, J. Zhao, B. Wu, and X. Huang, "An Analysis of the Charging Characteristics of Electric Vehicles Based on Measured Data and Its Application," *IEEE Access*, vol. 6, pp. 24 475–24 487, 2018.

- [106] A. A. Association, *Consumer Attitudes Electric Vehicles*, 2018. [Online]. Available: <https://publicaffairsresources.aaa.biz/download/10790/> (visited on 01/22/2019).
- [107] X. Chen, C.-W. Tan, S. Kiliccote, and R. Rajagopal, “Electric vehicle charging during the day or at night: A perspective on carbon emissions,” *arXiv:1811.07984 [cs]*, Nov. 2018.
- [108] J. McLaren, J. Miller, E. O’Shaughnessy, E. Wood, and E. Shapiro, “CO 2 emissions associated with electric vehicle charging: The impact of electricity generation mix, charging infrastructure availability and vehicle type,” en, *The Electricity Journal*, vol. 29, no. 5, pp. 72–88, Jun. 2016, ISSN: 10406190. DOI: 10.1016/j.tej.2016.06.005.
- [109] O. Ardakanian, C. Rosenberg, and S. Keshav, “Quantifying the benefits of extending electric vehicle charging deadlines with solar generation,” in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Venice, Italy: IEEE, Nov. 2014, pp. 620–625, ISBN: 978-1-4799-4934-2. DOI: 10.1109/SmartGridComm.2014.7007716.
- [110] S. Lee, S. Iyengar, D. Irwin, and P. Shenoy, “Shared solar-powered EV charging stations: Feasibility and benefits,” in *2016 Seventh International Green and Sustainable Computing Conference (IGSC)*, Hangzhou, China: IEEE, 2016, pp. 1–8, ISBN: 978-1-5090-5117-5. DOI: 10.1109/IGCC.2016.7892600.
- [111] R. Fu, D. Feldman, and R. Margolis, “U.S. Solar Photovoltaic System Cost Benchmark: Q1 2018,” Tech. Rep., Nov. 2018.
- [112] J. M. Freeman, N. A. DiOrio, N. J. Blair, T. W. Neises, M. J. Wagner, P. Gilman, and S. Janzou, “System Advisor Model (SAM) General Description (Version 2017.9.5),” 2018. DOI: 10.2172/1440404.
- [113] CAISO, *What the duck curve tell us about managing a green grid*, Available: https://www.caiso.com/Documents/FlexibleResourcesHelpRenewables_FastFacts.pdf, 2016. (visited on 05/08/2018).
- [114] —, *Today’s Outlook*, Jan. 2019. [Online]. Available: <http://www.caiso.com/TodaysOutlook/Pages/default.aspx> (visited on 01/31/2019).
- [115] Southern California Edison, *Charge Ready Pilot Quarterly Report 4th Quarter, 2020 March 1, 2021*, Mar. 2021. [Online]. Available: <https://www.sce.com/sites/default/files/custom-files/SCE%20Quarterly%20Charge%20Ready%20Pilot%20Report%202020%20Q4.pdf> (visited on 05/02/2021).