Title
Essays on Revenue Management in Urban Mobility

Author
Nyotta, Rajpal Bobby Singh

Publication Date
2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Essays on Revenue Management in Urban Mobility

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Management

by

Rajpal Bobby Singh Nyotta

2021

ABSTRACT OF THE DISSERTATION

Essays on Revenue Management in Urban Mobility

by

Rajpal Bobby Singh Nyotta
Doctor of Philosophy in Management
University of California, Los Angeles, 2021
Professor Fernanda Bravo Plaza, Co-Chair
Professor Melvin Keith Chen, Co-Chair

The essays in this dissertation lie at the intersection of revenue management, urban mobility, and technology. Some of the most well-studied problems in operations management and operations research have been inspired by the transportation sector. For instance, the traveling salesman problem, the vehicle routing problem, freight logistics, airline fleet planning, port operations, and rail scheduling are set in the transportation industry. In this dissertation, we restrict our analysis to urban mobility, which focuses on transportation in metropolitan cities. Urban mobility has evolved dramatically over the past decade due to advances in technology, in particular, the mobile phone. Bike-sharing, ride-sharing, and vehicle sharing are possible today because of the growth and popularity of mobile phones. Because of this growth, users are able to access train and bus schedules in real-time, pay fares, and instantaneously reserve and check-out shared cars, bikes, electric scooters, and other types of shared vehicles. While this accessibility provides users with more flexibility, the systems are also increasingly difficult to operate and manage. One way to address this operational complexity involves using tools and methods from revenue management. More

generally, using price and discounts as levers to shape customer behavior in a way that improves the system's service level, revenue, customer satisfaction, and other key performance metrics.

This dissertation is made up of four essays across three chapters that address questions in operating systems in urban mobility, and we use techniques from revenue management to study how these systems can operate more effectively.

In Chapter 1, we study free-ride policies as a mechanism to incentivize users of a "dockless" or "free-floating" electric vehicle sharing system (EVSS) to park vehicles at charging stations in order to maintain a charged fleet. A balanced system has a fleet that is adequately charged and evenly dispersed throughout the city. If left to unfold naturally, the system would fall out of balance, and revenue and customer experience might suffer. Most sharing systems use manual repositioning to achieve this balance, but we consider pricing incentives as an alternative method. We develop an infinite horizon dynamic program to analyze free-ride policies. We focus on an EVSS that offers free rides to customers if they return vehicles to charging stations. We build on this initial formulation to construct a mixed-integer program that outputs intuitive, battery-threshold rules for when to offer free rides. We also extend the model to accommodate more general discount-based policies. In a discrete-event simulation model using real data from an EVSS, we compare the performance of this simple policy against other sophisticated policies, including the commonly used fine-based policy, which fines users for street-parking vehicles with low battery. We first find that the simple threshold-based policy performs close to a more sophisticated, black-box policy in terms of revenue. We also discover that the free-ride policies generate customer utilities that are ten times higher than fine-based policies, but also generate less revenue. However, free-ride policies can be less costly to implement since they rely on manual repositioning up to 65-75% less than the benchmarking policies. Our simulation reveals this three-dimensional trade-off between customer satisfaction, revenue, and operational complexity. Our results are robust under many demand patterns and under a variety of network settings.

In the remaining chapters, we are motivated by the claim that 30% of metropolitan traffic is a result of individuals searching or "cruising" for parking (Shoup, 2017). It is theorized that this cruising behavior causes superfluous traffic congestion that can be assuaged and mitigated with more effective pricing polices. In particular, pricing policies that ensure there is at least one open spot available on each block at all times under regular demand. With this in mind, Chapters 2 and 3 examine how to develop dynamic pricing policies that both maximize revenue and address traffic congestion, with Chapter 2 focusing on estimating key parameters that feed into the pricing models and Chapter 3 focusing on developing the price optimization models.

In order to develop such pricing policies, one needs to know the price and spatial elasticity of parking, where price elasticity is a measure of the change in demand in response to a price change and the spatial elasticity is a measure of how much money a customer would require to park a mile or a block away from their destination. Using data from our industry partner, a venture-backed technology company that develops a software-as-a-service (SaaS) platform to manage parking, permitting, and micro-mobility for municipalities and organizations throughout the world, we are able to empirically estimate both of these values in Chapter 2. In this chapter, the context is parking-specific and the estimates are unique to the data from our partner city. However, we believe that our approach and the estimates can be used across urban mobility applications, and beyond, as these elasticities are often assumed to be known or given in many classic revenue management problems.

In Chapter 2.1, the first essay of Chapter 2, we estimate the price elasticity after a 20% price increase in a mid-sized U.S. city and find the average price elasticity of parking demand is between -3.42 and -1.57, which is higher than existing estimates (Lehner and Peer, 2019). One reason our study could be producing higher estimates is because, as far as we know, our work is the first to use transactions data from a mobile phone application for parking payments, which is more accurate and detailed than the data used in the existing literature. With our model, we can also measure how long it takes for customers to learn about and

respond to the price change. Despite the price change being publicly advertised, we find that customers do not respond to the price change until they experience it firsthand.

In Chapter 2.2, the second essay of Chapter 2, we estimate the spatial elasticity. We perform our estimation using a panel dataset of parking transactions spanning 21 months in a large U.S. city. During this time frame, there was an unannounced pricing error where two neighboring blocks were discounted by 67% for 16 months. We find that customers require approximately $81 to walk an additional mile to their intended destination. This estimate increases 13% in the presence of rain and 36% during the morning rush hour.

In Chapter 3, the final chapter of this dissertation, we study optimal, dynamic pricing policies for a system, or network, of reusable resources, where a parking spot on a city block can be interpreted as a resource that can be *reused* after it is vacated.

We focus our analysis on a single reusable product (i.e., a single zone or block with a fixed number of parking spaces) and aim to set the price as a function of the number of occupied spaces. Our objective is to maximize the long-run average revenue under Markovian assumptions (i.e., Poisson arrival and exponential usage times). In queuing theory, such a model is known as an Erlang loss system. We reformulate this objective function using a metric that we term the *conditional entry-state distribution*. There does not exist a method for computing this metric, so in Chapter 3, we develop an algorithm that converges to the metric's true value for any Erlang loss system. We also provide analysis on the performance and speed of the algorithm.

The dissertation of Rajpal Bobby Singh Nyotta is approved.

Velibor Mišić

Jake Feldman

Felipe Caro

Melvin Keith Chen, Committee Co-Chair

Fernanda Bravo Plaza, Committee Co-Chair

University of California, Los Angeles

2021

To my mom, Jasvinder Kaur, for teaching me compassion and encouraging me to fearlessly pursue my dreams.

ਮਾਂ ਵਰਗਾ ਮੀਤ ਨਾ ਕੋਈ, ਮਾਂ ਵਰਗੀ ਅਸੀਸ ਨਾ ਕੋਈ

Maa varga meet na koi, maa vargi asees na koi.

There's no friend like a mother, there's no blessing like a mother. –Punjabi Proverb

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am extremely grateful for my academic advisors, colleagues, friends, and family who have supported me during my doctoral journey. I cannot thank you all enough for your patience and encouragement.

I am indebted to Fernanda Bravo who has been vital in my development as a researcher and as a person. Since we began working together, she has been steadfast in her dedication to my success. She taught me how to think critically about challenging problems, and if it was not for her faith in me and her commitment to excellence, I would not have grown as much as I did during my time at UCLA. I feel lucky and blessed to have had her advising me. Keith Chen has been an invaluable resource and 3 of the 4 essays in my dissertation could not have been completed without the data he acquired and conversations we had during research meetings. His questions, comments, and observations were shrewd and perceptive, discussions with him were always exciting, and his energy for research and life is contagious. Jake Feldman has been a true friend and sage every step of the way. He was always accessible via phone to chat about research, offer advice, and brainstorm ideas. I am thankful for all of the time he spent educating me on core revenue management principles. He also taught me to be comfortable working outside of my comfort zone and how to express my ideas clearly, both of which will serve me well as I embark on my career as a researcher. Felipe Caro and Velibor Mišić have both been very helpful with their ideas, suggestions, and mentorship at various stages of my doctoral studies.

I also want to acknowledge the following professors: Elisa Long, for her assistance during her role as Ph.D. Liaison, and for setting an astounding and inspiring example of what it means to be a caring, thoughtful educator. Chris Tang for allowing me to teach and learn from him, for always stepping up as a leader during trying times, and for using his platform to bring awareness to important issues.

Other UCLA faculty and staff have selflessly offered career advice, taught classes, opened

their homes for social events, grabbed lunch, or assisted with funding opportunities; in short, they have made my time at UCLA pleasant.

I need to recognize my fellow students who accompanied me at different points of my Ph.D.: I am grateful to Taylor Corcoran, Kira Stearns, Anna Sáez de Tejada Cuenca, and Sean Bruggeman for being great colleagues and friends. In addition, Paul Rebiez, Sandeep Rath, Ali Fattahi, Prashant Chintapalli, Araz Khobadakhshian, Yi-Chun Chen, Nur Kaynar Keles, Jingwei Zhang, Jiayi Yu, and Dan Yavorsky for their camaraderie. There are many Anderson doctoral students, past and present, who I have not named but have also offered help or engaging conversation over the past several years, and I would not have made it this far without them either.

I would also like to thank the Harold and Pauline Price Center for Entrepreneurship and Innovation and the Easton Technology Management Center for generous financial support.

My friends were a key part of my off-campus experience during my doctoral studies. In particular, I need to give a shoutout to all the homies who have been there for me through thick and thin as I navigated this journey: Sunny and Lucky, the real ਯਾਰ ਬੇਲੀਆਂ, were always there to be a needed distraction; ਉਨ੍ਹਾਂ ਮੁੰਡਿਆਂ ਨਾਲ, ਕਦੇ ਵੀ ਨਹੀਂ "ਇਕ ਪੈੱਗ" ਹੁੰਦਾ. My buddies Nik and Neil were always there to chat via phone and text. My compañero Ethan for thousands of miles of bike rides and saddle banter during the first two years. My chota veer Munny for always being down for ਗੱਪ ਸ਼ੱਪ ਅਤੇ ਇਕ ਦੋ ਠੰਢੇ. There are countless various and nefarious others I could also mention, and I would not have made it through without them too. If you were a part of this journey, I thank you from the bottom of my heart.

Finally, I need to thank my family for their love and everything they did for me as I pursued my Ph.D. My granparents, Bachan Kaur (ਬਚਨ ਕੌਰ) and Late Devinder Singh Mudhar (ਦਵਿੰਦਰ ਸਿੰਘ ਮੁਧੱੜ), have inspired me with their bravery and sacrifices. They are true pioneers and I am honored to be their grandson. My grandfather was my hero, and my Ph.D. gave me the opportunity to forge an even stronger relationship with him and travel with him in his homeland before he passed. For that reason alone, I will always cherish

my time pursuing a Ph.D. My uncle has always been there for me, and my brother (a.k.a ਕਵਿੰਦਰ) inspires me with his resilience, grit, and wisdom. Most importantly, I need to thank my mother. She is the reason I was able to make it through my Ph.D. in one piece. She was there for me during the darkest hours when I wanted to quit and provided me with the never-ending, unconditional love that I needed to persevere. I would not have been able to complete a Ph.D. if it were not for her motivation, optimism, and empathy.

A version of Chapter 1 is currently under review for publication and is co-authored with Fernanda Bravo and Jake Feldman. The work presented in Chapter 2 is in preparation for publication. The modified manuscripts submitted for publication will be co-authored with Fernanda Bravo and Keith Chen. The research in Chapter 3 has been accepted for publication in *Operations Research Letters* and is co-authored with Fernanda Bravo and Keith Chen.

VITA

2006–2010	B.S. Industrial Engineering and Operations Research, Univeristy of California, Berkeley
2010–2011	Operations Research Analyst, Carnival Cruiselines
2011–2012	Consultant, Deloitte & Touche
2012–2013	M.Eng. Operations Research and Information Engineering, Cornell University
2013–2015	Analytics and Optimization, AIMMS
2015–2015	Business Intelligence Engineer, Amazon, Inc.
2015–2019	Anderson Fellowship, UCLA Anderson School of Management
2016–2021	Teaching and Research Assistant, UCLA Anderson School of Management

PUBLICATIONS

Nyotta, B. S., Bravo, F., & Chen, M. K. (2021). Computing the Conditional Entry-state Distribution in Erlang Loss Systems. *Operations Research Letters*, 49(3), 345-349.

CHAPTER 1

Free Rides in Dockless Electric Vehicle Sharing Systems

1.1 Introduction

In "dockless" or "free-floating," electric vehicle sharing systems (EVSSs), a fleet of electric rechargeable vehicles (i.e. cars, vespas, bicycles, or scooters) are scattered throughout a city. Users of the system can rent or check-out these vehicles for a small fee that is generally proportional to trip time or distance. When the user is finished with the vehicle, she can park it in any legal parking spot throughout the city. Included among these parking zones are charging stations, where the vehicle can be plugged in to regain charge. Users who end their trips at charging stations help the system, since as a vehicle charges, it gains the potential to serve a broader class of trips. One potential mechanism to incentivize users to end their rides at charging stations is to offer them a discount on their ride if they do so. In such a scenario, the system operator forgoes immediate revenue to ensure that the vehicle at hand is sufficiently charged for future rides. There is no guarantee, however, that the user will agree to take this discounted ride, since the proposed charging station could be far away from the user's desired end location. In this way, when the user decides whether or not to accept the offered discount, she trades off the price reduction with the potential inconvenience of concluding her ride far away from her desired end location.

In this paper, we study the trade-offs described above by considering how a system operator of a dockless EVSS can optimally offer discounted rides to incentivize users to end

their trips at a charging station. The models that we develop are able to incorporate a variety of discount structures, but we focus our analysis mainly on free-ride policies in which users are strategically given the option to take a free trip to a charging station in lieu of a full-priced ride to their desired destination. This simplified discount structure allows us to develop implementable policies, which we show via extensive simulations perform quite well on a wide variety of performance metrics related to revenue, operational costs and customer satisfaction.

This work builds on an extensive body of research on vehicle sharing systems (VSSs). The increase in VSS-related literature can be attributed to the rapid expansion of such systems and the intriguing operational challenges that accompany this growth. In what follows, we briefly describe the rise of VSSs before detailing the classical operational problems that accompany such systems and that motivate our work on discount rides in EVSSs.

The make-up and growth of VSSs. The first VSSs were comprised entirely of gasoline-powered vehicles, which are still present in many VSSs, i.e. car2go and Zipcar. Soon after, bike sharing systems (BSSs), i.e. CitiBike, were introduced to handle shorter trips, and most recently, cities have witnessed the adoption of EVSSs, i.e. Bird Scooters, which have documented environmental and financial benefits over gasoline-powered vehicles (U.S. Department of Energy, 2016). The gain in popularity of all three VSSs is without question, as membership is slated to exceed 12 million by 2020 and revenue is projected to reach $6.5 billion in 2024 (Navigant Research, 2016b,a). As these systems continue to grow, so too does the necessity to develop efficient solutions to the many daily operational challenges.

Operational problems in VSSs. The primary operational problem in VSSs revolves around balancing supply with demand. In a perfect system, there would always be an available vehicle in close proximity to every inquiring user. In practice, achieving this service level is nearly impossible due to the stochasticity in demand and the limited number of

vehicles. In fact, if left to run entirely on its own, most VSSs would inevitably experience an extremely lopsided dispersion of vehicles because supply and demand rarely match up perfectly. To combat this issue, VSSs have resorted to two main operational levers: rebalancing and pricing. The former refers to manually moving vehicles between stations in anticipation of future demand, which is effective but costly (Fishman, Washington, and Haworth, 2014).

Later generation VSSs became more sophisticated and started to offer dockless parking, which allows riders to park on any street in a pre-defined service region. While dockless systems provide users with more convenience and flexibility in terms of where users are permitted to finish rides, they also bring new flavors of operational problems. For one, merely keeping track of each vehicle's location is a more complex task in free-float systems since the pre-defined parking regions generally span the entire city. This is in stark contrast to traditional docked systems, in which the system's state can simply be described by the number of vehicle at each of the docking stations. This inherent difference significantly complicates the aforementioned rebalancing problem; manual rebalancing in dockless systems is a more tedious task since vehicles are not confined to docking stations. With this in mind, many of the dockless VSSs have flipped the rebalancing problem on its head; instead of manually rebalancing the system themselves, they attempt to incentivize users to accomplish this task for them. For example, LimeBike offers ride credits to users who check-out bikes that have sat idle for an extended period of time.

For the dockless EVSS that we consider, the system's state is characterized by the current location and charge level of each vehicle. In this setting, a balanced system not only has vehicles in close proximity to inquiring users, but it also ensures that these vehicles are adequately charged. In what follows, we describe how EVSS's have attempted to achieve this latter, more elementary notion of a balanced network.

Maintaining a charged fleet in EVSSs. For dockless EVSSs, there is perhaps an even more fundamental issue than that of balancing supply and demand. Paramount to oper-

ational efficiency and the profitability of such systems is the ability to keep the electric vehicle fleet adequately charged so that users do not forgo a ride because they cannot find a vehicle with enough charge. In some existing dockless EVSSs, the current practice is to aggressively fine riders who street-park vehicles with low battery. This approach, however, does not appear to be ideal since it often results in users choosing between the lesser of two evils when they unexpectedly finish a ride with a low-battery vehicle. In such scenarios, users are forced to either navigate out of their way to drop a low-battery vehicle at an open charging station or to park near their desired destination and pay a hefty fine, both of which negatively impact the user experience. Moreover, it is not obvious how exactly to choose this aforementioned "low-battery" threshold, which will have a dramatic effect on the day-to-day dynamics of the system. Choosing this threshold to be too low may result in many vehicles stranded on the street because they do not have enough battery to fulfill any rides. On the other hand, a threshold that is too high may lead to an overwhelming number of fines and hence an unhappy and frustrated customer base. In contrast, the pricing discount incentives that we consider have the potential both to keep the fleet charged while only improving the user experience, since any offered discounted ride can be turned down.

We consider a dockless EVSS consisting of n vehicles and m charging stations. The vehicles in our setting should be thought of as cars or Vespas, and so manually moving these vehicles is quite tedious and costly in relation to moving bikes. At any given time, the state of each vehicle can be described by its location and its current charge level. As time progresses, users rent vehicles and ride them to their desired location. Our goal is to develop and characterize simple conditions under which a dockless EVSS operator should offer free rides to users who end their trip at a charging station. The hope is that these free rides will ensure that the system has sufficiently charged vehicles to serve future demand.

As hinted at above, we seek free-ride policies that are straightforward for the system operator to explain to the user, easy for the user to interpret, and simple to put into action. All three of these characteristics are satisfied by what we call **single-offer range** (SOR)

policies. Under such policies, for each region in the network, there is a single, continuous, battery charge level interval that dictates when a free-ride will be offered. Upon rental, users who select a vehicle whose charge level falls within this interval will be offered a free ride, and those who select a vehicle with a charge level outside of this interval will not be offered a free ride. As we go on to show, these region-specific battery charge level intervals can be computed offline in an efficient manner, and hence implementing these so-called SOR policies is relatively straightforward compared to a nuanced dynamic pricing scheme.

1.1.1 Contributions

We first consider an infinite horizon dynamic program that maximizes the total discounted expected revenue when there are no restrictions on the structure of an optimal policy. The state space of the resulting dynamic program gives the current location and battery level of each vehicle, and the Bellman recursion encodes the trade-off between offering a free ride to a charging station and letting the user take a full-priced ride to her desired end location. This initial formulation has two central issues that hinder its practical use. First, the state space grows exponentially in n and hence the dynamic program is rendered intractable for realistic instances in which the EVSS system contains hundreds of vehicles. Second, even if we could derive an optimal policy from this dynamic program, there is no guarantee that it will be an easily implementable policy, let alone an SOR policy. In fact, it is not clear if it is possible to formulate this dynamic program so that only SOR policies are feasible.

To side-step the first issue, we focus on single-vehicle networks. In this setting, we can easily find the optimal policy, but again, there is no guarantee that this policy will be of the SOR variety. One somewhat counter-intuitive insight that we derive from this simplified one vehicle setting is that the value of a vehicle in a given location does not necessarily increase with its battery level. In Section 1.3, we describe how this observation is directly related to the second issue of deriving SOR policies. We eventually overcome this second difficulty by reformulating the initial infinite horizon dynamic program as a linear program

(LP). By adding binary variables and a set of auxiliary constraints to this LP, we arrive at a mixed-integer program whose optimal solution gives the optimal SOR policy.

We then consider the problem at full-scale, where there are n vehicles and m charging stations in the network. In this setting, we develop free-ride policies based on the approximate dynamic programming technique developed in Desai, Farias, and Moallemi (2012) in which the value functions in the original dynamic programming formulation of the problem are approximated via a linear combination of basis functions. The optimal weights on each basis function within the approximation are generated from the optimal solution to specially crafted LP.

After deriving the optimal SOR policies from the single vehicle formulation of the problem and the approximate multi-vehicle policies that result from our approximate dynamic programming solution, we test their efficacy within a large scale simulation that uses real data from a U.S.-based EVSS. We benchmark the performance of these free-ride policies against our EVSS partner's current practice, in which users who street-park low battery vehicles are fined. Our simulation results reveal that the free-ride policies generate slightly less revenue than the fine-based policy, but provide a significantly better customer experience, which is critical for the long term success of the system.

1.2 Related Literature

We begin by reviewing the previous work on BSSs and VSSs, which both pre-date EVSSs. Then, we summarize the past work on EVSSs, which is limited since these systems have only recently come into existence.

1.2.1 Bike Sharing Systems (BSSs).

Past BSS research has predominantly focused on network design. For instance, Lin and Yang (2011) determine where to build stations to maximize coverage, Sayarshad, Tavassoli,

and Zhao (2012) examine how fleet size affects demand, utilization, and rebalancing costs, and Kabra, Belavina, and Girotra (2020) study the effect of increasing station density on ridership. Freund, Henderson, and Shmoys (2017) develop a procedure to optimally redistribute bicycle docks across stations. All of these papers consider one-way BSSs, where riders can take bikes on one-way trips, which must end at a docking station. Our setting is less restrictive since users can take one-way trips, but they are not forced to end at a docking station.

Rebalancing in one-way BSSs has also been well-researched. This work involves efficiently designing truck routes that minimize the time and cost of moving bikes between docking stations (Raviv, Tzur, and Forma, 2013; O'Mahony and Shmoys, 2015; Schuijbroek, Hampshire, and Van Hoeve, 2017). Pricing has also been studied as a mechanism to rebalance a BSS. Chemla et al. (2013) propose a pricing strategy in which the fare is based on the availability at each station. Others have focused on minimizing underused stations by incentivizing riders to return bikes to these stations (Pfrommer et al., 2014; Singla et al., 2015; Fricker and Gast, 2014).

While the existing BSS research can serve as a starting point for tackling operational problmes in EVSSs, there are two features of our problem that have not yet been considered in the BSS literature. First, to the best of our knowledge, the dockless component has not been studied. Second, the charging element of EVSSs presents a new challenge that does not exist in BSSs, since bikes are human-powered. However, as BSSs grow to include electric-assisted bicycles, maintaining a charged fleet will require attention and we hope that work in this area will draw inspiration from our research.

1.2.2 Car or Vehicle Sharing Systems (VSSs).

While the VSS literature is not as vast, problems related to both system design (Chang et al., 2017; Lu, Chen, and Shen, 2017) and rebalancing (Nair and Miller-Hooks, 2011; Weikl and Bogenberger, 2013) have been studied. Rebalancing in a BSS is inherently simpler than

a VSS, since several bicycles can be placed on a truck and manually redistributed across a city on a single route. The same can obviously not be said for cars, so VSS rebalancing requires additional planning. We note that the existing rebalancing approaches tend to be costly, resource-intense, and time-consuming.

Several researchers have also explored how to rebalance a VSS through pricing discounts. Marecek, Shorten, and Yu (2016) propose a destination-based pricing scheme in dockless VSSs, in which the fare depends on the distance between the drop-off location and the nearest parked vehicle, and Waserhole and Jost (2016) develop a queuing model for setting prices in a one-way VSS to maximize the number of trips. While both papers capture the spirit of the pricing policies that we analyze, neither of these models accounts for a vehicle's remaining battery, which is critical in an EVSS. Banerjee, Freund, and Lykouris (2016) provide a general framework for pricing in any mobility sharing system, but it is not obvious if their approach is able to capture the additional complexity of keeping the fleet sufficiently charged. For an overview of system design and rebalancing in VSSs, see Jorge and Correia (2013).

1.2.3 Electric Vehicle Sharing Systems (EVSSs).

To date, the EVSS literature has primarily focused on system design. Boyacı, Zografos, and Geroliminis (2015) and Brandstätter, Kahr, and Leitner (2017) respectively study where to place charging stations in one-way systems and parking locations in dockless systems.. In the presence of uncertain adoption, He et al. (2017) use robust optimization to define the service area for car2go's dockless EVSS operation in San Diego, CA. Unfortunately, car2go replaced the electric vehicles with gas-powered vehicles and later ceased their San Diego service, confirming that operating a dockless EVSS is challenging (The San Diego Union-Tribune, 2016). For rebalancing an EVSS, Bruglieri, Colorni, and Luè (2014) consider how to dispatch cyclists on folding bikes to low-battery electric vehicles. Upon reaching the vehicle, the cyclist places the collapsed bike in the trunk and drives to a relocation point. In contrast,

we focus on ensuring that the EVSS fleet has enough battery to complete rides by offering a direct pricing discount to customers if they end rides at charging stations.

1.2.4 Related Research.

Lim, Mak, and Rong (2014) examine the behavioral factors behind electric vehicle adoption. Battery swapping, where users can go to stations to exchange depleted batteries for recharged ones, has also been studied (Avci, Girotra, and Netessine, 2014; Mak, Rong, and Shen, 2013). Separately, Bellos, Ferguson, and Toktay (2017) study how VSSs operated by auto manufacturers affects the firm's profit and decision to design more fuel efficient vehicles.

This recent research related to car sharing and electric vehicle usage suggests that both will continue to grow, motivating our goal of effectively managing a system at the intersection of VSSs and EVs.

1.3 Dockless EVSS Models and Free-Ride Policies

We begin by describing our model of the EVSS and then move to detailing our approaches for deriving the free-ride policies discussed above. More specifically, in Section 1.3.1 we describe our model of the EVSS that we consider as well as its underlying dynamics that govern how the system evolves over time. The model that we develop is highly realistic and accounts for battery recharging of idle vehicles at charging stations, uncertainty rooted in demand, manual repositioning movements by the system operator, and the utility trade-off faced by customers who must choose to accept or decline a free ride.

In Sections 1.3.2 and 1.3.3, we analyze a single-vehicle setting and develop a mixed-integer program to find the optimal SOR policies in this setting. Finally, in Section 1.3.4, we summarize our approximate dynamic programming approach for tackling the multi-vehicle problem.

1.3.1 Model Primitives

We partition the service area into r regions indexed by the set $\mathcal{R} = \{1, \ldots, r\}$. Each region $i \in \mathcal{R}$ could represent a street, block or neighborhood depending on the desired granularity of the system. There is a subset $\mathcal{Z} \subset \mathcal{R}$ of regions that house a charging station. The system consists of n vehicles. We assume that a vehicle's state can be fully described by the tuple (i, w) where $i \in \mathcal{R}$ and $w \in \mathcal{W} = \{0, \delta, 2\delta, \ldots, 1\}$ respectively represent the vehicle's current location and battery charge level. We use the convention that $w = 1$ corresponds to a full charge, and $\delta \in [0, 1]$ gives the granularity at which we keep track of battery charge.

We discretize time into disjoint time periods, whose length can be interpreted as the mean time between customer arrivals to the system. In each period, we assume that there is exactly one ride request. We let λ_i be the probability of seeing a vehicle request in region $i \in \mathcal{R}$ during each time period. Given a request for a vehicle in region i, we let p_{ij} be the probability that the user's desired end location is region $j \in \mathcal{R}$. We use $b_{ij} \in \mathcal{W}$ to denote the battery consumption of a trip from region i to region j. Further, we let d_{ij}, f_{ij} and t_{ij} respectively be the distance, fare collected, and duration for rides between regions i and j. A ride can only take place if the requested vehicle has sufficient battery to deliver the user to her desired destination. Thus, we let the set $\mathcal{R}(i, w) = \{j \in \mathcal{R} : w \geq b_{ij}\}$ give all reachable regions of a vehicle whose state is (i, w). Finally, we assume that vehicles located at charging stations gain $\gamma \in \mathcal{W}$ charge in each time period.

Next, we discuss how we incorporate manual vehicle repositioning events by the system operator into our model. Each vehicle is deemed eligible for a manual move to a nearby charging station if its remaining battery is below a predefined battery move threshold b_m. In each time period during which there exists at least one move-eligible vehicle, we assume that a repositioning event is initiated with probability p_m. We model a repositioning event as a "dummy" ride, in which a move-eligible vehicle is uniformly selected to be moved to the closest charging station over a random duration of t_m time periods. The dummy ride

reflects the efforts of a crew member and hence comes at a cost of c_m to the system.

For each user that rents a vehicle, the system operator has the option to offer a free ride to a charging station if the vehicle has enough battery to reach at least one charging station. If a free ride is offered, the user decides whether to accept or reject this free ride based on her realized utility for each of these two options. To formalize this notion, we let $u(d, f)$ be the random utility that a user associates with a ride that leaves her a distance of d from her desired location and whose cost is f. The randomness in the utility arises due to the assumption that there is heterogeneity in each user's sensitivity to price and walking distance. We refer the reader to Section 1.4.1 for the explicit form of the utility function that we use in our simulations. If a free ride is offered and accepted, we assume that the user parks the vehicle at the charging station closest to her desired destination. Finally, we define $\mathbb{P}(\mathit{Accept}_{ijz}) = \mathbb{P}\big[u(d_{zj}, 0) \geq u(0, f_{ij})\big]$ to be the probability that the user accepts a free ride to charging station z. If the user accepts the free ride, she pays nothing and must walk a distance of d_{zj} after dropping off the vehicle at z. On the other hand, if she rejects the offer, she commutes directly to j and pays f_{ij}, which occurs with probability $\mathbb{P}(\mathit{Decline}_{ijz}) = 1 - \mathbb{P}(\mathit{Accept}_{ijz})$. All of the notation introduced above is summarized in Table A.1.

1.3.2 Single Vehicle and Single Charging Station (1V1C)

We begin by studying a setting with a single vehicle and a single charging station, so $n = 1$ and $\mathcal{Z} = \{z\}$. We model the system's dynamics as a discrete time, infinite horizon dynamic program. The state space is given by the tuples $(i, w) \in \mathcal{R} \times \mathcal{W}$, which represent the possible locations and battery levels of the vehicle. The value function $V(i, w)$ gives the maximum total discounted expected revenue that can be derived from a vehicle in state (i, w). The per-period discount factor is $\beta \in (0, 1)$ and we define $\beta_{ij} = \beta^{t_{ij}}$ to be the discount rate for a ride between regions i and j, which takes place over t_{ij} periods.

Recall that in each time period, there is a customer arrival at region i with probability

λ_i, and this request is for a ride to region j with probability p_{ij}. If the vehicle has enough battery to reach the destination, that is $w \geq b_{ij}$, then the inquiring user will rent the vehicle. Otherwise, the user leaves the system and the vehicle remains at state (i, w). Further, if the vehicle has enough battery to reach the charging station z, i.e., $w \geq b_{iz}$, then the system operator must choose whether or not to offer a free ride to z. Finally, we note that if the vehicle's remaining battery satisfies $w \leq b_m$, then the vehicle is manually repositioned to the charging station with probability p_m. In what follows, we present the value functions of our dynamic program for the cases in which the vehicle is not at the charging station (i.e., $i \neq z$) and has enough battery to reach the charging station (i.e. $w \geq b_{iz}$). Thus, the recursion in (1.1) illustrates the cases in which $w \geq \max\{b_{iz}, b_m\}$, and the recursion in (1.2) corresponds to the case in which $b_m \geq w \geq b_{iz}$. The remaining cases are presented in Appendix A.2.

$$
\begin{aligned}
V(i,w) = \max\Bigg\{ & \underbrace{\lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big)}_{DoNotOffer}, \\
& \underbrace{\lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \Big(\mathbb{P}(Decline_{ijz}) \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big) + \mathbb{P}(Accept_{ijz}) \cdot \beta_{iz} V(z, w - b_{iz})\Big)}_{Offer} \Bigg\} \\
& + \Big(1 - \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij}\Big) \cdot \beta V(i,w). \qquad (1.1)
\end{aligned}
$$

$$
\begin{aligned}
V(i,w) = p_m \cdot \underbrace{\big(-c_m + \beta_m V(z,w)\big)}_{MoveOccurs} + (1 - p_m) \cdot \Bigg(\max\Bigg\{ & \underbrace{\lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big)}_{DoNotOffer}, \\
& \underbrace{\lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \Big(\mathbb{P}(Decline_{ijz}) \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big) + \mathbb{P}(Accept_{ijz}) \cdot \beta_{iz} V(z, w - b_{iz})\Big)}_{Offer} \Bigg\} \\
& + \Big(1 - \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij}\Big) \cdot \beta V(i,w) \Bigg). \qquad (1.2)
\end{aligned}
$$

The maximization in (1.1) weighs the trade-off between offering and not offering a free ride,

which is only relevant when the user's desired end location is reachable, i.e., $j \in \mathcal{R}(i, w)$. If the vehicle does not have enough battery to reach region j, then the system stays in the same state, but the value of the vehicle is discounted one period. The *DoNotOffer* term corresponds to the scenario in which the system operator does not offer a free ride. In this case, the system accrues f_{ij} in revenue and the vehicle moves to j, ending this trip in t_{ij} periods (hence the discount factor β_{ij}) with a charge of $w - b_{ij}$. The *Offer* term captures the value of offering a free ride and the recursion considers the probability that this offer will be accepted by the user. If the offer is accepted, the vehicle reaches the charging station z in t_{iz} periods with $w - b_{iz}$ remaining battery. If the offer is declined, then the user rides to her desired destination and pays the full fare.

Equation (1.2) models a scenario in which the vehicle is in state (i, w) and has enough battery to complete short trips, but is still eligible for a manual reposition to the charging station. The structure of (1.2) is similar to (1.1), but the *MoveOccurs* term accounts for the possibility of a manual repositioning event, which occurs with probability p_m. If a manual repositioning event occurs, the system incurs a cost of c_m and the vehicle is moved to the charging station z in t_m periods. If the vehicle is not moved, then the value function takes the form of (1.1). We note that the formulations in (1.1) and (1.2) can be modified to incorporate more flexible discounts, in addition to or in lieu of the free-ride discounts. For instance, offering a $(1 - \sigma)$-discount for some $\sigma \in [0, 1]$ is possible by adding an additional term into the maximization. This term would have the same structure as the *Offer* term, but the system would realize a revenue of σf_{ij} and the utility gained from accepting the $(1 - \sigma)$-discounted ride would be $u(d_{z_j j}, \sigma f_{ij})$. This generalization would allow the system operator to offer a full-fare ride, a $(1 - \sigma)$-discounted ride, or a free ride.

Free-ride policies. For free-ride policy $\pi : \mathcal{R} \times \mathcal{W} \mapsto \{\textit{DoNotOffer}, \ \textit{Offer}\}$, we define $S_\pi = \{(i, w) : \pi(i, w) = \textit{Offer}\}$ to be the set of states in which a free-ride is offered. A free-ride policy π is a single-offer range (SOR) policy if for each region i, there exists battery

charge levels $w_2^i \geq w_1^i \geq b_{iz}$, such that if the vehicle is in state (i, w), then $\pi(i, w) = \textit{Offer}$ if and only if $w \in [w_1^i, w_2^i]$. We let Π and $\Pi_{\text{SOR}} \subset \Pi$ respectively denote the set of all free-ride policies and all SOR policies. Further, let $\pi^* \in \Pi$ be the optimal free-ride policy, which can easily be derived via value function iteration since the dynamic program given in (1.1)-(1.2) has only $r \cdot |\mathcal{W}|$ states, however there is no guarantee that $\pi^* \in \Pi_{\text{SOR}}$.

Next, we present conditions that would guarantee that $\pi^* \in \Pi_{\text{SOR}}$. At first glance, these conditions seem to be trivially satisfied for any reasonable network, however we are able to present simple counter-example to break this intuition. First, note that by re-arranging the *DoNotOffer* and *Offer* terms in (1.1) and (1.2), we see that $\pi^*(i, w) = \textit{Offer}$ if

$$\sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \mathbb{P}(\textit{Accept}_{ijz}) \cdot \left(\beta_{iz} V(z, w - b_{iz}) - \beta_{ij} V(j, w - b_{ij}) - f_{ij}\right) \geq 0, \tag{1.3}$$

and $w \geq b_{iz}$. Hence an SOR policy will be optimal if $\beta_{iz} V(z, w - b_{iz}) \geq f_{ij} + \beta_{ij} V(j, w - b_{ij})$ for a single continuous battery charge level interval. A sufficient pair of conditions for this to hold are (i) the value functions $V(i, w)$ are increasing in the battery level w and (ii) the marginal value of each percentage of charge is larger at charging stations than at standard regions. With respect to (i), it seems intuitive that a vehicle with more charge should be able to generate more revenue than a vehicle with less charge, since vehicles with more charge can serve a broader collection of ride requests. However, we quickly discovered that it is not difficult to construct a system in which more battery is not always beneficial. An example of such a network is provided in Appendix A.3. Consequently, (1.3) can be satisfied for several, disjoint battery ranges and hence π^* is not guaranteed to be an SOR policy. In what follows, we show how to use the above dynamic program to obtain optimal SOR policies in a setting with a single vehicle and multiple charging stations.

1.3.3 Single Vehicle and Multiple Charging Stations (*1VMC*)

In this section, we consider an EVSS with a single vehicle and m charging stations indexed by the set $\mathcal{Z} = \{1, \ldots, m\}$. We assume that if a user whose desired destination is region j accepts a free ride, then she will only park her vehicle at the charging station closest to her destination, which we define as $z_j = \arg\min_{z \in \mathcal{Z}} d_{jz}$. We define $\bar{b}_i = \min_{z \in \mathcal{Z}} b_{iz}$ as the minimum battery level required to reach a charging station from region i, and note that the system operator will only consider offering a free ride if the vehicle's battery level satisfies $w \geq \bar{b}_i$, so at least one charging station can be reached. Since we assume that each user's destination is unknown to the system, it is possible for a free ride to be offered to a user who cannot reach the charging station closest to her desired location. In the case, we assume that the user will reject the free ride.

To find the optimal SOR policy, we first consider the equivalent linear programming of the dynamic program for the 1VMC instance, which is provided in Appendix A.2. For simplicity, in the LP that follows, we only include the constraints for the cases in which $i \notin Z$ and $w \geq b_m$, but note that the analysis holds when the constraints are added for all cases. Let

$$V^{no}(i,w) = \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big) + \Big(1 - \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij}\Big) \cdot \beta V(i,w)$$

and

$$\begin{aligned} V^{o}(i,w) =& \lambda_i \sum_{\substack{j \in \mathcal{R}(i,w): \\ w \geq b_{iz_j}}} p_{ij} \cdot \Big(\mathbb{P}(\mathit{Decline}_{ijz_j}) \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big) \\ &\qquad + \mathbb{P}(\mathit{Accept}_{ijz_j}) \cdot \beta_{iz_j} V(z_j, w - b_{iz_j})\Big) \\ &+ \lambda_i \sum_{\substack{j \in \mathcal{R}(i,w): \\ w < b_{iz_j}}} p_{ij} \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big) + \Big(1 - \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij}\Big) \cdot \beta V(i,w), \end{aligned}$$

where these two expressions respectively correspond to the case in which the system operator does not and does offer a free ride. The linear program of interest in given below in *LP Full*.

$$
\begin{aligned}
Z^* = & \min_{V(\cdot)} \sum_{i \in \mathcal{R}} \sum_{w \in \mathcal{W}} V(i,w) && (LP\ Full)\\
& V(i,w) \geq V^o(i,w) && \forall i \notin \mathcal{Z}, w \geq \max\{b_m, \bar{b}_i\}\\
& V(i,w) \geq V^{no}(i,w) && \forall i \notin \mathcal{Z}, w \geq \max\{b_m, \bar{b}_i\}\\
& V(i,w) = \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big)\\
& \qquad + \Big(1 - \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij}\Big) \cdot \beta V(i,w) && \forall i \notin \mathcal{Z}, b_m \leq w < \bar{b}_i.
\end{aligned}
$$

From the optimal solution to *LP Full*, which we denote as $V^*(i,w)$, we can find the optimal free-ride policy by setting $\pi^*(i,w) = \textit{Offer}$ if $V^*(i,w) = V^{o*}(i,w)$, and hence the set of states where the system operator should offer a free ride is $S_{\pi^*} = \{(i,w) : V^*(i,w) = V^{o*}(i,w)\}$. In fact, for any free-ride policy $\pi \in \Pi$, we can find the total discounted expected revenue from a vehicle in region i with battery level w by solving the following LP

$$
\begin{aligned}
Z(\pi) = & \min_{V(\cdot)} \sum_{i \in \mathcal{R}} \sum_{w \in \mathcal{W}} V(i,w) && (LP\ Policy)\\
& V(i,w) = \mathbf{1}_{(i,w) \in S_\pi} V^o(i,w) + \mathbf{1}_{(i,w) \notin S_\pi} V^{no}(i,w) && \forall i \notin \mathcal{Z}, w \geq \max\{b_m, \bar{b}_i\}\\
& V(i,w) = \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big)\\
& \qquad + \Big(1 - \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij}\Big) \cdot \beta V(i,w) && \forall i \notin \mathcal{Z}, b_m \leq w < \bar{b}_i.
\end{aligned}
$$

Building on the above two LPs, we can find the optimal SOR free-ride policy by solving

the following mixed-integer linear program

$$\tilde{Z} = \max \sum_{i \in \mathcal{R}} \sum_{w \in \mathcal{W}} V(i,w) \qquad (\textit{Single Threshold})$$

$$V(i,w) \leq V^o(i,w) + Mx^{no}(i,w) \qquad \forall i \notin \mathcal{Z}, w \geq \max\{b_m, \bar{b}_i\}$$

$$V(i,w) \leq V^{no}(i,w) + Mx^{o}(i,w) \qquad \forall i \notin \mathcal{Z}, w \geq \max\{b_m, \bar{b}_i\}$$

$$V(i,w) = \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \left(f_{ij} + \beta_{ij} V(j, w - b_{ij})\right) + \left(1 - \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij}\right) \cdot \beta V(i,w) \qquad \forall i \notin \mathcal{Z}, b_m \leq w < \bar{b}_i$$

$$x^o(i,w) + x^{no}(i,w) = 1 \qquad \forall i \notin \mathcal{Z}, w \geq \max\{b_m, \bar{b}_i\}$$

$$x^o(i, w-\delta) \geq x^o(i,w) - y(i,w) \qquad \forall i \notin \mathcal{Z}, w - \delta \geq \bar{b}_i \qquad (1.4)$$

$$\sum_{w \in \mathcal{W}} y(i,w) + x^o(i, \bar{b}_i) \leq 1 \qquad \forall i \notin \mathcal{Z} \qquad (1.5)$$

$$x^o(i,w),\ x^{no}(i,w),\ y(i,w) \in \{0,1\}\ \forall i \notin \mathcal{Z},\ w \geq \max\{b_m, \bar{b}_i\},$$

where M is a large constant. The binary decision variables $x^o(i,w)$ and $x^{no}(i,w)$ respectively denote whether or not a free ride is offered when the vehicle is in state (i,w). The binary variable $y(i,w)$ denotes whether or not battery w is the lower threshold battery level w_1^i for region i. The upper threshold w_2^i is equal to 1 if $x^o(i,1) = 1$, otherwise it is equal to smallest w such that $x^o(i,w) - x^o(i, w+\delta) = 1$. Furthermore, we call attention to two fundamental changes in *Single Threshold* in relation to *LP Full*: (1) the objective is a maximization and (2) the sign of the inequalities in the constraints is reversed. In *Single Threshold*, the "big-M" terms in the first two constraints force the one of the two right-hand sides of these constraints to be quite large, effectively making this constraint irrelevant. The constraint without the large right-hand side is the binding constraint and corresponds to the optimal action that maximizes the expected discounted reward.

In the optimal solution to *Single Threshold*, we refer to the optimal, binary decision variables as $\tilde{x}^o(i,w)$, $\tilde{x}^{no}(i,w)$ and $\tilde{y}(i,w)$. Based on this optimal solution, we define the free-ride policy $\tilde{\pi}(i,w) = \textit{Offer}$ if $\tilde{x}^o(i,w) = 1$, and therefore $S_{\tilde{\pi}} = \{(i,w) : \tilde{x}^o(i,w) = 1\}$. In the following proposition, we show that $\tilde{\pi} \in \Pi_{\text{SOR}}$. All proofs are provided in Appendix A.4.

Proposition 1.3.1. *Let $\tilde{\pi}$ be the free-ride policy derived from Single Threshold. We have that $\tilde{\pi} \in \Pi_{\text{SOR}}$.*

Next, we build on Proposition 1.3.1 and show that $\tilde{\pi}$ is actually the optimal SOR policy.

Theorem 1.3.1. *The policy $\tilde{\pi}$ that is derived from the optimal solution to Single Threshold is the optimal SOR free-ride policy. In other words, $\tilde{\pi} = \arg\max_{\pi \in \Pi_{\text{SOR}}} Z(\pi)$.*

Theorem 1.3.1 shows that we can recover the optimal SOR policy by solving *Single Threshold* and constructing $\tilde{\pi}$.

1.3.4 Multiple Vehicle and Multiple Charging Stations (NVMC)

For the multiple vehicle setting, our dynamic program must keep track of the location and battery level of every vehicle in the network. Further, since each ride can last multiple time periods, we also must account for vehicles that are currently in use. More formally, the state of each vehicle can be represented by the 4-tuple (i, w, τ, j), where $i \in \mathcal{R}$ gives the vehicle's current (or last) location, $w \in \mathcal{W}$ gives the vehicle's battery level when it was at region i, and τ gives the number of time periods until the vehicle reaches the desired destination j. We assume access to each user's end destination j only after deciding whether or not to offer them a free ride. If the vehicle is idle, we set $\tau = 0$ and $j = 0$. Finally, we let $\mathcal{S}$ be all possible 4-tuples for the n vehicles of the system, and for state $s \in \mathcal{S}$, we let $V(s)$ be the optimal value functions. We do not give the explicit form of these value functions since they are simply more cluttered versions of those presented for the 1VMC problem instance. Naturally, computing the optimal policy in this setting, much less characterizing

its structure, is quite a difficult task. As such, we elect to employ the approximate dynamic programming approach of Desai, Farias, and Moallemi (2012). In what follows, we summarize this approach and describe how we apply it to our setting.

We develop and test free-ride policies by solving the smoothed approximate linear program (SALP) introduced by Desai, Farias, and Moallemi (2012). In this approach, the dynamic program is formulated as an equivalent linear program and the value functions are approximated by a linear combination of L basis functions, which capture key properties of the state of the system. More specifically, we approximate the value functions $V(s)$ by $\tilde{V}(\mathbf{s}) = \sum_{l=1}^{L} r_l \cdot \phi_l(\mathbf{s})$, where $\phi_l : \mathcal{S} \mapsto \mathbb{R}$ is the l-th basis function and r_l is its weight. These weights are the decision variables within the linear program. Once the optimal weights have been derived, free-ride policies can be developed by using the value function approximations $\tilde{V}(s)$ to approximate the revenue trade-off between offering a free ride or letting the user take her desired ride.

This approximation reduces the number of variables in the LP formulation of the dynamic program (we have just L decision variables, one for each basis function), but there is still a constraint for every state-action pair, and hence the resulting linear program can be intractable when the state space is large, as is the case in our problem. In such scenarios, Desai, Farias, and Moallemi (2012) propose an approach in which the constraints are randomly sampled and then the linear program is solved using this subset of states. Further, the optimal solution to this linear program is permitted to violate the constraints up to a certain "budget", whose magnitude reflects the extent to which the sampled linear program is further approximated. We explain how we sample constraints and choose the budget in Appendix A.8. Surprisingly, Desai, Farias, and Moallemi (2012) show both theoretically and numerically that the number of constraints that one must sample to arrive at reasonable[1] approximation of the original linear program that contains all possible constraints does not

[1]The exact theoretical guarantee is fairly technical and hence we leave our description of this guarantee at this high level.

depend on the size of the underlying state space $\mathcal{S}$, but only on L, the number of basis functions.

1.4 Numerical Experiments

In this section, we present the details and results of a series of three distinct large-scale discrete event simulations, which we carry out on EVSS networks inspired by those of our collaborator. We crafted these experiments in an effort to study the efficacy of free-ride policies under varying demand patterns and system parameters. We benchmark the performance of the free-ride policies that we develop against our EVSS collaborator's current fine-based practice, under which users are fined for street-parking low-battery vehicles. We present the details of our experiments in the following three sections. In Section 1.4.1, we begin by providing a high-level description and motivation for each of the three experiments that we conduct. Following this high-level summary of each experiment, we describe the key ingredients and assumptions that go into setting up and running each experiment. Next, in Section 1.4.2, we discuss the various policies that we test, and list the performance metrics that guide our assessment of each policy's performance. Finally, in Section 1.4.3, we summarize the results of the three experiments, and in doing so, we provide high-level managerial insights that surface from our extensive series of simulations.

1.4.1 The Three Experiments: Motivation and Set-Up

We begin by summarizing the distinguishing features and high-level goals of our three experiments. This summary is followed by a description of how we design and set up each individual experiment to match these intentions. As part of this latter description, we explain how we use the historical ride data provided to us by our EVSS collaborator, which includes transactional data based on all rides from 9/20/15 to 11/21/15. Each vehicle in the fleet is equipped with a device that continually transmits information every minute to

the company's database. As a result, for each ride, we know the starting and ending timestamps, origin and destination coordinates, fare paid, battery consumed, distance traveled, and a variety of other readings. As suggested by the company, we removed rides greater than 15 miles, since it is likely that these rides do not reflect a commute, but rather a leisure trip without a pre-conceived destination. Further, we removed rides that end in regions that do not serve as an origin for any other rides, since these rides will create "sink" regions in our simulation. The data cleansing ultimately leaves us with ∼28,300 rides. About 85% of these rides are less than an hour, and the average distance traveled is 3.6 miles. It is important to note that over the two month period that our data set spans, the system was static and had no change in fleet, service area, or pricing. As we discuss later on, in each of the three experiments, we use this data set to varying degrees to guide our choices for the key system parameters within our simulations.

The three experiments that we conducted are respectively labeled True_EVSS, Parameter_Sensitivity, and Demand_Sensitivity and are summarized below. We note that we refer to the arrival probabilities λ_i and the transition probabilities p_{ij} as demand parameters, while any other system parameter is referred to as an operational parameter.

- Experiment True_EVSS: The intent of this first set of experiments is to measure the performance of each policy that we consider using a simulated EVSS network that most closely resembles that of our collaborator. For this purpose, we use the historical ride data to guide many of the underlying parameter values within our discrete event simulation. We carry out this simulation on only a single set of parameter values, which we refer to as the baseline parameters. We believe these values to be the most realistic based on the historical ride data and discussions with our EVSS collaborator.

- Experiment Parameter_Sensitivity: In this experiment, we fix demand to be uniform across the network, and then we study the overall impact of changing key operational parameters on both the performance of various policies, and on the specific structure

of the optimal SOR policies. We delay a detailed description of the exact nature of this parameter sensitivity analysis until after fully formalizing the set-up of our simulations.

- Experiment Demand_Sensitivity: In our final experiment, we fix all of the operational parameters that were varied in the previous experiment to the baseline values, and instead only vary the two parameters related to demand. Our goal in this setting is to understand how the performance and structure of the optimal SOR policies is affected by shifting the demand towards or away from the charging stations in the network. Again, we delay a formal description of how we vary demand in this way until after fully formalizing the set-up of our simulations.

Next, we move to describing how we set up each of the three experiments. As should be evident from the summaries above, the three experiments only differ by the manner in which the demand and operational parameters are set and varied across simulations. As such, the experimental set-ups for each of experiment share quite a bit of common ground. In an effort to succinctly describe the set-up for each experiment, we first describe their shared features, before sequentially addressing the defining elements that make each of the experiments unique.

Common features of each experiment.

In what follows, we detail the elements of our simulation that remain fixed as we move from experiment to experiment. First, among these common elements is the structure of the underlying EVSS network, which includes how we partition the service area into discrete regions, the number of vehicles in the system, and the number/location of the charging stations. Furthermore, throughout each of our experiments, we use the same random ride generator to determine each user's exact trip when they rent a vehicle, and we also use the same random utility model to capture customer preferences when a free ride is offered. Finally, the random process governing manual respositions by the system operator remains unchanged across the

experiments. Next, we discuss each of these common aspects individually in greater detail.

EVSS network. The service area that we use within the simulations is modeled off of the location of our EVSS collaborator. We create a discretized service region by partitioning our EVSS collaborator's service area into a collection of square regions that comprise $\mathcal{R}$, the set of feasible locations for vehicles and charging stations within our models and simulations. The size of each square was chosen so that it takes no more than four minutes to walk from end-to-end based on the assumption that people walk at 3.1 miles per hour (Browning et al., 2006). This partitioning scheme leaves us with 577 square regions, where the distance between opposite corners is approximately 0.2 miles. Our EVSS collaborator has charging stations positioned in 40 of these regions, and manages a total of 300 electric vehicles.

Generating trips. To generate the inter-arrival times of users, we fit the historical inter-arrival data to several distributions and find that the best fit is the inverted beta distribution with shape parameters $\alpha = 0.92$ and $\beta = 4.07$, location parameter $= 0.01$, and scale parameter $= 8.86$. This parameter set gives a mean inter-arrival time of 2.66 minutes, which we use as our period length when deriving the discount policies. If a customer arrives to a region with several vehicles available, we assume that she always rents the highest-charged vehicle [2]. In the event that a vehicle is rented in region $i \in \mathcal{R}$ and the user's desired location is $j \in \mathcal{R}$, we use a linear regression model fitted to the historical ride data to randomly generate the ride duration (in minutes) t_{ij} and battery consumption b_{ij} of the given user's trip. We refer the reader to Appendix A.5.1 for the full regression output and a detailed description of how the regressions are used to generate each ride in the simulation. The fare for the given trip is then set to be $f_{ij} = 1 + 0.15 \cdot t_{ij}$, which is a pricing scheme vetted by our collaborator and common in many modern dockless EVSSs.

[2] Our historical data set does reveal that users generally choose the highest charged vehicle when presented with a choice among many vehicles.

Customer utility model. If the system operator offers a free ride, we assume that the user trades off the inconvenience of walking a distance of d to her desired destination and paying a fare of f for her ride. We capture this trade-off using a common linear structure for the utility function throughout all of our experiments. More specifically, we set $u(d, f) = -\alpha_d \cdot d - \alpha_f \cdot f$, where $\alpha_d \sim U(0, \mathcal{DM}),\ \alpha_f \sim U(0, 1)$ are generated randomly for each arriving user. The operational parameter $\mathcal{DM}$ captures the extent to which there is variability within the customer population with regards to the inconvenience of walking the extra leg to the desired destination of the user. Note that we only consider the disutility associated with each ride, however this utility function could easily be updated by simply adding a constant term μ, which reflects the utility gained from a successful commute. We discard this positive term simply for notational convenience.

Manual repositioning events. Finally, recall that a vehicle is deemed move-eligible if it has a battery level below b_m. Furthermore, in each time period, if there are any move-eligible vehicles in the network, then with probability p_m we uniformly select one to be moved to a charging station. The selected vehicle is then assumed to be unavailable for a random amount of time, generated from a truncated normal distribution with a mean of four hours and standard deviation of 30 minutes. Given that such moves could require an employee to go between opposite ends of the city, we felt this was a reasonable distribution for the time required for a manual move.

Battery recharging. When solving for the policies that we test, we discretize the set of battery levels $\mathcal{W}$ by $\delta = 0.02$. Further, we set the re-charging rate to be $\gamma = 0.02$, which corresponds to a charging time of 2.25 hours to replenish a completely depleted vehicle. While our collaborator's vehicles take 3-5 hours to fully recharge, we were not able to discretize the battery levels in any finer increments than 0.02 and still tractably solve *Single Threshold* to optimality in a reasonable amount of time. In the simulation however, we take a more conservative approach and assume that the battery takes 5 hours to replenish.

Distinguishing features for experiment True_EVSS.

As noted above, the intention of this experiment is to recreate the EVSS network of our collaborator with as much accuracy and nuance as possible. For this purpose, we use the historical ride data to govern the demand parameters. More formally, for each region i, we set $\lambda_i = \frac{\text{\# of rides originating at } i}{\text{\# of total rides}}$, and for each pair of regions $i, j \in \mathcal{R}$, we set $p_{ij} = \frac{\text{\# of rides from } i \text{ to } j}{\text{\# of rides originating at } i}$. In total, the data set of ∼28,300 rides is spread across 16,720 unique origin-destination pairs out of a possible $r^2 = 332,929$ such pairs. The remaining set of operational parameters are fixed to the following values, which we denote as our *baseline parameter set*. We set the cost c_m of a manual move to be \$25 since this is the fine our collaborator enacts for street-parking a low-battery vehicle that eventually must be moved to a charging station. We set the battery move-threshold $b_m = 0.2$, the probability of a move $p_m = 20\%$, and the dollar-to-mile sensitivity parameter in the customer utility function $\mathcal{DM} = 5$. These values are our best guesses at reality after several discussions with our EVSS collaborator.

Distinguishing features for experiment Parameter_Sensitivity.

In this experiment, we assume uniform demand and transition probabilities across the entire network. As a result, rides between any pair of regions can occur, hence we have r^2 possible trips and we set $\lambda_i = p_{ij} = \frac{1}{r}$. The motivation behind this assumption is rooted in the historical arrival probabilities, which are reasonably uniform across each region as indicated in the heatmap of λ_i in Appendix A.9. For this fixed uniform demand pattern, we vary $p_m = \{5\%, 10\%, \ \mathbf{20}\%\}$ and use a battery move thresholds of $b_m \in \{0.05, \ 0.10, \ \mathbf{0.20}\}$. We also consider manual move costs of $c_m \in \{\$5, \ \mathbf{\$25}, \ \$50\}$ and dollar-to-mile sensitivity parameter $\mathcal{DM} \in \{0.5, \mathbf{5}, 20\}$. The baseline values introduced in our description above of experiment True_EVSS are bolded for reference.

Distinguishing features for experiment Demand_Sensitivity.

In this final experiment, we fix the operational parameters at their baseline values and we vary the two demand parameters to capture scenarios in which demand is either generally close or far away from the charging stations. In total, we test nine different ride patterns that are generated by varying arrival and destination demand to reflect the following three scenarios: either clustered close (C) to charging stations, uniform (U) across the service area, or far (F) from charging stations.

When demand is assumed to be clustered close to (far away from) charging stations, we assume that the arrival probability at region i and the probability that a user ends her trip at region i is linearly decreasing (increasing) in d_{iz_i}, the distance between region i and its closest charging station. Specifically, when demand is close to charging stations, we set $\lambda_i = \frac{d^* - d_{iz_i}}{\sum_{j \in \mathcal{R}} d^* - d_{jz_j}}$, where we define $d^* = 0.1 + \max_{j \in \mathcal{R}}\{d_{jz_j}\}$. Note that regions that are closer to charging stations will have larger arrival probabilities. When demand is assumed to be farther from charging stations, we set $\lambda_i = \frac{0.1 + d_{iz_i}}{\sum_{j \in \mathcal{R}} 0.1 + d_{jz_j}}$. In both cases, we use the additive factor of 0.1 to ensure that the relative magnitudes of the arrival probabilities are reasonable and non-zero. We set the transition probabilities for each of the three scenarios in a similar fashion.

1.4.2 Policies Tested and Performance Metrics

In this section, we summarize the various free-ride and benchmark policies that we implement within the three experimental settings that we consider. We begin by describing the handful of free-ride-based policies that are derived from the models presented in Section 1.3. These policies are all computed using a discount factor of $\beta = 0.999$ and discretized battery levels in increments of $\delta = 0.02$. Next, we describe two benchmark policies; one of which is the current fine-based practice of our EVSS collaborator and the other is a "hands-off" policy in which the system operator lets the system unfold naturally. We then detail the numerous

performance metrics that we use to measure the efficacy of each policy. The union of all policies that we consider across the three experiments are summarized below:

1VMC-SOR: This is the optimal SOR free-ride policy $\tilde{\pi}$ that is derived by solving *Single Threshold*. Recall that under the policy $\tilde{\pi}$, we offer a free ride each time that a user rents a vehicle in region i with battery level w and $(i, w) \in S_{\tilde{\pi}}$.

1VMC-50: This is the discount-ride policy that we derive from solving the *1VMC* dynamic program modified so that in addition to having the option to offer a free-ride, the system operator can also offer a half-priced, or 50%-discounted, ride.

NVMC-SALP: This is the policy derived from solving the smoothed-ALP described in Section 1.3.4. Due to the fact that this policy is computationally intensive to implement, we only consider its performance in experiment True_EVSS. We seed our value function approximation using ten basis functions, which we list in Appendix A.8 along with other key details for implementing this policy.

Fine-Based (FB): Under the Fine-Based policy, users will be fined \$25 for street-parking a vehicle that has a charge level less than b_m, the battery threshold for a manual move. The user can avoid this fine by parking the low-battery vehicle at a charging station. This is the current policy implemented by our EVSS collaborator to alleviate "stranded" low-battery vehicles in their network. Hence the interesting scenario within the simulation arises when the user's preferred trip does not end at a charging station and leaves the rented vehicle depleted. In this case, the user will trade off the utility $u(0, \$25 + f_{ij})$ of incurring a fine, but getting to her desired location, with the utility $u(d_{z_j j}, f_{ij})$ of dropping the vehicle at a charging station. Ultimately the user will select the higher utility option.

Never-Offer (NO): Under this policy, the system operator lets the system unfold naturally and so the only way for a vehicle's battery to be replenished is if the customer's intended

destination is a charging station or if the vehicle is selected for a manual repositioning to a charging station.

We evaluate the performance of these policies via Monte Carlo simulation. Each simulation begins with fully-charged vehicles assigned to regions according to the distribution of arrival probabilities, and then runs for 100 days. Since the network we consider is fairly large, we use the first 70 days in each simulation as a warm-up period. Using rides from the last 30 days, we compute a wide variety of performance metrics, which are all listed in Table 1.1. The values that we eventually report in our results are per-day averages of each metric over 30 distinct simulations of the 100 day time horizon. We note that we do not report profit in Table 1.1, but we do track revenue generated from fares and the number of manual repositioning moves, which is a proxy for operational costs. Both of these metrics can be used together to develop a sense of profit. Additionally, for experiments Parameter_Sensitivity and Demand_Sensitivity, we summarize the structure of the SOR policies that we derive from solving *Single Threshold* by reporting the average battery charge-level range that characterizes these threshold-based policies.

1.4.3 Results and Managerial Insights

In this section, we sequentially present the results of each experiment. In doing so, we concisely summarize the core trade-off between revenue earned and customer satisfaction that arises when designing incentive-driven (or penalty-driven) policies in dockless EVSS networks. Furthermore, we highlight the high-level managerial insights that we are able to glean from our simulations, which we believe to be impactful take-home points.

1.4.3.1 Results of Experiment True_EVSS.

The performance of all five policies listed above is presented in Table 1.2. What is immediately evident is that the Fine-Based policy garners more revenue that the discount-based

Table 1.1: Description of Performance Metrics.

Metric	Description
Revenue	Daily revenue accrued by the system.
Rides Completed	Rides taken per day.
Moves	Number of manual repositioning moves completed per day.
Offers	Number of discounted-ride offers extended per day. For the Fine-Based policy, this is the number of times per day that a customer decides between accepting or avoiding a fine.
Accepts	Number of discounted-ride offers accepted per day. For the Fine-Based policy, this is the number of times per day that a customer avoids a fine by ending their trip at a charging station.
Unmet Demand | Vehicle	Total unmet demand per day due to users not finding a vehicle at their origin region or at one of the neighboring regions.
Unmet Demand | Battery	Total unmet demand per day due to vehicles not having enough charge.
Utility per Ride	Average utility of users who were offered the discounted ride option. For the Fine-Based policy, this metric is computed only using rides where the user had to trade-off a fine and the inconvenience of dropping the vehicle at a charging station.
Average Battery	Average charge of the fleet (with and without the vehicles at charging stations) at the end of the day.
Proportion in $\mathcal{Z}$	Proportion of the fleet at a charging station (in $\mathcal{Z}$) at the end of the day.
Rides Fulfilled at i	Proportion of rides fulfilled at a customer's arriving region versus at one of the neighboring regions.

policies, but users of the system under the Fine-Based policy experience at least double the disutility of users in the discount-based policies. More specifically, we observe that the revenue earned under the Fine-Based policy is 2% and 10% higher than the revenue earned under the NVMC-SALP and 1VMC-SOR policies respectively. This trend is likely a result of the stringent $25 fine that is enough to prevent users from leaving uncharged vehicles on the street, without having to discount their ride. Consequently, users will often forgo their desired ride in order to avoid a fine, and instead park the vehicle at a (potentially undesirable) charging station. This leads to far fewer manual moves per day, but an average utility that is ten times worse than the 1VMC-SOR policy. Hence the notion that the Fine-Based policy generates the most revenue should be taken with a grain of salt, as our simulation does not account for the long-term consequences of a dissatisfied user-base. In short, the potential negative impact of the Fine-Based policy on customer satisfaction could outweigh the short-term benefits of increased revenue.

1.4.3.2 Results of Experiment Parameter_Sensitivity.

Recall that in this experiment, we depart from the historical demand patterns and set the arrival and transition probabilities to be uniform across the entire network. For this fixed demand pattern, we consider eight configurations of the operational parameters, where each configuration is distinguished by a deviation from the baseline setting along one parameter. More specifically, we test $c_m \in \{\$5, \mathbf{\$25}, \$50\}, b_m \in \{0.05, 0.10, \mathbf{0.20}\}, p_m \in \{5\%, 10\%, \mathbf{20\%}\}$, and $\mathcal{DM} \in \{0.5, \mathbf{5}, 20\}$, where the bolded values indicate the baseline values. Again, we arrive at eight different parameter configurations by choosing one parameter whose value will deviate from the baseline, and then enumerating over all such combinations. For this second experiment, we only test the 1VMC-SOR and the Fined-Based policies, whose performance along all of the dimension listed in Table 1.1 is reported in Appendix A.6. While our primary focus of this experiment is to conduct a performance sensitivity analysis with regards to the many operational parameters, we first comment on

the relative overall performance of the two tested policies. In general, we observe that the relative performance of these two policies matches that of experiment True_EVSS, however interestingly, when we consider the test case where the battery move threshold is set to its lowest value ($b_m = 0.05$), the FB policy is dominated by the 1VMC-SOR policy along all performance metrics. This is likely a result of the fact that when b_m is low, users can street-park low-battery vehicles that cannot serve many future rides without incurring a fine.

With regards to the sensitivity analysis, Table 1.3 summarizes the impact of varying each operational parameter on the performance of 1VMC-SOR policies. More specifically, this table presents the percentage change in each performance metric as the operational parameters c_m and b_m are changed from their baseline values. This table does does include the sensitivity analysis for $\mathcal{DM}$ and p_m since changing these parameters had only a mild impact on the many performance metrics that we consider. Of particular interest is how the repositioning cost c_m affects the number of manual repositioning events, which is reported in column four of Table 1.3. When the cost c_m decreases, the system realizes 16% more revenue, but the caveat is that the number of manual movements jumps by 238% because manual moves are cheaper. As expected, the system relies on manual repositioning much more to bring vehicles back to charging stations when c_m is low. When the cost c_m is higher, we see the opposite effect: accrued revenues drop by 9% and the system utilizes free rides instead of high-cost more manual movements to keep to prevent stranded low-battery vehicles. Interestingly, we also see dramatic swings in the levels of both types of unmet demand as the battery-move threshold b_m is changed. Naturally, as b_m is decreased from its baseline value of 0.2, we see more unmet demand due to insufficient battery levels, since low-battery vehicles will not always prompt a manual move. However, increases of over 100% and 800% are surprising and reflect the potential impact of the parameter b_m. These large percentage increases can be explained by noting that when b_m is set to its lowest value of 0.05, a vehicle with a battery level of 0.06, for example, will not be manually moved to a charging station and will result in lots of unmet demand due to the fact that this vehicle

cannot serve many rides.

In addition to monitoring changes in performance metrics, we also study how the structure of the SOR policy changes as we vary key operational parameters. The first column of Table 1.4 reports the average length of the offer range across all regions of the SOR policy under each parameter configuration that we consider. The remaining three columns report the correlation of the length of the offer range with several region-specific features, such as the arrival rate λ_i and the expected battery consumption from i when the vehicle is fully charged (i.e., $w = 1$). Note that if the correlation is negative, the average SOR length decreases with the corresponding feature. This analysis helps illuminate additional drivers behind the frequency with which free-rides are offered. For instance, the last column indicates that offer ranges are larger at regions where users take rides that consume more battery, since, in this case, vehicles are more likely to end trips with low battery.

Perhaps the most striking insight from Table 1.4 is how the cost of manual moves c_m and customers' willingness to walk $\mathcal{DM}$ impact the length of the SOR. Our results indicate that the mean SOR length approximately doubles both when c_m increases from \$5 to \$50, and when the variability in willingness to walk increases from $\mathcal{DM} = 0.5$ to $\mathcal{DM} = 20$. This latter change corresponds to a decrease from 97% to 49% in the probability of accepting a free-ride offer, and hence we observe larger free-ride offer ranges as the system operator faces more uncertainty surrounding each customer's response to a free-ride offer.

Results of experiment Demand_Sensitivty.

In this experiment, we fix the operational parameters at their baseline values and we study the impact of varying demand patterns on the performance of the Fine-Based and 1VMC-SOR policies. For the 1VMC-SOR policy, we also study how the structure of the SOR changes as we shift demand. To accomplish this task, we create nine different demand settings which are characterized by the generall proximity of the underlying demand to the scattered charging stations. The full set of results are presented in Table 1.5, where

the performance of the 1VMC-SOR policy is reported relative to that of the Fine-Based policy on six different performance metrics (the top six charts) for each of the nine demand scenarios. The full set of results are available in Appendix A.7.

As is the case in the previous two experiments, the revenue generated under the Fine-Based policy is greater than that generated under the 1VMC-SOR policy over each demand scenario that we consider. However again, this realization should be taken with a grain of salt, since this lower revenue is driven by the fact that the 1VMC-SOR policy forgoes revenue in exchange for the opportunity to keep a charged fleet, and not because it fulfills fewer rides per day. As a result, the 1VMC-SOR policy is indeed able to preserve a higher charged fleet, which in turn results in less unmet demand as users are able to access sufficiently charge vehicles that allow them to take their desired rides. And like we see in the previous experiments, the free-ride policy provides a better customer experience with an average disutility that is 13-16% of the disutility under the Fine-Based policy.

We note that the metric most strongly affected by the shifting demand is the number of moves per day. As users' intended destinations shift to being farther from charging stations, the number of manual moves decreases. This occurs because free rides are offered more liberally in this setting in an attempt to lure vehicles back towards charging stations.

The bottom three charts in Table 1.5 show how the structure of the SOR changes as demand is varied. Generally, we find that if users are already planning on ending their trip near a charging station, then there is less of a need to offer discounts to charging stations. When the demand pattern flips in the opposite direction and users generally begin their rides near charging stations and end far away from charging stations, then the SORs are selected to be quite large in an effort to keep traffic in close proximity to where demand is expected to arise. The results in the bottom right of Table 1.5 shows that d_{iz_i}, the distance between each region and its closest charging station, can be useful in characterizing SOR length when destinations are far away from charging stations. In the left-most column (i.e., $p_{ij} = \text{C}$), the correlation between distance and offer range is fairly non-existent, but as the

destinations move farther away from charging stations, the correlation becomes larger. One explanation for this phenomena is that when regions near charging stations are not common destinations, free rides need to be offered liberally to maintain a charged fleet.

1.5 Conclusion and Directions for Future Work

In this paper, we study the use of discounted rides as a mechanism to directly incentivize users to park vehicles at charging stations in order to keep the fleet of vehicles adequately charged. We focus on developing simple free-ride offer policies, which we refer to as single-offer range policies. These policies specify a critical battery levels for each region i, which serve as cut-off points for whether or a not a free ride should be offered. Not only do we provide a formulation to find the optimal single-offer range policy for certain settings, but we also demonstrate that such policies can be quite effective in terms of their ability to generate revenue, keep the fleet of vehicles charged and keep the user-base happy. We show this latter point through an extensive discrete event simulation that is seeded with historical ride data from a real EVSS. While offering price incentives to users to change their destination to charging stations seems like a good idea, our results show that Fine-Based policies are also effective, and the advantages between the two types of policies depend on the system's objectives and features of the network and user base.

There are many interesting directions for future work with regards to dockless EVSS systems. Since most mobility sharing systems experience travel demands varying by hours over the day, one potential extension of our work involves incorporating time-varying ride patterns. We attempted to extend our infinite horizon framework to a time-varying setting where we solved for an time-specific SOR policy for several time windows throughout the day. Depending on the time block of an arriving customer, the corresponding time-specific policy is used to determine if a free ride should be offered. Unfortunately, this time-block policy did not perform as well as the free-ride policy generated from using the ride data from the entire

Table 1.2: Experiment True_EVSS: Performance of Policies on Historical Data.

Metric	1VMC-SOR	1VMC-50	NVMC-SALP	NO	FB
Revenue	\$2,174.39	\$2,184.89	\$2,336.79	\$2,391.83	\$2,386.87
Rides Completed	348.84	344.44	347.31	350.16	349.61
Moves	8.31	5.39	24.24	24.50	2.26
Offers	35.79	62.01	5.62	–	38.40
Accepts	32.05	49.96	5.06	–	36.11
Unmet Demand | Vehicle	79.65	84.97	79.82	77.16	77.50
Unmet Demand | Battery	3.33	2.42	4.70	4.50	4.71
Utility per Ride	-0.47	-2.06	-0.47	–	-4.75
Average Battery (All Vehicles)	71.2%	75.5%	68.0%	66.9%	68.3%
Average Battery (w.o. Vehicles at $\mathcal{Z}$)	55.4%	58.9%	51.8%	51.9%	53.1%
Proportion at $\mathcal{Z}$	41.1%	46.3%	39.0%	36.6%	38.1%
Rides Fulfilled at i	55.9%	55.5%	55.7%	55.9%	56.0%

Note: Simulation based on historical ride demand and baseline parameters. Vehicles start at 50% battery. Metrics are averaged over 15 demand sample paths.

Table 1.3: Experiment Parameter_Sensitivity: Impact of Network Parameters on Performance Metrics.

Parameter Change	Revenue	Rides Completed	Moves	Offers	Unmet Demand Vehicle	Unmet Demand Battery	Average Battery	Proportion in $\mathcal{Z}$
Baseline	\$1,185	231.5	4.84	63.70	196.83	4.31	0.88	0.78
$c_m = \$5$	\$1370	238.5	18.53	40.21	184.78	9.32	0.84	0.74
	(16%)	(3%)	(283%)	(-37%)	(-6%)	(116%)	(-4%)	(-5%)
$c_m = \$50$	\$1074	217.5	3.09	65.82	212.49	3.14	0.90	0.81
	(-9%)	(-6%)	(-36%)	(3%)	(8%)	(-27%)	(2%	(4%)
$b_m = 0.05$	\$1262	248.9	2.99	68.72	141.78	38.98	0.79	0.67
	–	–	–	–	(-28%)	(804%)	(-10%)	(-14%)
$b_m = 0.10$	\$1240	244.0	3.00	68.26	175.27	12.11	0.85	0.73
	–	–	–	–	(-11%)	(181%)	(-3%)	(-5%)

Note: Performance metrics computed as the average over 30 demand sample paths. For each configuration, we use the performance of the *1VMC-SOR* policy to report the percentage change in each metric relative to the baseline. Only if the difference in performance is significant at 5%, then we report the relative percent change below the metric.

Table 1.4: Experiment Parameter_Sensitivity: Sensitivity of SOR Size for Various Network Configurations.

Instance	Parameter Changes	Mean Length	Correlation between SOR Length and: Arrival Probability	Distance to Closest Charging Station	Expected Battery Consumption with $w = 1$
1	Baseline	37.97	-0.29	-0.27	0.62
2	$c_m = \$5$	22.32	-0.26	-0.26	0.49
3	$c_m = \$50$	42.59	-0.26	-0.26	0.53
4	$b_m = 0.05$	33.53	-0.32	-0.18	0.70
5	$b_m = 0.10$	33.53	-0.32	-0.18	0.70
6	$p_m = 5\%$	37.50	-0.29	-0.21	0.66
7	$p_m = 10\%$	37.79	-0.29	-0.26	0.64
8	$\mathcal{DM} = 0.5$	26.30	-0.07	-0.43	0.14
9	$\mathcal{DM} = 20$	53.40	-0.34	-0.09	0.80

Note: The mean length of the SOR policy is computed as the average length over all the regions in the network

Table 1.5: Experiment Demand_Sensitivity: Impact of Varying Ride Demand (λ_i and p_{ij}).

Revenue
Relative to FB

		p_{ij}		
		C	U	F
λ_i	C	74%	73%	70%
	U	76%	74%	69%
	F	84%	82%	75%

Moves
Relative to FB

		p_{ij}		
		C	U	F
λ_i	C	98%	83%	58%
	U	104%	85%	62%
	F	243%	182%	90%

Unmet Demand | Vehicle
Relative to FB

		p_{ij}		
		C	U	F
λ_i	C	110%	110%	116%
	U	106%	107%	114%
	F	101%	101%	102%

Unmet Demand | Battery
Relative to FB

		p_{ij}		
		C	U	F
λ_i	C	39%	33%	25%
	U	40%	36%	26%
	F	69%	58%	37%

Average Disutility per Offer
Relative to FB

		p_{ij}		
		C	U	F
λ_i	C	13.5%	13.5%	13.6%
	U	13.5%	13.4%	13.7%
	F	13.5%	13.5%	16.3%

Rides Fulfilled
Relative to FB

		p_{ij}		
		C	U	F
λ_i	C	97%	98%	95%
	U	98%	97%	93%
	F	100%	100%	97%

Mean SOR Length

		p_{ij}		
		C	U	F
λ_i	C	38.40	40.70	44.67
	U	36.14	39.69	43.47
	F	28.38	31.46	39.52

Correlation: SOR Length
& Arrival Probability (λ_i)

		p_{ij}		
		C	U	F
λ_i	C	-0.20	-0.10	0.05
	U	-0.35	-0.33	-0.23
	F	-0.28	-0.39	-0.49

Correlation: SOR Length
& Distance to Closest
Charging Station (d_{iz_i})

		p_{ij}		
		C	U	F
λ_i	C	-0.12	-0.22	-0.33
	U	0.04	-0.24	-0.36
	F	-0.16	-0.23	-0.34

Note: The arrival probability (λ_i) and the transition probabilities (p_{ij}) vary from close (C) to charging stations, uniform (U) across the entire service area, and far (F) from charging stations. We use baseline parameters values. The optimal *1VMC-SOR* policy is obtained by solving *Single Threshold*.

day. Another direction for future work within the multiple-vehicle framework could consider a setting in which the EVSS consists of multiple vehicle types. In our case, we assume that all vehicles are homogeneous, but many VSSs have several vehicle types (i.e. sedan and SUV or e-scooters and e-bikes) whose functionalities are all different. On the behavioral revenue management side, there are several fascinating directions. For instance, in a ride-sharing setting, Cohen, Fiszer, and Kim (2018) compare the effectiveness of immediate ride discounts versus future ride credit. In the same spirit, an interesting question is examining the long-term implications that discounts and fines have on customer retention and ridership in a dockless, EVSS.

CHAPTER 2

Estimation of Price and Spatial Elasticity

In this chapter, we use two different datasets from our industry partner, a software company that develops a suite of tools to transportation-related operations, such as manage parking and permitting, for cities, universities, and organizations throughout the North America. In many cities, our partner provides a mobile phone application for on-street parking payments. Using the detailed transactions panel data from this mobile phone application, we are able to estimate the price elasticity and spatial elasticity of parking.

In Chapter 2.1, we estimate the price elasticity of parking demand using a regression discontinuity framework. The dataset comes from a mid-sized U.S. city's mobile phone application for parking payments and spans several years before and after an announced price change. Following an increase of 20% from $1.25 per hour to $1.50 per hour, we find the average price elasticity of parking demand to be between -3.42 and -1.57, which is slightly higher than existing estimates. Our study, to the best of our knowledge, is the first to use transactions data from a mobile phone application for parking payments, so this could be driving the higher estimates. We also use our model to explore how long it takes for customers to learn about and respond to the price change. We find that it takes between 6 to 8 weeks after the price change for users to respond to change their behavior. This duration aligns with the mean inter-purchase time on the pre-price change data, suggesting that it takes about one transaction for customers to experience the price change firsthand before responding.

In Chapter 2.2, we estimate spatial elasticity, a measure of how individuals quantify the

cost of walking an additional mile, in an urban transportation setting. We leverage a panel dataset of parking transactions spanning 21 months from a large U.S. city's mobile application. During this time frame, there was an unannounced pricing error where two neighboring blocks were discounted by 67% for 16 months. Using this natural pricing experiment, our identification strategy involves using the pre-exposure data to build a person-specific distribution over true, intended destinations, and then using the post-exposure data to estimate how distance affects customer's actual destination choice. In a perfect foresight model where consumers have full information on all prices and know their exact duration before transacting, we find that customers require approximately $81 to walk an additional mile to their intended destination. This estimate increases 13% in the presence of rain and 36% during the morning rush hour. Under alternative utility specifications that relax the perfect foresight assumptions, we find that customers may be more sensitive to distance. The estimates we compute are valuable in any context where walking is a mode of transportation, such as metropolitan facility location and spatial pricing for dockless, vehicle and bike sharing. However, we note that the estimates will vary based on the realities and demographics of the city from where the data was collected. For instance, neighborhoods surrounding universities tend to have price-sensitive students, which can affect

2.1 Price Elasticity of Parking

2.1.1 Introduction

The economics of on-street parking have been studied since automobiles became popular (Vickrey, 1955; Roth, 1965; Gillen, 1978). One reason for the interest in research on parking is that a significant portion of metropolitan traffic congestion is caused by individuals "cruising," or circling the block, in search of a vacant, convenient parking space. Thirty percent is the most commonly used figure by academics and policy makers for the share of traffic attributable to cruising (Shoup, 2006). Other estimates range from as low as 15% to as high as 50% (Hampshire and Shoup, 2018; Arnott and Rowse, 1999). In a neighborhood in Los Angeles, California, it has been estimated that cruising causes over 3,500 miles of superfluous travel (Shoup, 2017). One of the major, hypothesized causes of cruising is that public parking is severely under-priced. However, correctly priced public parking can mitigate cruising and traffic congestion, and also achieve other benefits, such as lower greenhouse emissions, fewer accidents, and cleaner air, can also be realized. In order to price public parking in a manner that can reduce congestion, city planners must understand how parking demand fluctuates with price. In other words, transportation agencies that want to implement congestion-aware, demand-based pricing, first need to know the price elasticity of parking. In this paper, we use transactions data from a mid-sized U.S. city's mobile phone application for metered parking payments to estimate this elasticity.

Accurately setting the prices of parking, tolls, and other transportation modes can have implications beyond revenue gained and the traffic congestion that is induced by the prices. For instance, Christensen and Osman (2021) run a field experiment with Uber in Egypt and find that a 50% price discount quadruples ride-sharing usage and leads to a 42% increase in total travel, but they also find that the gains are realized more by women, who are less mobile and feel less safe on public transit. In another study, Simeonova et al. (2019) investigate the impact of a tax on entering the central business district in Stockholm, Sweden, by car.

They find that after this policy was instituted, inner-city traffic dropped 20–25%, and even more impressive is that ambient air pollution decreased 5–15% and the rate of acute asthma attacks among young children decreased between 16-50%. These studies highlight the impact that price changes can have beyond the immediate transportation setting.

Using parking transactions data from the mid-sized U.S. city's mobile parking application, we examine the impact of a 20% increase in the price of public, on-street parking after a publicly announced price change. We specifically study how a price change from $1.25 per hour to $1.50 per hour affects the total volume of transactions and the amount of time it takes for customers to respond to the price change. The dataset we received is from a privately-held technology company that provides transportation and mobility software solutions to cities across North America.

2.1.1.1 Related Literature

There exists extensive research on the price elasticity of parking and the various methods available to compute it. In a meta-study of over 50 articles that examine the price elasticty of parking, Lehner and Peer (2019) distinguish between price elasticity of occupancy (EPO), dwell time (EPD), and volume (EPV), where EPO is the sum of EPD and EPV. Furthermore, they find that the elasticities based on stated preference (SP) data (i.e., detailed surveys) lead to different managerial and policy insights than estimates based on revealed preference (RP) data. For the (dis)advantages of SP versus RP data in a transportation context, we refer the reader to Chapter 4 of Dell'Olio et al. (2017). The majority of parking elasticty estimates come from SP data research. By comparison, there are only a few articles using RP data from actual customer transactions during real pricing experiments, which is where our work is positioned. In the context of the meta-study, our work classifies as (1) an RP study since we examine the actual choices that individuals have made and (2) an EPV study since we examine how the total transaction volume changes after the price change.

There are several papers that estimate the price elasticity using transactions data and

our work contributes to this literature. For example, Yan, Levine, and Marans (2019) use a joint parking and mode choice model to estimate the elasticity for quarterly parking permits. In contrast, we use a detailed, daily-level panel data of transactions to estimate the elasticity. Kelly and Clinch (2009) and Ottosson et al. (2013) both focus on how elasticities change as a function of time of the day. In the former, the authors focus on EPO and use the transactions from on-street parking stations before and after a 50% price increase in Dublin, Ireland, and find that price elasticity by time of day ranges from -0.15 to -0.61 during weekday, with demand being more sensitive in the mornings. They also investigate how parking duration changes due to the price change, which we do not. In the latter, the authors use data from parking meters in Seattle, Washington and calculate the price elasticity of on-street parking by time of day at the block level and as a function of distance to the city's center. Since the authors did not have data on parking occupancy, they infer occupancy from the meter payment data. Both of these studies find that, while elasticisty of parking varies depending on the time of day and the specific location or block, parking is, on average, inelastic in the short-term and is likely elastic in the long-term.

Some of the most recent and closely related research to ours analyzes data from SFpark, a large-scale, controlled, dynamic pricing experiment in San Francisco, where prices were adjusted on blocks that deviate from a 60-80% occupancy target (Pierce and Shoup, 2013). The SFpark parking initiative was designed to improve the utilization of on and off-street parking, and a key aspect is that the project includes both treatment and control areas. The San Francisco Municipal Transportation Agency (SFMTA) managed the program and collected both parking sensor data, which consists of hourly block-level occupancy rates and on-street meter payment data containing all parking transactions. However, the sensor data eventually became unreliable and incomplete due to battery failures and sensor outages. Pierce and Shoup (2013) calculate the elasticity of demand revealed by over 5,000 price changes during SFpark's first year, and find that elasticity varies according to location, time, day of the week, initial price of the block, and the date of the price change. They also find that

short term-parking is inelastic whereas longer-term parking likely is elastic. These takeaways align with other studies that use the SFpark data (Fabusuyi and Hampshire, 2018). Millard-Ball, Weinberger, and Hampshire (2013) rebut the estimates and methodological approach in Pierce and Shoup (2013), and instead present estimates using a regression discontinuity methodology, which we also use. Using this methodology, they find parking to be less elastic than what Pierce and Shoup (2013) estimates for short-term parking suggest. Millard-Ball, Weinberger, and Hampshire (2014) also analyze SFpark data, but they focus on cruising and find that the SFpark program reduced cruising by 50%. Feldman, Li, and Tsai (2020) use the same data and find similar estimates of elasticity as (Pierce and Shoup, 2013). However, they focus on the program's welfare implications.

2.1.1.2 Contributions

This paper on the price elasticity of parking differs from the current literature in several ways. First, to the best of our knowledge, there are no studies using transactional data from a mobile phone application (app) that allows users to pay for parking via the app to estimate price elasticity. This is important because it gives us precise counts of parking volume. The price elasticity estimates reported in previous studies use data from meters, which is often aggregated at the hour level or can be unreliable due to logging errors. We note that studies that estimate the share of cruising use GPS data from mobile phones. The transactional dataset we use has high granularity, a panel structure (i.e., it contains timestamps and unique identifiers for each individual), spans several years, and comes directly from the city's mobile app. Because our dataset only contains transactions by customers who pay via the app, we cannot estimate EPO, and instead focus on EPV.

Second, we use a regression discontinuity (RD) approach to estimate the elasticity of demand. We are aware of only one study that uses a similar approach (Millard-Ball, Weinberger, and Hampshire, 2013). Our work differs in that we estimate the elasticity by measuring how the adoption rate of the mobile app changes, whereas Millard-Ball, Weinberger, and

Hampshire (2013) measures how the raw volume changes. We do note that other empirical studies using detailed transactions data have used RD to compare how parking occupancy, as opposed to parking volume, changes before and after a price change.

Third, we adopt a behavioral perspective by estimating the time it takes customers to learn about a price change after it goes into effect. The existing literature does not study this behavioral question of "how long does it take for customers to respond to a price change?" Even though customers knew about the price change six weeks before it went into effect, our analysis suggests that it takes users one transaction post-price change to modify their behavior.

2.1.1.3 Paper Organization

The paper is organized as follows: Section 2.1.2 describes the setting and the data used in the estimation. Section 2.1.3 describes the model and estimation strategy, and presents our main elasticity estimates. The approach and results for understanding a customer's time to learn about the pricing change are in Section 2.1.4. Finally, we conclude with a summary, discussion of policy implications, and several extensions in Section 2.1.5.

2.1.2 Setting and Data

This section provides details about our industry partner, setting, and data we use in the estimation.

2.1.2.1 Setting & System Dynamics

Our collaborator develops a software-as-a-service platform to help cities, universities, and organizations throughout North America manage their transportation infrastructure. One service they provide to its customers is a mobile phone application (app) for processing payments from parking lots and on-street, metered parking. Users always have the ability

to pay in-person with cash or credit card, but those who pay via the app benefit by saving time and receiving notifications before their meter is about to expire, in which case they can remotely buy additional time. Cities also benefit by digitizing their revenue and having access to real-time usage across their network. Users who pay via the app pay a convenience and credit card processing fee of $0.25 per transaction. Two examples of these types of mobile phone apps are ParkMobile[1] and PayByPhone.[2]

Our partner city is a mid-sized, U.S. city with approximately 100,000 residents. In the downtown, central business district (CBD), the hourly parking rate increased from $1.25 per hour to $1.50 per hour on April 1, 2017. The city council voted on this increase approximately six weeks before the price change was to go into effect. The CBD has both on-street, metered parking and off-street (i.e., lot and garaged) parking available. Since the price change only affected on-street parking, our study only includes on-street parking transactions.

In the CBD, parking is enforced from 8:00AM to 6:00PM from Monday through Saturday. Occasionally parking enforcement can extend beyond these times and days due to special events. Parking spaces are grouped into large, multi-city block areas called zones. The timeline of a parking transaction is as follows. Users search for an open space to park near their desired destination. When they find a parking space, they pay for it for a fixed amount a time. The maximum length of stay allowed in the CBD is 2 hours. Users can pay either directly at the meter (with cash or credit card) or via the mobile app. To pay on the app, they must locate the zone number on the meter or on a street sign next to the parking spot (see Figure 2.1) and enter this information in the mobile app to purchase time. Regardless of how users pay, they must vacate their spot before their time expires; otherwise, they could be fined for occupying a parking space without paying to park there.

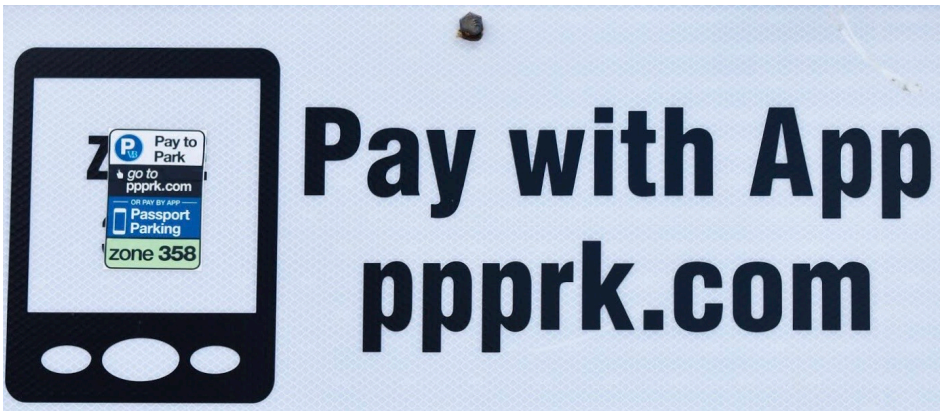
[1]See https://parkmobile.io/

[2]See https://www.paybyphone.com/

Figure 2.1: Example of Stickers and Signs with Zone Numbers in CBD.

(a)

(b)



(c)

Note: the images are sourced from Virgina Beach, Virginia's Parking Management website; see: https://www.vbgov.com/government/departments/sga/parking-management/pages/parking-meters.aspx

2.1.2.2 Data Description

We consider a panel dataset that includes the transactions of all customers paying their parking fees via the mobile app over a period that lasts over four years between 2012 and 2018. For each transaction, we know the unique customer identification code, zone name and number, starting timestamp, duration, rate information, and, in some cases, the specific space number.

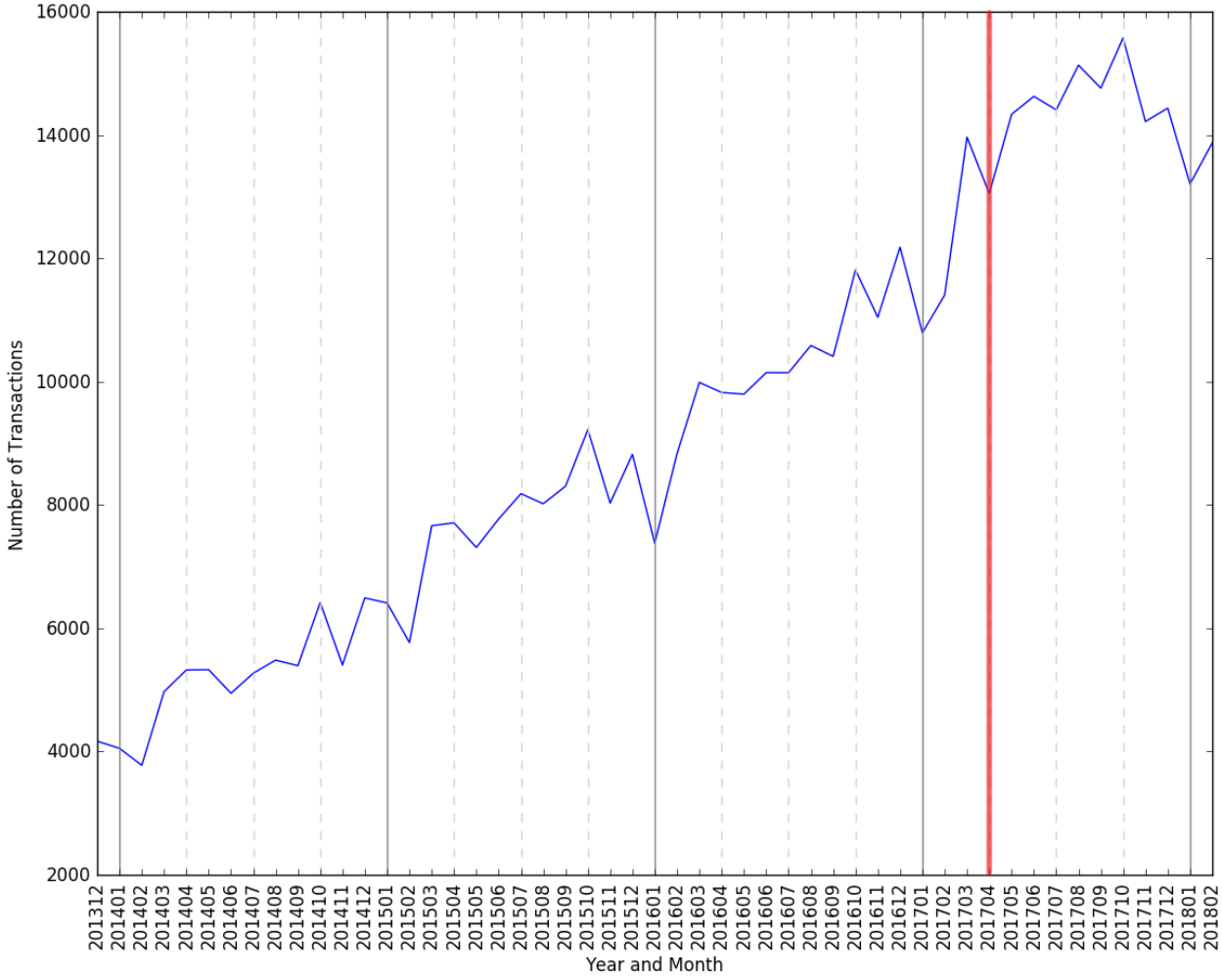
We use the day of the price change, April 1, 2017, to split the data into two periods: First, the pre-treatment period, which includes all transactions through March 31, 2017. Second, the post-treatment period, which includes all transactions beginning April 1, 2017. In our analysis, we define T_{PC} to be the date of the price change. Since T_{PC} is a Saturday, we also redefine weeks to start on Saturdays and end on Fridays. As we describe in Section 2.1.3, we perform our analysis at the week-level, so we also define W_{PC} to be the week of the price change, where weeks strictly before W_{PC} fall in the pre-treatment period and weeks on or after W_{PC} fall in the post-treatment period.

The data only includes transactions on the mobile application and does not contain payments made directly at the meter. As shown in Figure 2.2, which plots the number of transactions over our observation window, adoption and usage of the mobile app increased over time. The figure indicates that the growth in the volume of TRANSACTIONS slows down and flattens after the price change came into effect. Because of this, our model considers the adoption rate and the increasing volume of transactions per week, and then measures how these figures shift after the price change.

2.1.2.3 Data Preparation

Our initial data contains 103,153 customers who completed 572,538 parking transactions in nine parking zones from November 19, 2013 through March 5, 2018. Before engaging in any analysis, we complete several data cleansing and pre-processing steps:

Figure 2.2: Transactions Per Month From 12/2012 through 1/2018.



The x-axis contains the year and month expressed as YYYYMM in ascending order. The red, vertical line is the month of the price change.

- Remove transactions from 2 zones, resulting in the removal of ∼10 transactions.
- Remove transactions where customers received a discount or parked for free, which affected ∼500 transactions.
- Remove transactions on Sundays (∼6,000 transactions).[3]
- Remove transactions before 8:00AM.[4]
- Only include transactions from January-2014 through February-2018.

This leaves us with 524,405 transactions spread in seven parking zones. To give a sense of the distribution of transactions across zones, in Table 2.1, we show the number of transactions by zone for the pre- and post-treatment periods. We refer to the zones with the number assigned to them in our dataset from our partner. Zone 44 is the zone with the most transactions, so we use it as a reference point in our model and also refer to it as the "main" zone. During the range that our data spans, we confirmed that the size of the zones did not change. In our estimation approach, which we describe in Sections 2.1.3.1-2.1.3.2, we describe how we restrict the date range of the dataset to have an equal number of months in both the pre- and post-treatment periods.

We note that we do not combine "re-up" occurrences in this study. A "re-up" occurs when a customer consecutively buys multiple blocks of parking time, or re-purchases time within same zone after their initial time expires. Most often, customers combine these transactions in order to park for a longer time than what is allowed. While this is a parking violation, it is quite common in practice and our data reflects this. These transactions in a "re-up"

[3]Parking is free on Sundays, but due to a few special events, occasionally parking was not free on several Sundays.

[4]Parking is free before 8:00AM and after 6:00PM. Due to special events, parking was not free outside of this time range, and fees were assessed. There are approximate 80,000 transactions between 6:00pm and 12:00AM in the dataset, so we keep these, but only less than a few hundred between 12:00AM and 8:00AM, so we remove these. We suspect the transactions between 12:00AM and 8:00AM are due to data logging errors.

chain can be collapsed into a single transaction where the duration and transaction cost are both updated to reflect the totals across the entire chain. In this study, we do not combine these transactions because we do not study the impact of the price change on total duration or occupancy, but instead are only interested in how the price change affects the volume of transactions.

Table 2.1: Transactions By Zone for Pre- and Post-Treatment Periods

Zone Number	Pre-Treatment	Post-Treatment	Total
Main	62,718	33,277	95,995
41	57,408	25,789	83,197
42	38,804	22,740	61,544
43	47,116	25,109	72,225
45	47,434	25,555	72,989
46	34,908	17,587	52,495
48	57,401	28,559	85,960
Total	345,789	178,616	524,405

2.1.3 Estimating Price Elasticity

This section describes our modeling approach, which we use to obtain estimates of the price elasticity of parking, and the results.

2.1.3.1 Model

To study how parking demand changes after the price change, we develop four models that allow us to ultimately estimate the price elasticity of parking demand. These are designed using the regression discontinuity (RD) framework; we refer the reader to Imbens and Lemieux (2008) for detailed RD information.

In all of the specifications, we define Z to be the set of zones in the datset and let $z \in Z$ refer to an arbitrary zone. Due to high variability in the weekly and daily time series data, we perform our analysis at the week level. We define $Week$ to be the week number in the

dataset and we always begin numbering weeks at 1. For robustness, we vary the size of the dataset to measure how the estimates vary with the amount of data before and after T_{PC}. One implication of this numbering approach is that W_{PC} takes a different value depends on the number of weeks that we include in the estimation data.[5] In all models, the dependent variable is $Y_{z,Week}$, the number of transactions in zone z during week number $Week$.

Eqs. (2.1)-(2.4) contain the four models. The first is the baseline model and follows the canonical RD design and contains a fixed effect for each zone (INT_z), a zone-specific coefficient for $Week$ ($\beta_{z,1}$), and a single post-treatment adjustment on $Week$ (β_2), which is the amount the demand changes after the price change. This is the main coefficient of interest. At the week of the price change ($Week = W_{PC}$), there is a discrete jump or discontinuity in demand. If $\beta_2 > 0$, the jump is upwards and the demand increases in weeks beginning W_{PC}, and if β_2 is negative, the jump is downwards and the demand decreases.

The next two models, in Eqs. (2.2)-(2.3), are generalizations of the first model. In Eq. (2.2) we add a second intercept (INT_2), that only affects the fit in weeks on and after W_{PC}. The model in Eq. (2.3) is a further relaxation as we let the second intercept be zone-specific ($INT_{z,2}$). In this sense, we are adding a second fixed effect for zone that only kicks in when $Week \geq W_{PC}$. Both of these extensions result in better fits, which is seen by the R^2 being greater than in the baseline model.

Instead of allowing a discontinuity at $Week = W_{PC}$, the last specification, in Eq. (2.4), enforces a continuity at this changeover point. In effect, this enforces a "hinge" at the joint $Week = W_{PC}$, and is sometimes called a "hinge estimator" or a "hockey stick regression" due to the resemblance of the fitted regression line to an ice hockey stick. This approach has a rich history, and its variants have been studied by many researchers (Quandt, 1960; Sprent, 1961; Hudson, 1966; Vieth, 1989; Barrowman and Myers, 2000). While Eqs. (2.2)-(2.3) are less constrained versions of the baseline model, this fourth specification is neither more or

[5]This numbering system does not effect the efficacy of our estimates; we include this detail to clearly define our approach.

less constrained version of the baseline; it is simply a different model.[6]

Basic Model

$$Y_{z,Week} = INT_z + \beta_{z,1} \cdot Week + \beta_2 \cdot Week \cdot \mathbb{I}[Week \geq W_{PC}] \tag{2.1}$$

Two Intercept Models

$$Y_{z,Week} = INT_z + \beta_{z,1} \cdot Week + (INT_2 + \beta_2 \cdot Week) \cdot \mathbb{I}[Week \geq W_{PC}] \tag{2.2}$$

$$Y_{z,Week} = INT_z + \beta_{z,1} \cdot Week + (INT_{z,2} + \beta_2 \cdot Week) \cdot \mathbb{I}[Week \geq W_{PC}] \tag{2.3}$$

Hinge Estimator Model

$$Y_{z,Week} = INT_z + \beta_{z,1} \cdot Week + \beta_3 \cdot (W_{PC} \cdot \mathbb{I}[Week < W_{PC}] + Week \cdot \mathbb{I}[Week \geq W_{PC}]) \tag{2.4}$$

2.1.3.2 Methodology

To accurately measure the price elasticity of demand for parking, we focus on customers with transactions before and after the price change. Some customers only transact before T_{PC} and some only after, but these individuals are uninformative because we do not know their

[6]While the hinge model does add a constraint to the baseline since it enforces a hinge, the fit is not always worse than the baseline case due to technical details of the model; we refer the reader to the aforementioned articles for a discussion of this.

behavior in the period we do not have data for them. From a methodological perspective, the RD framework does not require the panel units (i.e., customers) to be the same. In fact, when we do not restrict our analysis to customers with pre- and post-treatment transaction, our results hold convincingly, so we are being conservative by only focusing on customers with data before and after T_{PC}.

We estimate the price elasticity by using different slices of data. Each slice is created by varying the amount of data we include before and after the price increase. First, we only include those customers with at least one transaction Δ months (where one month is equivalent to four weeks) before T_{PC} (i.e., before $T_{PC} - \Delta$) and Δ months after T_{PC} (i.e., after $T_{PC} + \Delta$). Next, in the estimation, we only use the transactions in the date range $[T_{PC} - \Delta, T_{PC} + \Delta]$. Table 2.2 contains the number of customers and transactions in the data slices given by the different values of Δ. We note that in the first row, we include all customers, which contain individuals who only transacted before or only after, and in the second row, we include those individuals who transact at least once before and after the price change. All subsequent rows include customers who transact at least once Δ months before and after the price change.

2.1.3.3 Estimation Results

We present the parameter estimates of each model in Table 2.3, using $\Delta = 3$ months (12 weeks). In our robustness checks, we also perform the estimation for $\Delta > 3$ months. In Figure 2.3, we plot the dependent variable $Y_{z,Week}$ for each zone, along with the linear estimates using the outputs in Table 2.3. For robustness, in Appendix B.1, we also present the results for $\Delta = 0$ months for various date ranges and for the cases when $\Delta \in \{4, ..., 11\}$ months. From these additional checks, we notice that as Δ increases from 4 to 11 months, the magnitude of β_2 decreases. Or, as the window of consideration around W_{PC} expands, the effects weakens. This is maybe to be expected as the RD methodology literature suggest using a smaller window around the policy change date for the most accurate estimates.

Table 2.2: Descriptive Statistics

	Entire Dataset		Data Range: $T_{PC} \pm \Delta \cdot 4$ Weeks	
Δ (Months)	# of Customers	# of Transactions	# of Customers	# of Transactions
–	94,985	524,405	–	–
0	12,434	315,921	–	–
3	9,561	281,542	6,498	48,104
4	8,574	269,740	6,406	58,462
5	7,675	256,260	6,105	67,039
6	6,843	244,075	5,716	74,899
7	6,026	229,953	5,205	80,839
8	5,138	213,772	4,594	84,335
9	4,330	194,884	3,964	84,187
10	3,372	171,589	3,188	80,764
11	2,225	136,917	2,143	70,405

Note: We start with $\Delta = 3$ months because anything smaller than this did not give us enough data to accurately measure the effect.

Finally, we note that in the results in Table 2.3, we include $\beta_{z,1}$ for each of the zones listed in Table 2.1.

2.1.3.4 Price Elasticity

Using the coefficients from the estimation output, we can calculate the price elasticity using the formulas in Eqs. 2.5-2.6. The first is based on the standard definition, which states the elasticity is equal to the percent change in demand divided by the percent change in price. We know the price increased by 20% from \$1.25 per hour to \$1.50 per hour, so this is in the denominator of Eqs. (2.5)-(2.6). The second formula is based on the arc-log formula, which accounts for the non-linearity of the demand curve in parking contexts (Lehner and Peer, 2019). Regardless of which formula is used, if parking is elastic, the numerator will be negative, indicating demand decreases with a price increase. We note that for small values of $\frac{\beta_2}{\beta_1+\beta_{z,1}}$, both formulates yield similar results.

$$\frac{\%\text{ Change in Demand}}{\%\text{ Change in Price}} = \frac{\frac{\beta_2}{\beta_1+\beta_{z,1}}}{0.20} \tag{2.5}$$

$$\frac{\log\left(\frac{\text{New Demand}}{\text{Old Demand}}\right)}{\log\left(\frac{\text{New Price}}{\text{Old Price}}\right)} = \frac{\log\left(1+\frac{\beta_2}{\beta_1+\beta_{z,1}}\right)}{\log\left(1.20\right)} \tag{2.6}$$

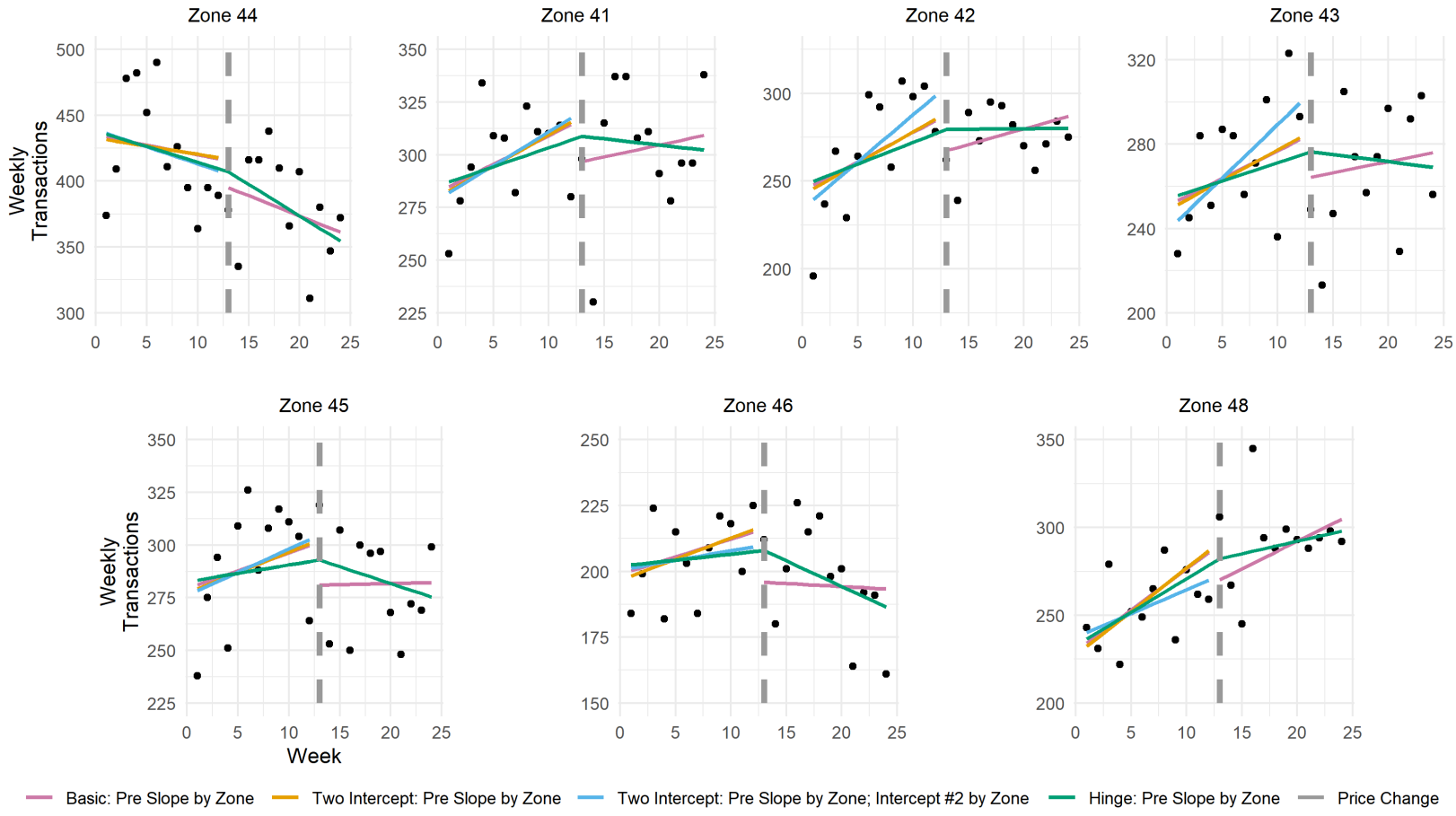
In Table 2.4, we report the price elasticity from estimating the baseline model (Eq. 2.1) for different values of Δ. We report the minimum, maximum, and average elasticity over all zones except the main zone ($z = 44$). Since we use zone fixed effects with zone 44 as the reference zone, we do not report an estimate $\beta_{1,z=44}$ in Table 2.3, and therefore do not include this reference zone in the elasticity calculations.

We first note that the sign of our estimations agree with our intuition: as prices increase, demand decreases. Next, we observe that our estimates align with the estimates

Table 2.3: Combined Results: PIDs with $\Delta = 12$ Weeks; Data Range: 12 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	-1.51	-1.24	-2.60	-2.39^{**}
$Week\,(\beta_{z=41,1})$	4.20^{***}	4.20^{***}	5.82^{***}	4.20^{***}
$Week\,(\beta_{z=42,1})$	4.85^{***}	4.85^{***}	7.97^{***}	4.85^{***}
$Week\,(\beta_{z=43,1})$	4.13^{***}	4.13^{***}	7.67^{***}	4.13^{***}
$Week\,(\beta_{z=45,1})$	3.19^{***}	3.19^{***}	4.81^{**}	3.19^{***}
$Week\,(\beta_{z=46,1})$	2.84^{**}	2.84^{**}	3.32	2.84^{**}
$Week\,(\beta_{z=48,1})$	6.21^{***}	6.21^{***}	5.31^{**}	6.21^{***}
$Week\,(\beta_2)$	-1.57^{***}	-2.58^{**}	-2.58^{**}	
W_{PC} or $Week\,(\beta_3)$				-2.40^{**}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	168	168	168	168
R^2	0.84	0.84	0.85	0.83
Residual Std. Error	26.57	26.57	26.34	26.85
F Statistic	55.36^{***}	51.72^{***}	37.99^{***}	53.95^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 48,104 transactions and 6,498 parkers.			

Figure 2.3: Transactions per Week ($Y_{z,Week}$) for Zone with Each Estimation Fit for $\Delta = 12$.



in the literature (Lehner and Peer, 2019). Existing estimates have been in the following ranges $[-1.89, -0.21]$ (Yan, Levine, and Marans, 2019), $[-3.00, -0.30]$ (Kanafani and Lan, 1988), and $[-1.80, -1.20]$ (Albert and Mahalel, 2006). Yan, Levine, and Marans (2019) use a discrete choice modeling framework, Kanafani and Lan (1988) use a simpler approach leveraging raw vehicle counts before and after the price change, and Albert and Mahalel (2006) use stated preferences data analysis. In contrast, our estimates are $[-3.42, -1.57]$, suggesting that parking is more elastic than what other studies have found. We also note that the log elasticity formula returns larger estimates.

One reason that our estimates may be higher is because prices are directly accessible via the mobile app, and are thus more salient than prices posted directly at meters. It is well-known in economics that salient pricing has larger effects on demand, which aligns with our estimates. It is also worth noting that our estimates could be conservative because our methodology does not include nor account for parkers who stop using the app in response to the price change. If we were to include these individuals, the magnitude of the estimates would be even larger.

Finally, we see that as the amount of time before and after the price change at week W_{PC} increases, the price elasticity decreases. Note that as Δ increases, it gets less likely that the identifying assumption required by RD holds, so for large Δ, the elasticity estimates have to be interpreted with caution. This is to be expected as Table 2.3 and Appendix B.1 show that β_2 decreases as Δ increases. This could suggest that customers initially react to price changes, but in the long-run, they become desensitized to the higher prices and demand reverts back to its usual, pre-treatment levels. The discussion of how customers respond to the price change motivates the next section, where we examine how many transactions it takes for customers to react to a price change.

Table 2.4: Price Elasticity as a Function of Δ

Δ (Months)	Standard Formula (Eq. (2.5))			Arc-Log Formula (Eq. (2.6))		
	Minimum	Maximum	Average	Minimum	Maximum	Average
3	-5.90	-1.67	-3.42	-14.95	-2.23	-6.65
4	-5.20	-2.12	-3.04	-17.58	-3.03	-6.40
5	-3.90	-1.60	-2.72	-8.31	-2.12	-4.71
6	-3.59	-1.81	-2.55	-6.93	-2.47	-4.14
7	-2.99	-1.40	-2.23	-4.99	-1.80	-3.35
8	-2.95	-1.30	-2.17	-4.89	-1.65	-3.20
9	-2.85	-1.42	-2.20	-4.62	-1.82	-3.25
10	-2.74	-1.36	-2.13	-4.36	-1.74	-3.11
11	-1.96	-0.96	-1.57	-2.72	-1.17	-2.09

2.1.4 Estimating Optimal Price Change Date

In this section, we describe how long it takes customers to react to a change in the price of public parking. To do this, we relax the assumption that customers change their behavior exactly on the day or week of the price change coming into effect. For example, customers could change their behavior before the price change, such as when the legislation passed, or at a later point, such as the next occasion when they park their car. Thus, we are no longer looking at W_{PC} as the week of the price change, but instead we want to see if another value of W_{PC}, such as $W_{PC}-1$ week, $W_{PC}+1$, or some other week, fits the data better. For a fixed Δ (months), we create a slice of the data (as described in 2.1.3.2) that spans weeks in the range from $W_{PC}-4\cdot\Delta$ to $W_{PC}+4\cdot\Delta$. We multiply Δ by 4 to clearly indicate weeks. Then, we let W_{PC} to take every possible value in $\{W_{PC}-4\cdot\Delta, ..., W_{PC}-1, W_{PC}, W_{PC}+1, ..., W_{PC}+4\cdot\Delta\}$. For each of these "fictitious" price change weeks, we record how well the model fits the data. We measure fit by sum of squared errors and pay particular attention to which fictitious price change week corresponds to the best fit. This will help illuminate when customers actually responded to the price change.

Using the true, actual value of W_{PC}, we estimate the model in Eq. 2.1, which gives us the estimated coefficients (i.e., $I\hat{N}T_z(\forall z), \hat{\beta}_1, \hat{\beta}_{z,1}(\forall z), \hat{\beta}_2$). Then, with these coefficients

and with every possible fictitious value of W_{PC}, we compute $\hat{Y}_{z,Week}$, the model's estimate of $Y_{z,Week}$. Finally, we compute the sum of squared errors (SSE), $\sum_z \sum_{Week}(Y_{z,Week} - \hat{Y}_{z,Week})^2$, to measure how well the model fits the data. A lower SSE indicates a better fit. We do this for every possible value of $\Delta \in \{3, ..., 11\}$.

We report the results in the heatmap in Figure 2.4, where the spectral color range is blue if the SSE is low (fit is high) and red if SSE is high (fit is low). We can only interpret the numerical SSE value within columns (i.e. within the same value of Δ) because the data in each column is constant. We cannot compare the SSE across columns because there is generally more data as Δ increases (see Table 2.2), so the SSE will likely be greater. This is why we use a heatmap to indicate where the SSE fit is best for each Δ. This allows us to visually compare across all values of Δ.

For nearly all data ranges, i.e., $\Delta > 16$ weeks, the SSE is lowest six to eight weeks after the price change. This shows that it took time until customers learned about and responded to the price change. Using all of the pre-treatment data, we inspect the inter-purchase (or interarrival) time of customers before the price change, i.e., the average time between two transactions by the same customer. We find that the mean time is six to nine weeks.[7] Table 2.5 contains these interarrival times for customers with at least two through six transactions before the price change.

These figures indicate that even though the price change legislation passed in mid-February, and press releases were made through the date of the price change on April 1, customers did not respond until 6-8 weeks after the price change, or approximately the mean interarrival time between transactions. This suggests that customers do not respond until they actually experience the price change firsthand. This notion of a "learning transaction" after an experience seems intuitive and natural, but, to the best of our knowledge, it appears to be understudied and not yet widely acknowledged in the literature.

[7]We use two or more transactions as the minimum threshold because we cannot compute the interarrival time between transactions for customers with only 1 transaction.

There are studies on the long term effects of interventions after the treatment period ends. These studies focus on how well customers remember and how quickly (or slowly) they forget, and behavior reverts to pre-intervention levels (Allcott and Rogers, 2014; Frey and Rogers, 2014; Simon and Spiller, 2016; Brandon et al., 2017). DellaVigna and Kaplan (2007) is tangentially related as the authors study teases out temporary versus long-term learning effects of a media news outlet on voting behavior. However, none of this related work explores the response time after an intervention starts or the response time after the first exposure to the intervention.

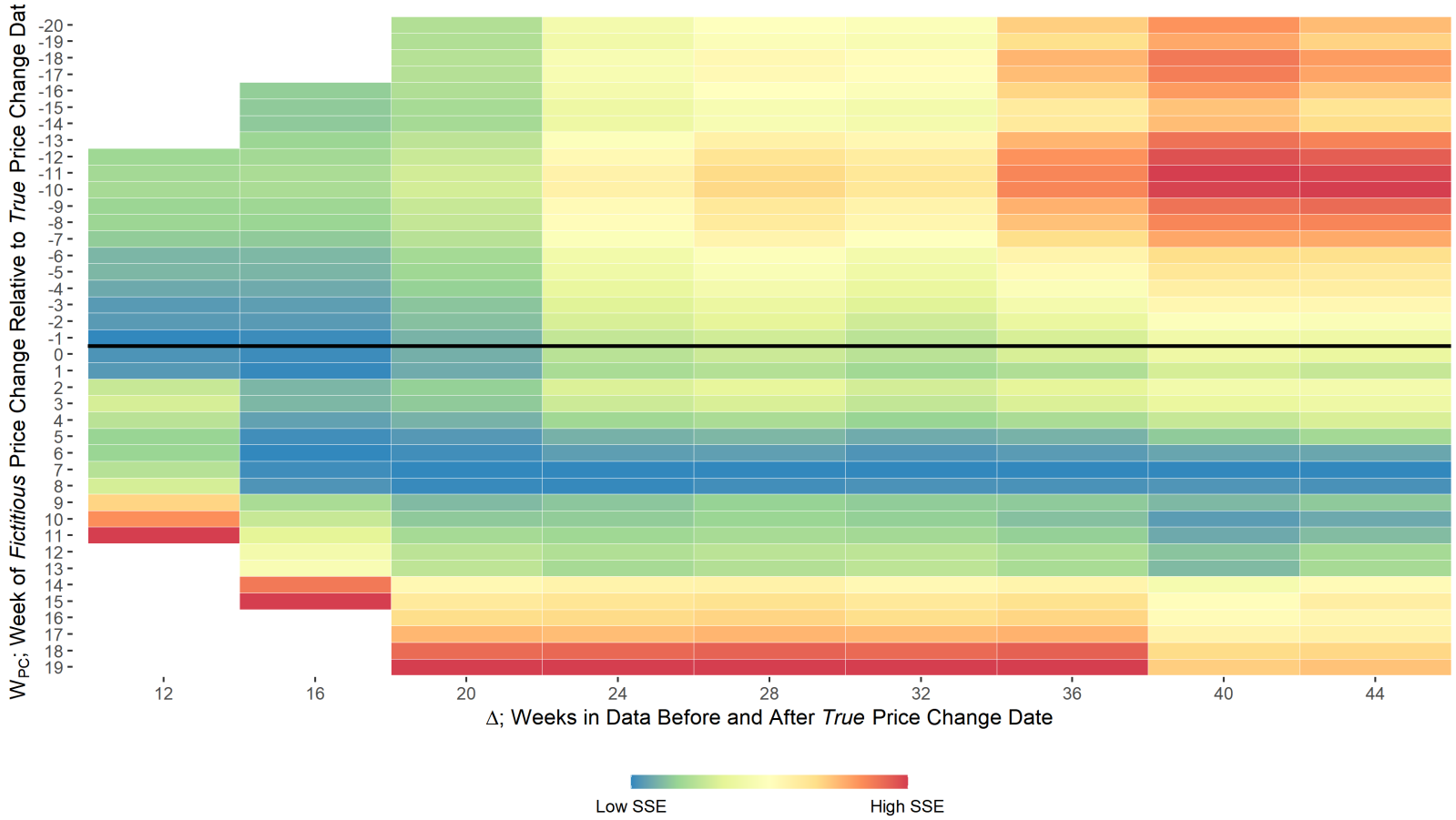
Table 2.5: Mean Interarrival Time During Pre-Treatment Data

Minimum Number of Pre-Treatment Transactions per Customer	Number of Customers	Mean Number of Weeks between Transactions
2	22,188	9.6
3	14,385	8.8
4	11,221	7.6
5	9,411	6.8
6	8,288	6.2

2.1.5 Discussion and Future Work

Using data from a mid-sized U.S. city before and after 20% increase in the hourly price of parking, this paper uses the regression discontinuity framework to estimate the price elasticity of parking. Though the price elasticity of parking demand has been studied previously, our paper contributes to the literature in several ways. First, our data comes from a mobile phone app that customers use to pay for parking. The existing literature estimated price elasticities with stated preference data. The studies that do use revealed preference data, such as our ours, rely on parking meter data, which can be less reliable and less granular than the data we have. To the best of our knowledge, our study is the first to estimate the price elasticity of parking using data on payment transactions from a mobile phone ap-

Figure 2.4: SSE from the Basic Estimation Model (Eq. 2.1) as a function of Δ and W_{PC}.



The y-axis represents the week of the imposed, fictitious price change that we are evaluating. The x-axis represents Δ, in weeks, which is the how we filter which customers to include in the analysis. The horizontal black line represents the week of the actual price change.

plication. Second, we develop several models that can be used to estimate the change in the weekly number of transactions before and after a price change while accounting for the increasing adoption of the app over time. Previous approaches using parking meter data simply measure the change in demand before and after the price change. Finally, we study the time after the price change it takes for customers to change their behavior. To do this, we leverage the panel structure of our data and quantify the inter-purchase time of customers in the time before the price change.

Our study shows that the price elasticity of parking lies between -3.42 and -1.57, which is greater than the existing estimates in the literature. These estimates suggest that parking is more elastic than what has been thought previously. Policy makers can use this information for price setting of public parking in metropolitan areas. For example, our industry partner, which provides a parking payment app for local governments in U.S. and Canadian cities, is beginning to offer dynamic pricing services for their parking clients. Many of the algorithms designed for the dynamic pricing of reusable resources (e.g., parking), assume knowledge of the price elasticity (Owen and Simchi-Levi, 2018; Rusmevichientong, Sumida, and Topaloglu, 2020). Even to compute the optimal static prices for reusable resources, knowing the price elasticity is vital (Besbes, Elmachtoub, and Sun, 2021).

2.1.5.1 Limitations and Future Work

Future research could extend our work and address its limitations. With parking, the price elasticity is likely to vary significantly across individual customers and parking occasions. While we have access to unique identifiers for each customer in our panel data, we do not have any socioeconomic or demographic information about the customers, such as the gender, income, or education level. Incorporating this information into our data could show which type of customers are more or less sensitive to price changes. Furthermore, our data comes from a central business district with many retail stores in a mid-sized city. The existing literature shows that elasticities vary depending on whether long-term parking is required,

the city size, and location type (i.e. retail versus non-retail).

While we have a rich, detailed dataset, our study only utilizes data from transactions that occur via the mobile app. We do not have any data on transactions that were made directly at the meter, e.g., with cash or credit card. Due to this limitation, we could be missing out on the accurate demand profile. Another issue is that the number of transactions is not stationary, as more users sign up for the payment app over time. We address this by using the weekly rate of transactions instead of the transactions per week. However this could be skewing our results and estimates to be higher than the existing literature. Replicating our study once demand has stabilized on the mobile phone app could be worthwhile.

2.2 Spatial Elasticity

2.2.1 Introduction

With ride sharing and vehicle sharing becoming more popular over the past several years, the pricing of these services has been highly scrutinized by the media and well studied by researchers. Of particular interest is the emergence of spatial, or location-based, pricing, where prices vary as a function of the precise geo-coordinates, in addition to the traditional factors, such as time, inventory, and forecasted future demand, that are present in classic dynamic pricing. Spatial pricing is particularly important in the context of micro-mobility, which includes various light transportation modes, such as electric scooters and bicycles, which have gained ridership across the world thanks to companies like Bird, Lime, and Spin who operate dockless, electric vehicle sharing systems (BBC, 2020). While shared vehicle fleets are not perfect (Observer, 2020), they are popular and convenient because they can be street-parked on any block or sidewalk in the service area and are affordable with a typical start-up price of $1.00 and per-minute rate between $0.15 and $0.40 (Santa Monica Daily Press, 2019). As a result, they are commonly used to travel the "last mile" or "micro mile," which are distances that are too far to walk but too close to drive. For example, the distance between a parking spot or a public transit center (e.g., bus stop or rail station) and an individual's desired destination (e.g., home, doctor's office, etc.) could take over 20 minutes to traverse on foot, but can be covered in less than 5 minutes with a fee of less than $5 on a dockless electric scooter.

In this paper, we study this relationship between distance (or time) and cost to answer "how much money would one pay to directly reach their intended, desired destination, instead of having to walk the last mile to reach this destination?" This trade-off, which we term "spatial elasticity," hinges on the assumption that individuals are utility-maximizing, which, in our setting, means they minimize their fiscal costs and walking distances. The former is tied to basic pricing theory of cost-minimization and profit-maximization, and the latter is

backed by marketing and economics research. For example, Bucklin, Siddarth, and Silva-Risso (2008) show that prospective car buyers are more likely to purchase vehicles as the distance to dealers decreases. Similarly, utilizing the locations of consumers and car dealers, Albuquerque and Bronnenberg (2012) find that consumers experience a disutility for travel. In a grocery store setting, Hui et al. (2013) use path tracking data in a supermarket to measure the impact of path length, or travel distance, on purchases and the effectiveness of mobile promotions. Rennhoff and Owens (2012) show that competition amongst churches decreases with distance. In all of these examples, consumer preferences and competition are both higher with shorter distances, underscoring the value that consumers put on close proximity to their destinations.

To estimate spatial elasticity, we study how a non-publicized pricing change in on-street, metered parking prices impacts customer choice. In particular, the shift in choice behavior after customers are exposed to the pricing change allows us to quantify how customers value their time against walking.

In modern transportation settings, having accurate estimates of spatial elasticity is critical. For instance, consider the micro-mobility setting. Proponents of theses shared mobility services contend they are solutions to traversing the micro mile and will therefore encourage more public transportation usage and adoption. However, there are anecdotal accounts of a dearth of dockless vehicles at key public transit hubs. One reason is because the vehicles are unevenly dispersed across the service area, leading to a unavailability at popular origins. This is typically assuaged through rebalancing and repositioning efforts, which is the practice of manually moving vehicles across the network in anticipation of future demand, but it is known to be costly and operationally intense (Fishman, Washington, and Haworth, 2014). Another form of rebalancing is called "rider-based rebalancing," where customers are incentivized to end their rides at key locations. For instance, Chung, Freund, and Shmoys (2018) study "Bike Angels," point-based incentive program in New York City's bikesharing system, and Nyotta, Bravo, and Feldman (2019) examine the effectiveness of offering free-

ride discounts to users who end their trips at charging stations. For either of these incentive schemes and pricing algorithms to be realistically implemented, operators need to precisely understand how their riders value walking an additional leg in their journey versus reaching their exact destination, which is exactly what spatial elasticity quantifies.

Beyond these pricing incentives in micro-mobility, spatial elasticity can be used in other aspects of revenue, operations, and transportation management. Ride share operators, such as Uber and Lyft, can inform the pricing of their services that require riders walk to designated pick-up points or accept a closeby drop-off point that is away from the specific, desired destination, in exchange for a cheaper fare (CNN, 2017). Fielbaum, Bai, and Alonso-Mora (2021) study this exact problem and assume that the willingness to walk is known. City planners should consider spatial elasticity when pricing and placing public transit locations and street parking options. In facility location models where walking is incorporated as a transportation option (Owen and Daskin, 1998), an accurate estimate of spatial elasticity is necessary. In the instance of delivery networks, Amazon can use it to optimally locate their locker pick-up locations (Deutsch and Golany, 2018) or place their in-city, metropolitan fulfillment centers to meet the tight, two-hour delivery window promises they offer with Prime Now (Amazon, 2020).

The parking transactions data that we use comes from a company that provides transportation and mobility software solutions to cities across North America. In one city, the metered parking spots on two blocks in a high-traffic, downtown area were accidentally underpriced by 67% from the true hourly rate of $3.75 to $1.25. This unannounced, discounted price was only accessible to users who pay via the mobile phone application. Users who paid directly at the meter paid the true rate. Our data only covers those users who paid via the mobile application. Our estimation framework involves maximum likelihood estimation (MLE) with the latent class logit (LC-MNL) discrete choice model, which is also known as the discrete, or finite, mixture of multinomial logit (MMNL) model. McFadden and Train (2000) define it as a mixed logit model with discrete mixing distributions, to emphasize the

similarities with the original, continuous-mixture logit model, and demonstrate the model's flexibility by showing it can accurately approximate any choice model based on random utility maximization. While this is a strong result that highlights the value of the continuous-mixture model, Hess et al. (2011) identifies the possible advantages of latent class structures over the continuous version, especially in a transportation context. We refer the reader to McFadden and Train (2000) and Hensher and Greene (2003) for more information on the LC-MNL and its variants. While none of these explicitly consider the LC-MNL model in an revenue management application, the model has been used in this domain. For instance, when customers chose according to the LC-MNL, Bront, Méndez-Díaz, and Vulcano (2009) and Feldman and Topaloglu (2015) provide methods for computing the revenue-maximizing assortments, and Li et al. (2019) study the multi-product pricing problem.

2.2.1.1 Contributions

In this paper, we have two main goals. First, from an empirical perspective, we want to estimate spatial elasticity. We do this by leveraging panel data that spans 21 months and examining changes in parking behavior before and after an unannounced pricing change is released via the mobile application. While the true destination is unobserved in our data and we only observe the actual parking destinations, our strategy involves using data from the pre-treatment period to create a distribution over the true, intended destination blocks. Then, using the during-treatment data, which includes all transactions after customers are exposed to the pricing change, we measure how behavior changes. Furthermore, we want to understand factors that may cause changes to the spatial elasticity, so we examine the impact of the presence of rain and the morning rush hour on the baseline estimates. We find that customers require approximately $81 to walk a mile to their intended destination. This estimate increases 13% in the presence of rain and 36% during the morning rush hour. In both scenarios, the impact on spatial elasticity attenuates as we expand the window of when rain must occur near a transaction time or the window of what constitutes "morning

hours." These core estimates hold under extensive robustness checks (see Section 2.2.4.4 and Appendix C.3 for details).

Secondly, we demonstrate how spatial elasticity impacts decisions. In a series of numerical experiments, we compare the performance of two dynamic pricing policies, one that considers spatial elasticity and one that does not, when customers are spatially-sensitive. We find that accounting for the spatial component can significantly increase revenue and satisfied demand– this effect is more pronounced in cities with shorter walking distances between zones or larger arrival rates.

To the best of our knowledge, our work provides one of the early estimates of spatial elasticity, and can contribute to the increasing use of empirical methods in revenue management. Recent examples in this area are Li, Granados, and Netessine (2014), who use structural estimation to determine the fraction of strategic consumers who delay purchases in anticipation of future price discounts, Fisher, Gallino, and Li (2018) who use field experiments to find measures of price elasticity and test their impact on pricing strategy, and Stamatopoulos, Bassamboo, and Moreno (2020) who use difference-in-differences to quantify the effect of physical menu costs on retail performance metrics.

The paper is organized as follows: Section 2.2.2 describes the setting and the data used in the estimation. Section 2.2.3 describes the model and estimation strategy. Results and robustness checks are presented in Section 2.2.4. To demonstrate the importance of spatial elasticity in an urban mobility setting, in Section 2.2.5 we perform a numerical study where we compute the optimal dynamic prices of parking spaces with and without spatial considerations. Finally, we conclude with a summary and discussion of several extensions in Section 2.2.6.

2.2.2 Setting and Data

This section provides details the setting and its dynamics, and the data we will use in the estimation.

2.2.2.1 Setting & System Dynamics

Our research partner operates a software-as-a-service (SaaS) platform to facilitate mobile payments for parking, permitting, and micro-mobility management for municipalties and organizations throughout the world. In many cities, our collaborator provides a mobile phone application (app) to aide in the management of parking lots and on-street, metered parking. While users still have the opportunity to pay in-person with cash or credit card, users who choose to pay through the app save time and are given notifications when their time is about to expire, in which case they can remotely buy additional time. Cities benefit by digitizing their revenue and having access to real-time usage across their network. ParkMobile (ParkMobile, 2021) and PayByPhone (Pay By Phone, 2021) are similar examples of mobile phone apps.

Parking spaces on city blocks (or zones), are priced at an hourly rate with a maximum duration, which, in most cities, is enforced during working hours (i.e., 8AM-6PM) from Monday through Saturday. The timeline of a parking transaction is as follows. Users looking for an open space park when they find one. Users who want to pay via cash or credit card directly at the meter can do so, but those who wish to pay via the mobile app locate the zone and space number on the meter (See Figure 2.5), and use this information in the mobile app to purchase time. Those who pay with the app are subjected to a flat, 15-cent transaction fee. Regardless of how users pay, they must vacate their spot before their time expires. If not, they could receive a fine from parking enforcement for occupying a parking space without paying to park there. Of course, customers can always purchase more time up until the maximum, allowable time limit. If customers procure multiple, consecutive transactions

at the same spot without moving their vehicle, they are also risking a fine as this would qualify as a single transaction beyond the maximum, allowable limit. In a popular, high-

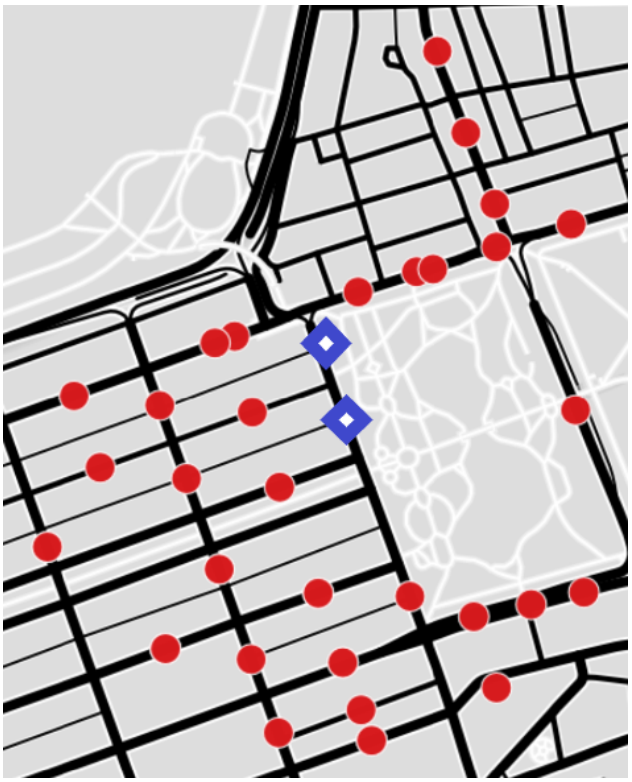
Figure 2.5: Example of a Sticker With the Zone Number on Parking Meters

traffic neighborhood in our partner city, which is just under 50 square miles, there is only on-street, metered parking and there are not any parking lots. For 16 months, two adjacent blocks, containing a total of 14 parking spaces, were priced incorrectly, but only via the mobile app. Those who paid with cash or credit card at the meter were subjected to the true hourly rate of $3.75, but those who chose to pay via the app paid the incorrect, reduced rate of $1.25 per hour. This error occurred because the wrong pricing code was entered in the back-end system. The error was not public knowledge and only those who parked in the affected zones after the pricing error was live and paid via the mobile app realized the discount. It is possible that the affected subset of users who experienced this pricing change modified their parking behavior in response to these newly discounted prices. Since we have users' parking behavior both before and after the pricing change, this error is giving us the pricing variation we need to measure how customers value the total cost with the distance they need to walk between zones.

We group the two, mispriced blocks and refer to the pair as "treated" zones. We restrict our analysis to the quarter-mile radius around the treated zones, with the assumption that

most individuals will not walk more than 0.25 miles for a parking spot. We refer to this circle as the "neighborhood" or the "catchment area." During our analysis period, neither the capacity (i.e., number of spaces) of each zone nor the signage in the neighborhood changed. We provide a map of the catchment area, with the treated and untreated zones, in Figure 2.6.

Figure 2.6: Map of the Catchment Area



Note: Diamonds denote the two treated zones and solid circles denote the non-treated zones.

2.2.2.2 Data Description

We consider a panel data that includes all transactions from all customers who pay with the mobile application during a 21-month timeframe in 2017-2018. The data does not include transactions that occur directly at the meter. Approximately 40% of all transactions take

place via the app. For each transaction, we know the unique customer identification code, zone name and number, starting timestamp, duration, and rate information. Our data spans two periods: (1) the pre-treatement period, which lasts nearly five full months in 2017, and (2) the during-treatment period, which lasts just over 16 months spanning 2017 and 2018. We use "before treatment" or "before pricing error" and "during treatment" or "during pricing error" to refer to these time periods. Our data ends approximately two weeks before the pricing error is discovered and corrected. In the catchment area, customers do not specify a space number when procuring time, so our data only contains the zone number of the transaction. As such, we do not know the precise parking location in the zone, so we assume that all transactions on a particular block occur at the block's centroid. We cull the geocoordinates, i.e. latitude and longitude, for each block's center point information into a supporting dataset using Google's Geocoding API.

Data Preparation. We begin with 105,401 customers and 609,516 total transactions in the catchment area. To prepare our dataset for our estimation, we complete several pre-processing steps. First, there are instances where zones are re-assigned unique identifiers over the time horizon, so we rectify this. The data also has zones and rate codes that appear only during special events and holidays, so we remove these transactions since hourly rates and maximum allowable durations are relaxed in these transactions. We also remove a handful of transactions that have a length of stay of 0 minutes. We suspect this anomaly is related to a data-logging error. Finally, we consolidate several zones during the data preparation. For instance, a street can have parking on both sides of the street, and in some cases this is considered two distinct zones, so we combine these in our analysis. In total, we remove 4,285 transactions, and are left with 105,401 unique customers who generate 609,516 transactions, 99.3% of our original dataset, across 43 zones in our neighborhood.

While 3,291 customers are exposed to the pricing error (i.e., parked at least once in a treated zone after the price change was live and paid via the mobile app), the final data set

is reduced to 180 customers who we identify as regular users. These are individuals who were also parking in the treated zones before experiencing the price error in the app. Using each user's first exposure to the pricing error, we split their transactions as *BeforeExposure* and *AfterExposure*. As we describe in detail in Section 2.2.3, we remove the transaction where customers are exposed to a treatment and we note that transactions in non-treated zones after the pricing error was live are in the *BeforeExposure* set as the consumer had not yet been exposed to the pricing error at the time of these transactions. Overall, this group of 180 users amasses 6,477 transactions during the price change and 4,183 transaction prior to it. Since these were regularly parking in the area prior to the pricing error, we will use the *BeforeExposure* data set to model customers' *true* destination preferences, before any price change could have affected their choices. In our robustness checks, which we describe in Section 2.2.4.4, we consider the full set of 3,291 users, which includes new users to the system and users who were not regular users prior to the pricing change. We provide several descriptive statistics of the pre and post exposure datasets in Table 2.6.

2.2.3 Econometric Analysis

We index customers with i from the set of all customers $\mathcal{I}$, and use the set $\mathcal{T}_i$ to represent customer i's transactions, where $t \in \mathcal{T}_i$ is a single transaction. We define $\mathcal{Z}$ to be the set of zones in the catchment area. We use $j, k \in \mathcal{Z}$ to represent an arbitrary zone and let $d_{jk} \in \mathbb{R}_+$ be the distance between zones. We use $z_A, z_B \in \mathcal{Z}$ to refer to the two treated zones. Since these are adjacent to one another, we group them and simply consider treated versus non-treated zones, we define $\mathcal{Z}_T = \{z_A, z_B\} \subset \mathcal{Z}$ to represent the set, or pair, of treated zones in our analysis. We further define $\mathcal{Z}_{NT} = \mathcal{Z} \setminus \mathcal{Z}_T$ to represent the zones not affected by the price change.

Since we assume each parker learns about the pricing error when they first park in one of the treated zones, each parker has a different exposure date to the treatment. We define τ_i to be the transaction where parker i is exposed to the rate change and we refer to this

Table 2.6: Descriptive Statistics

		BeforeExposure	*AfterExposure*
Number of Users		180	180
Transactions		4183	6477
Transactions in Treated Zones		525	1061
Transactions per Customer	Mean	23.24	35.98
	St. Dev.	28.12	67.61
	Min	1	1
	Max	173	613
	Median	12.5	12.5
Transactions per Day per Customer	Mean	0.18	0.26
	St. Dev.	0.27	0.36
	Min	0.01	0.01
	Max	2.00	2.00
	Median	0.09	0.1
Length of Stay (Hours)	Mean	1.56	1.49
	St. Dev.	0.58	0.61
	Min	0.05	0.03
	Max	2.00	2.00
	Median	2.00	1.83

transaction as the "learning transaction." At τ_i, the parker does not know about the rate change, but upon completion, the parker observes the total cost is one third of what it should be and officially learns that this block is priced incorrectly. At this point we assume the user is aware of the pricing error and we group the subsequent transactions, i.e., $t > \tau_i$ in the *AfterExposure* dataset. Similarly, transactions prior to the learning transaction, i.e., $t < \tau_i$ are part of the *BeforeExposure* data set. In the remainder of the paper we use $p \in \{\textit{BeforeExposure}, \textit{AfterExposure}\}$ to refer to the two time periods. Figure 2 describes how we structure the data based on τ_i.

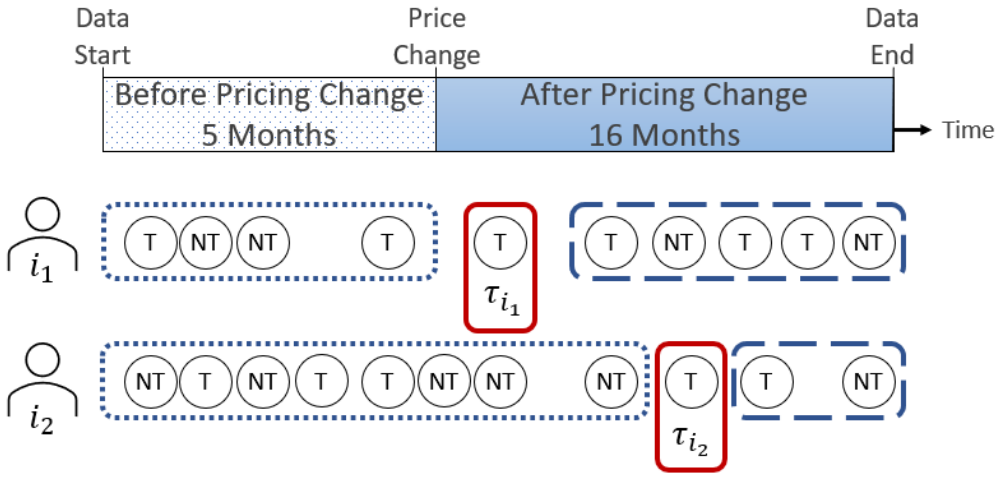
In the analysis, we drop the learning transaction from the data to account for any idiosyncratic effects the first time a user is exposed to the pricing error, assuming user will fully incorporate the learned information for their next transaction. This approach is also used in "regression discontinuity design studies" (Imbens and Lemieux, 2008). For instance, Aguiar and Waldfogel (2018) do it with when studying the impact of a song's popularity by comparing streams a full day before and a full day after inclusion on key playlist, and ignoring the 48 hours surrounding the inclusion time to eliminate time-of-day effects and to ensure a full day of streaming data has been accrued before doing their analysis.

2.2.3.1 Model-Free Support

Before introducing our model, we conduct several expository analyses to determine if there is any evidence in a behavioral change after the pricing change.

Change in Proportion of Transactions: Entire Catchment Area. In this analysis, we consider all of the zones in the entire catchment area. For each parker i, we compute $Y_{i,p} \in [0, 1]$, the proportion of parker i's transactions in the treated zones for both time periods $p \in \{BeforeExposure, AfterExposure\}$. Figure 2.8 contains a boxplot of both proportions side-by-side and we see that the mean and median proportion is greater post-exposure. To avoid bias, we restrict this analysis to only those users who have at least

Figure 2.7: Visual Depiction of How the *BeforeExposure* and *AfterExposure* Datasets are Created



Note: For two users, $i_1, i_2 \in \mathcal{I}$, the transaction history where circles with a T and NT to respectively represent transactions in treated and non-treated zones.

one transaction in $\mathcal{Z}_T$ both before and after τ_i. Without this additional criterion in this comparison, we have a glut of individuals with either $Y_{i,AfterExposure} > 0$ or $Y_{i,BeforeExposure} = 0$ because they never parked in a treated zone before their learning transaction. The shift in the treated zones proportion of transaction is significant (p-value $< 10^{-4}$) as suggested by a t-test for matched pairs and a Wilconxon signed-rank test. To further buttress this diagram, we estimate $Y_{i,p} = \alpha + \gamma \cdot \mathbb{I}[p = AfterExposure]$ using ordinary least squares. In Table 2.7, we present the results and note that the sign of γ is positive and significant, again suggesting that the pricing change could have played a part in driving more demand into the treated zones after the pricing mistake.

Change in Proportion of Transactions: Adjacent Zones. With a 0.25-mile radius neighborhood, the proportions $Y_{i,p}$ could include zones that are several city blocks away from the treated zones, which may be too far from the treated zones to include in the analysis since some customers may not be willing to walk that far for a cheaper rate. Rather than computing these proportions across the entire catchment area, we compute them for each

Figure 2.8: *BeforeExposure* and *AfterExposure*: Boxplot of Proportion of Transactions in Treated Zones.

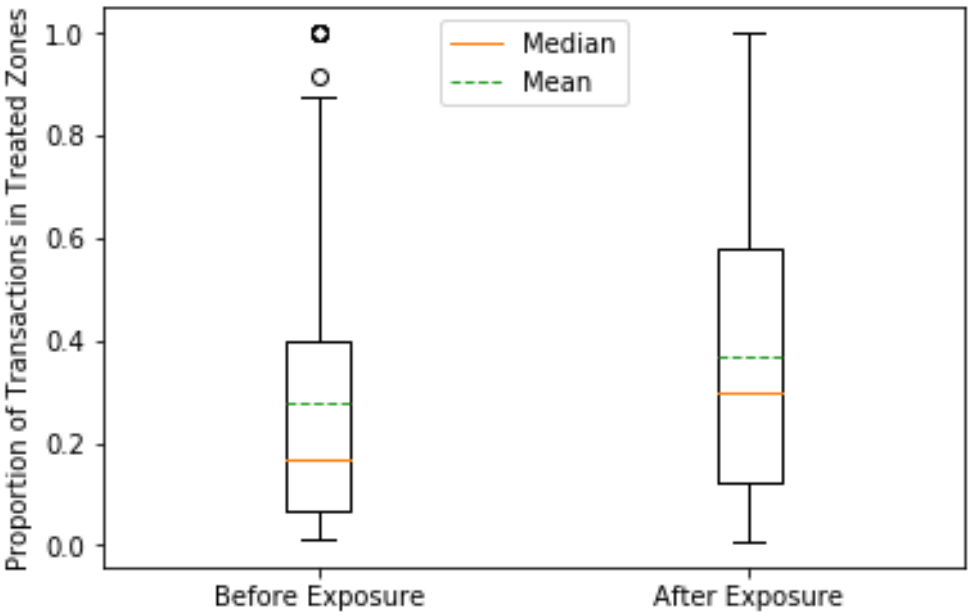
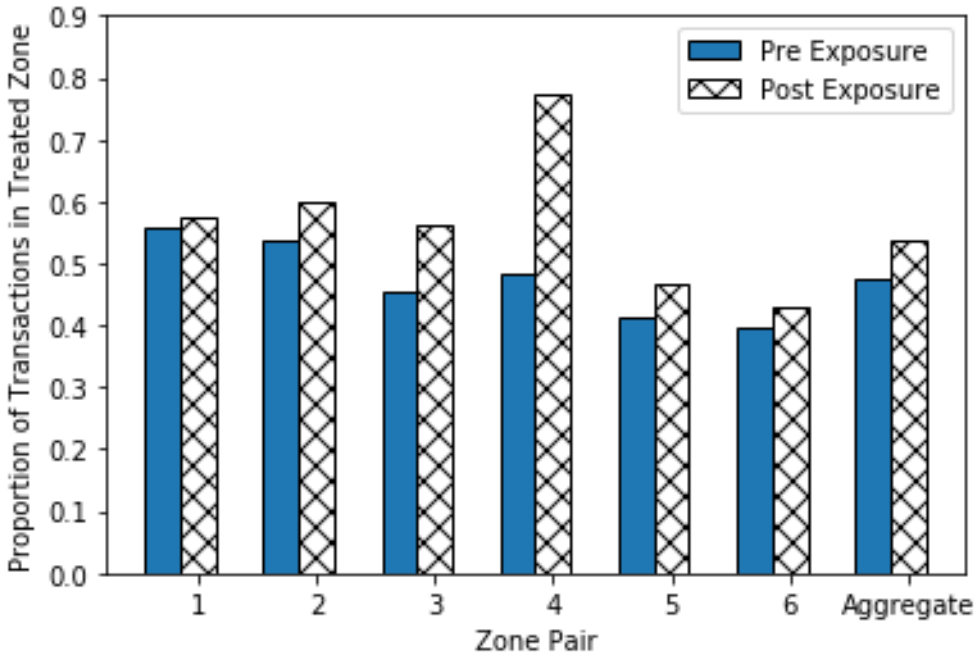


Table 2.7: Proportion of Transactions in Treated Zones

	$Y_{i,p}$
Intercept (α)	0.276***
	(0.026)
AfterExposure (γ)	0.095***
	(0.036)
Observations	246
R^2	0.027
F Statistic	6.798***
Note:	*p<0.1; **p<0.05; ***p<0.01

adjacent zone-pair combinations. In the map in Figure 2.6, we observe that there are 5 zones that are directly adjacent to the treated zones, with 3 next to z_A only, 1 next to z_B only, and 1 next to both. In total, this gives six zone pairs for comparison. In Figure 2.9, we plot the proportion of transactions in the treated zone for each of the six zone-pairs and when aggregated across all comparison pairs. For each comparison, the proportion is greater during the treatment period, but we note that the difference is only significant at the aggregate level.

Figure 2.9: Proportion of Transactions in Treated Zones (Only Using the Adjacent Non-Treated Zones)

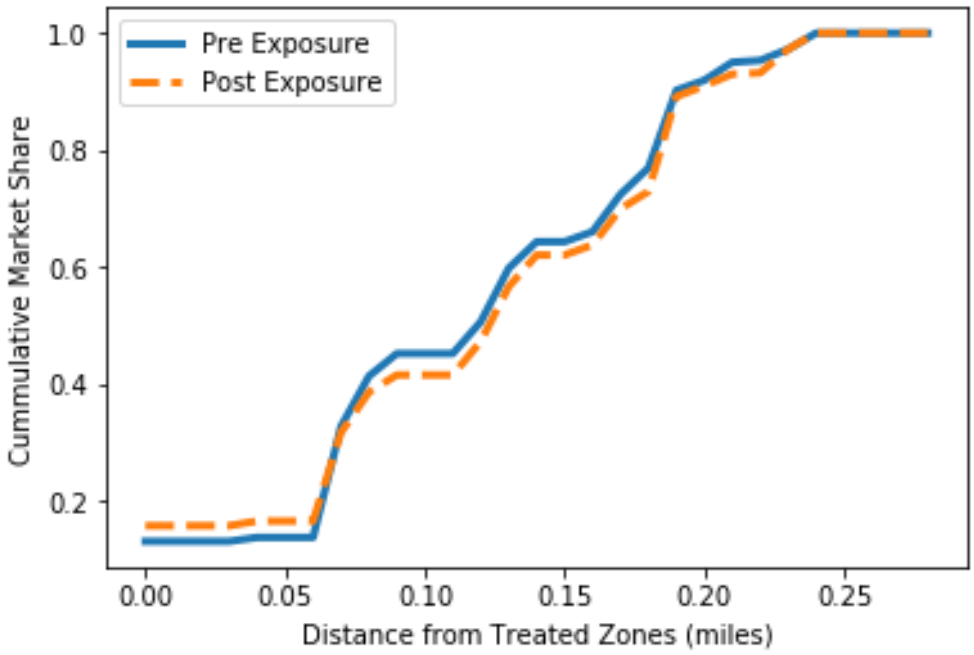


Note: Zone Pairs 1-4 correspond to z_A and Zone Pairs 5-6 correspond to z_B.

Change in Market Share. Finally, for the entire population we compute the share of all transactions in each zone $z \in \mathcal{Z}$, both before and after exposure. We denote this value $MarketShare_{z,p}$ and for each period we plot the cumulative proportion of transactions as a function of distance from the treated zones. While $Y_{i,p}$ represents the proportion of user i's

transactions in the treated zones during period p, $MarketShare_{z,p}$ represents the proportion, or percentage, of the population's transactions in zone z during period p. In Figure 2.10, the x-axis is the distance (d) from the treated zones, and the y-axis plots the cumulative share of transactions within d miles of Z_T. Mathematically, for each distance d on the x-axis, the y-axis plots $\sum_{z \in \mathcal{Z} | \min(d_{z_A,z}, d_{z_B,z}) \leq d} MarketShare_{p,z}$. We see that the post-exposure line lies above the pre-exposure line for small values of d, which captures the total market share for the treated zones and the zones that are closest to the treated zones. However, as the distance d around the treated zones increases, the post-exposure line falls below the pre-exposure line. This shift in market share from zones farther away from the treated zones before the pricing change into the treated zones after the pricing mistake is additional evidence in the population's behavioral change after learning of the pricing error.

Figure 2.10: Shift in Market Share Towards Treated Zones



2.2.3.2 Latent Class Multinomial Logit Model

In this section we describe the model we use to estimate the spatial elasticity while highlighting our underlying assumptions.

Assumption 2.2.1. *A customer's parking choice behavior is captured by the latent class multinomial logit (LC-MNL) model.*

First, we assume the latent class multinomial logit (LC-MNL) choice model governs parking decisions. Embedded in the LC-MNL is the multinomial logit (MNL) choice model, which is commonly used in many disciplines including economics, marketing, transportation, and revenue management. In the LC-MNL, a customer belongs to a particular class k with a probability captured by the mixing distribution. Given membership in a class k, choices for a particular product j occur according to the MNL model, which is based on the Random Utility Maximization (RUM) framework where consumers choose the product j that maximizes their utility. Under RUM theory, each product's utility can be divided into a random, stochastic component and a deterministic component. This deterministic component represents the utility gained from procuring product j, and is a weighted sum of the product features, such as price and quality. In the utility, the features are multiplied by parameters. Depending on the feature, the associated parameter can be either class-agnostic (i.e., the same parameter for all classes) or class-specific (i.e., the parameter varies by class) in the LC-MNL. We refer the reader to (Train, 2009) for a thorough description of both models. Extending the class-product framework to the parking context, class k is analogous to zone k being the desired, or intended, destination street block and product j corresponds to parking zone j being the actual chosen street block that a user selects.

In our setting, the discrete mixing distribution represents the probability that parking zone k is the intended destination zone, or the zone the customer wants to ultimately reach. Given that a customer's intended destination is zone k, we let $q_{jt|k}$ denote the conditional MNL choice probability that a customer chooses to park in zone j at time t. When we

uncondition $q_{jt|k}$ with a specific discrete mixing distribution, we arrive the expression for the LC-MNL probability any given customer parks in zone j, as described in Assumption 2.2.1.

Assumption 2.2.2. *The utility a customer gains during a parking transaction is solely a function of the total cost and the walking distance from their parking zone to their intended destination.*

With this structural form for the utility, we are imposing the simplest behavioral model for customers. We assume that they have full information of their length of stay and the rates in each zone, linearly trade off their transaction cost with walking distance, and value their utility in terms of dollars.

Initially, utility is only a function of two components: (1) the total service cost, $LOS_t \cdot rate_{jt}$, which is the length of stay (LOS) of transaction t multiplied by the hourly rate at zone j during transaction t, the zone where the customer actually parks, and (2) the distance between zone j and k, d_{jk}, where zone k is the customer's desired, yet unobserved, intended destination. We note that the hourly rate is static over time across all zones during the treatment period, so we could drop the index t and simply use $rate_j$, but we keep the transaction index as it is helpful for describing the extensions in Section 2.2.4.1. This yields the baseline utility function in Eq. (2.7), where the coefficient β is the spatial elasticity we will estimate and ϵ_{jkt} is a random noise term that captures unobserved features, which is assumed to be an i.i.d standard Gumbel random variable with mean zero.

$$v_{jt|k} = \beta \cdot d_{jk} - LOS_t \cdot rate_{jt} + \epsilon_{jkt} \tag{2.7}$$

Under this functional form, the maximum attainable utility from any transaction is 0, which occurs when the customer pays nothing and parks exactly at their intended destination (i.e., $rate_{jt} = 0$ and $d_{jk} = 0$). With $v_{jt|k}$, we can derive an expression for $q_{jt|k}$ in Eq. (2.8). We note note that we do not account for the no-purchase option in the choice probability $q_{jt|k}$ because our dataset only considers purchases. This follows classic choice model estimation

(Kamakura and Russell, 1989; Balachander and Ghose, 2003).

$$q_{jt|k} = \frac{\exp(v_{jt|k})}{\sum_{j' \in \mathcal{Z}} \exp(v_{j'kt})} \tag{2.8}$$

As mentioned before, in the traditional LC-MNL, parameters can be either class-specific (e.g., β_k) or take the same value across all classes. We do the latter and compute a single, population-wide spatial elasticity. In our setting, the equivalence to a product feature is the distance from the chosen zone j (the product) to the intended zone k, which obviously varies with the class. This is in contrast to the canonical LC-MNL model, where the features for each product j do not vary by the class k, which in our setting will be equivalent to having all zones equally spaced from each other. With this in mind, we can only estimate a single value for β for the entire population.

Since $q_{jt|k}$ is a conditional choice probability, conditioned on zone k being the customer's true, intended destination, we must uncondition the intended destination to get the full LC-MNL choice probability of a customer choosing to park in zone j described in Assumption 2.2.1. To uncondition, we define λ_k^i to be the probability zone k is the customer i's desired destination zone, where $\sum_k \lambda_k^i = 1$. In this sense, λ_k^i is our discrete mixing distribution over intended zones.

We note that the expression for $q_{jt|k}$ is not person-specific and generic for the entire population, but as we describe next in Assumption 2.2.3, λ_k^i is computed to be a person-specific mixing distribution.

Assumption 2.2.3. *Pre-exposure transactions capture each individual's true, intended destination zone preferences.*

Computing λ_k^i is challenging because each customer i's intended zone is unknown and unobserved in our dataset. Under Assumption 2.2.3, we consider the transactions from the *BeforeExposure* period to capture customers' true, intended destinations preferences. The rationale is that the entire system was static during this period—prices and the capacity

of each block did not change—so transactions should represent customers' true, intended, desired destinations if the blocks were not full at the time of the transaction.

To compute λ_k^i, we first determine counts n_{ik} and n_i, respectively the number of transactions customer i has during the *BeforeExposure* period in zone k and over the entire neighborhood. To ensure that we are computing the true preferences of intended zones, we only include transactions in the counts n_{ik} and n_i that occur when all zones in the catchment area were not at full capacity. While this may be conservative, it ensures that customers could have theoretically parked at any zone they wanted to at the time of their transaction–the included records better capture their true preferences.

With n_i and n_{ik}, we compute the population's probability for zone k being the intended destination zone. We denote this value λ_k and it is defined $\lambda_k = \frac{\sum_{i \in \mathcal{I}} n_{ik}}{\sum_{i \in \mathcal{I}} n_i}$. We use population-level distribution over intended zones as a prior belief on customer i's specific distribution, and update it using Bayesian smoothing. We refer the reader to Manning, Schütze, and Raghavan (2008) for a complete description of the mechanics of various smoothing techniques. To arrive at λ_k^i, the prior belief λ_k is given a pseudo-count of 1 and updated using n_{ik} and n_i. The expression for λ_k^i computed via Bayesian smoothing is in Eq. (2.9).

$$\lambda_k^i = \frac{\lambda_k + n_{ik}}{1 + n_i} \tag{2.9}$$

The advantage of using smoothing to compute λ_k^i instead of using the simple, counted-based, empirical distribution, i.e. $\lambda_k^i = \frac{n_{ik}}{n_i}$, is that it does not suffer from the "zero-probability problem" or "zero-frequency problem" (Witten and Bell, 1991). The count-based approach can be problematic when there are not any pre-treatment transactions in a particular zone k (i.e., $n_{ik} = 0$), then $\lambda_k^i = 0$. We acknowledge that the count-based distribution is an intuitive, natural way of thinking about the distribution over intended zones, so in our robustness checks, which we describe in Section 2.2.4.4, we also include a series of estimates using the count-based distribution.

We note that the typical LC-MNL implementation considers customer-agnostic mixing probabilities and assumes a parametric representation for them, and the parameters are jointly estimated in the MLE step. In contrast to the typical LC-MNL, we use customer-specific probabilities λ_k^i, and estimate their values using *BeforeExposure* data. The estimates are then used as exogenous parameters in the MLE procedure. The motivation behind this approach is the fact that these probabilities do not represent membership to a class in our setting, but rather probabilistic preferences over destinations. Moreover, we note that this two step approach will return accurate estimates because the estimation of λ_k^i and the MLE procedure use different data. The λ_k^i distributions are computed using the *BeforeExposure* data and the MLE uses the *AfterExposure* data. In our setting, if λ_k^i is estimated simultaneously with β during the MLE procedure, i.e. on the same dataset, the estimates of λ_k^i and β can be uninformative and undefined. More generally, the following Lemma 2.2.1 shows that, in our setting, the traditional LC-MNL approach of estimating both the mixing distribution and the parameters in the utility function can result in undefined estimates. For readability, the proof is available in the Appendix.

Lemma 2.2.1. *Estimating λ_k^i and β jointly via Maximum Likelihood Estimation can result in uninformative estimates, i.e., $\beta \to -\infty$ and λ_k^i equal to user i's exact proportion of transactions in zone k in the estimation dataset.*

The observation in Lemma 2.2.1 highlights the necessity of exogenously providing values of λ_k^i to the MLE estimation. Note that the behavior as $\beta \to -\infty$ is unique to our setting since distance, the key feature in our utility function, varies with both the class (or intended zone) k and the product (or actual parking zone chosen) j. This is contrast to the canonical LC-MNL as mentioned in the explanation of Assumption 2.2.2.

Finally, combining the expressions for λ_k^i and $q_{jt|k}$, we can derive an expression for P_{jt}^i, the customer-specific, LC-MNL probability that customer i chooses to park in zone j during occasion t, which we define in Eq. (2.10). This expression is a key input in the estimation

procedure which we describe next.

$$P_{jt}^{i} = \sum_{k \in \mathcal{Z}} q_{jt|k} \cdot \lambda_{k}^{i} \tag{2.10}$$

2.2.3.3 Estimation

To estimate the spatial elasticity, we do maximum likelihood estimation (MLE) using the *AfterExposure* data. Building on the notation in the previous section, we respectively define $\mathcal{L}(\beta)$ and $l(\beta)$, the likelihood and log-likelihood functions, in Eqs. (2.11) and (2.12), where the binary indicator S_{jt}^{i} takes the value 1 if customer i parked in zone j during transaction t.

$$\mathcal{L}(\beta) = \Pi_{i \in \mathcal{I}} \Pi_{t \in \mathcal{T}_i} \Pi_{j \in \mathcal{Z}} \Big(\sum_{k \in \mathcal{Z}} \lambda_{k}^{i} \cdot q_{jt|k}(\beta) \Big)^{S_{jt}^{i}} \tag{2.11}$$

$$l(\beta) = \ln(\mathcal{L}(\beta)) = \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}_i} \sum_{j \in \mathcal{Z}} S_{jt}^{i} \cdot \ln \Big(\sum_{k \in \mathcal{Z}} \lambda_{k}^{i} \cdot q_{jt|k}(\beta) \Big) \tag{2.12}$$

Since the log-likelihood function, is not concave in its parameters, the MLE can converge to a local maxima depending on the initial point supplied to the optimization procedure. To assuage this, we run the maximization procedure with many different starting points and only retain the point estimate that returns the maximum log-likelihood value. To generate standard errors, we use the bootstrapping procedure described in Efron and Tibshirani (1986) with 50 resampling iterations.

Finally, we note that the LC-MNL, when features of products (or actual zones) do not vary by class (or intended zones), is readily estimated in off-the-shelf statistical software (Sarrias and Daziano, 2017; Croissant, 2020; Beath, 2017), but as mentioned previously, in our setting a critical feature of each "product" or parking zone is the distance to the intended zone, which is directly defined by the "class". Because of this, we cannot use any pre-existing packages and must develop a custom estimation procedure, which is available upon request.

2.2.4 Results

Using the baseline utility model, defined in Eq. (2.7) (also included in Table 2.8), we present our estimation results in Table 2.9. Like we suspect, the sign of β is negative, indicating that as the distance between the customer's intended and actual destinations increases, the utility of the choice decreases. Ultimately, the results show that spatial elasticity is \$80.84 per mile.

Table 2.8: Utility Specifications

Specification	Parameters	$v_{jt\|k}$
Baseline	β	$\beta \cdot d_{jk} - LOS_t \cdot rate_{jt}$
Rain	$\beta; \beta_{rain}$	$(\beta + \beta_{rain} \cdot rain_t) \cdot d_{jk} - LOS_t \cdot rate_{jt}$
AM	$\beta; \beta_{AM}$	$(\beta + \beta_{AM} \cdot AM_t) \cdot d_{jk} - LOS_t \cdot rate_{jt}$
LOS-II	$\beta; \beta_{cost}$	$\beta \cdot d_{jk} + \beta_{cost} \cdot LOS_t \cdot rate_{jt}$
LOS-III	$\beta; \beta_{rate}$	$\beta \cdot d_{jk} + \beta_{rate} \cdot rate_{jt} - LOS_t \cdot rate_{jt}$
LOS-IV	β; β_{rate}; β_{cost}	$\beta \cdot d_{jk} + \beta_{rate} \cdot rate_{jt} + \beta_{cost} \cdot LOS_t \cdot rate_{jt}$

2.2.4.1 Extensions

Next, we make several modifications to the baseline utility specification in Eq. (2.7) to explore the impact of rain and morning hours on the elasticity estimate. We present the utility expression for both of these alongside the baseline utility specification in Table 2.8.

Impact of Precipitation. The first extension deals with the idea that people do not enjoy walking in the rain, so spatial elasticity should be stronger when precipitation is present. We gather rain data using `weather.com`'s historical data API which logs all real-time, weather statistics at approximately one hour intervals from the nearest weather station to our catchment area, which is less than 6 miles away from the treated zones. In the worst

Table 2.9: Regression Results: Baseline, Rain, and Morning

	Utility Specification					
	Baseline	Rain Effect			Morning Effect	
		(a)	(b)	(c)	(d)	(e)
		± 30 Min.	± 60 Min.	Day	60 Min.	120 Min.
β	-80.84^{***}	-80.34^{***}	-80.79^{***}	-81.14^{***}	-78.65^{***}	-79.31^{***}
	(6.47)	(5.21)	(6.11)	(7.66)	(6.19)	(5.98)
β_{rain}		-11.26^{***}	-0.75	-5.94^{***}		
		(3.89)	(1.74)	(1.94)		
β_{AM}					-31.28^{***}	-12.70^{***}
					(4.54)	(3.04)
Num PIDS	180	180	180	180	180	180
Num Txns	6477	6477	6477	6477	6477	6477
Log Likelihood	-19132.6	-19131.2	-19132.6	-19042.5	-19014.4	-19035.4
AIC	38267.3	38266.4	38269.2	38089	38032.9	38074.8
BIC	38274	38280	38282.8	38102.5	38046.4	38088.4
Note:						*p<0.1; **p<0.05; ***p<0.01

case, the time between weather log entries can be separated by slightly over 2 hours. We note that light snow is included as precipitation in the weather data. We use the binary variable $rain_t$ to represent if precipitation was present during transaction t. If $rain_t = 1$, then rain is present when person i transacts during occasion t and the spatial elasticity is accordingly adjusted by β_{rain}, which has the same units as β, dollars per mile. Most of the time, $rain_t$ refers to rain, but it can also refer to snow. We note that days with heavy snowstorms are removed from our data as the city usually closes during these days and driving is deemed dangerous. We consider rain in three cases: (a) if precipitation occurs (a) 30 minutes or (b) 60 minutes before or after of the transaction; and (c) if precipitation is logged at any time on the day of the transaction. For transactions without data within 30 or 60 minutes of the starting timestamp, we conservatively presume that there was not any precipitation present.

Impact of Morning. The second extension involves the "morning rush hour" phenomenon. In particular, we hypothesize that during the morning hours, spatial elasticity should be larger since people usually have appointments and meetings first thing in the morning, and if they are running late then walking distance is something they would try to minimize. In the same way we define $rain_t$, we define a binary variable AM_t, which takes the value 1 if transaction t occurs during the morning time frame. In transactions where $AM_t = 1$, the spatial elasticity is adjusted by β_{AM}. We consider the morning period with two morning windows: the first (d) 60 minutes and (e) 120 minutes of the day.

Discussion. The estimates of the rain and morning extensions are presented in Table 2.9. Precipitation being present increases the magnitude of spatial elasticity. This is to be expected and aligns with the intuition that people do not like to walk in the rain, especially in a metropolitan area where they are likely headed into an office, store, or restaurant. However, at the day-level (c) the effect is mild, as it only increases spatial elasticity by \$5.94 per mile, or 7.7%. We believe that the day-level modeling of rain may not be granular enough to capture impact of precipitation. Rain is likely to disrupt users experience only when present

around the parking transaction. So, we also consider situations where rain is present within 30 minutes of a transaction (a), the magnitude increases by $11.26 per mile, or about 13% over the baseline estimate and is significant. Note that the effect size doubled compared to the day-level estimate (c). And when we increase the window around the transaction to be 60 minutes (b), the effect weakens and is statistically insignificant. As the length of time for measuring the presence of rain widens, we lose detail and granularity, and the effect attenuates. With a 60-minute window on either side of the transaction, it is possible that rain could dry up or stop with such a large window, in which case the rain effect vanishes. However, with a 30-minute window, rain is more likely to be present at the transaction time, so the effect is stronger.

When we consider the impact of the morning hours, the estimates confirm the intuition that people are usually busy in the mornings, and therefore less willing to walk. When we consider the morning to only be the first hour of the day (d), the magnitude increases by $31.28 per mile, or 36%. Like the precipitation time frame, when we increase the morning window to two hours (e), this early morning impact dissipates rapidly. The raw data reveals that the duration of transactions that begin in the morning tend to be longer than transactions that begin in non-morning hours. These observations are combined in Table 2.10 where we show the mean length of stay of transactions that occur in morning and non-morning hours. We report the means for when the morning window definition is the first 60 minutes and the 120 minutes, and for both the *BeforeExposure* and *AfterExposure* datasets. The differences in duration across morning and non-morning are significant in an unpaired, two-sample t-test.

If we assume that consumers interpret of the cost of a transaction as two separate costs – the fixed cost, which is a function of the walking distance, and the variable cost, which is a function of the length of stay – then a longer length of stay in the morning means the variable cost component is greater in the morning. Because the hourly rate is multiplied by the length of stay, the impact of a cheaper rate is felt more in the morning when the duration

Table 2.10: Mean Length of Stay in the Morning Hours

	AM Window: 60 Minutes		AM Window: 120 Minutes	
Time Frame	Morning	Non-Morning	Morning	Non-Morning
BeforeExposure	1.53	1.48	1.57	1.47
AfterExposure	1.59	1.49	1.62	1.47

of transactions is longer. This increased variable cost could also be driving the larger spatial elasticity in the morning, (d) and (e) in Table 2.9, since there are larger potential savings available.

2.2.4.2 Relaxing Behavioral Assumptions

In our baseline structural form, we assume the decision maker has perfect foresight on their duration and all prices in the catchment area. We impose this by using a unit coefficient in front of the transaction cost term $LOS_t \cdot rate_{jt}$. However, it could be the case that customers value length of stay and rates differently than assumed, in which case the units of utility would not be in dollars. In light of this, we test three additional structural utility forms to model how users value transaction cost (length of stay and rate) in their parking choice. We denote these LOS-II, LOS-III, and LOS-IV and present these in the last three rows of Table 2.8.

In LOS-II, we add a coefficient β_{cost} on $LOS_t \cdot rate_{jt}$. This coefficient normalizes the utility expression to achieve a better estimate of β. We note that LOS-II subsumes the baseline utility specification, and we recover the baseline utility from LOS-II if we constrain β_{cost} to -1 in LOS-II. In LOS-III, we add the term $\beta_{rate} \cdot rate_{jt}$ to the baseline utility expression. This allows us to measure if the rate alone causes any change in behavior. We note that since this functional form still has the term $-LOS_t \cdot rate_{jt}$ and builds on the baseline specification, we can interpret the units of utility in dollars. Finally, in LOS-IV, we combine both of the

changes incorporated into LOS-II and LOS-III.

Table 2.11: Regression Results: Baseline, LOS-II, LOS-III, LOS-IV

	Utility Specification			
	Baseline	LOS-II	LOS-III	LOS-IV
β	-80.84^{***}	-39.41^{***}	-44.13^{***}	-40.26^{***}
	(6.47)	(3.13)	(3.06)	(3.27)
β_{rate}			1.30^{***}	0.21^{**}
			(0.03)	(0.09)
β_{LOS}		-0.15^{***}		-0.27^{***}
		(0.02)		(0.04)
Num PIDS	180	180	180	180
Num Txns	6477	6477	6477	6477
Log Likelihood	-19132.6	-18125.8	-18235.6	-18093.2
AIC	38267.3	36255.5	36475.3	36192.3
BIC	38274	36269.1	36488.9	36212.7
Note:				*p<0.1; **p<0.05; ***p<0.01

We present the new estimates, next to the baseline, in Table 2.11. Since we interpret utility in terms of dollars in the baseline specification, the units of spatial elasticity are dollars per mile. However, under LOS-II, LOS-III, and LOS-IV, the spatial elasticity cannot be interpreted with these units. Instead, we compare the ratio of $\frac{\beta}{\beta_{cost}}$. Under LOS-II, this ratio is 262.73, and under the baseline utility, this ratio is 80.84. This difference indicates that customers appear to value distance (or time to get) to their desired destination more than the total transaction cost. We also note that $\beta_{cost} < 0$, indicating that as the total transaction cost of transacting at zone j increases, the utility gained from choosing zone j decreases.

In LOS-III and LOS-IV, we see a similar relationship in the ratio $\frac{\beta}{\beta_{cost}}$. This suggests that

the baseline utility formulation may be overemphasizing the transaction cost, and distance is a relatively bigger driver of customer choice. One plausible explanation is that parking preferences are inelastic and last-mile/walked distance may be a more important factor in urban transportation settings. In Section 2.2.5 we demonstrate the importance of spatial elasticity by evaluating two pricing policies, one that considers spatial elasticity and one that does not, in the presence of spatially sensitive customers.

2.2.4.3 Partial Effects

Because the LC-MNL model is highly non-linear, our estimates only provide insight on how utility changes as a function of distance, length of stay, rate, and total transaction cost. However, it is unclear how these factors truly interact and ultimately affect individual customer choice. To more thoroughly understand this relationship, we consider the structural form in LOS-III, where the units of utility can be interpreted in dollars, and compute the partial effect to study how $q_{j|k}$, the conditional choice probabilities, change with length of stay and distance between zones.

We assume that the intended destination zone k is known, and we compute two conditional partial effects for selecting zone j: one with respect to length of stay and one with respect to distance between zones j and k. Since we are conditioning on the intended zone k, we refer to these as conditional partial or marginal effects. We provide the expressions in Eq. (2.13)-(2.14), which are obtained following Section 15.9 of Wooldridge (2010). Note that when computing partial effects, it is common practice to take the average partial effect over all individuals. But since $q_{j|k}$ does not vary with customer i, the average partial effect is not informative, so we do not compute it here. For notational brevity, we drop the index

t in this section.

$$\frac{\partial q_{j|k}}{\partial LOS} = \frac{-rate_j \cdot \exp(v_{j|k})\big(\sum_{j'} \exp(v_{j'|k})\big) - \exp(v_{j|k}) \cdot \big(\sum_{j'} -rate_{j'} \cdot \exp(v_{j'|k})\big)}{\big(\sum_{j'} \exp(v_{j'|k})\big)^2} \tag{2.13}$$

$$\frac{\partial q_{j|k}}{\partial d_{jk}} = \frac{\beta \cdot \exp(v_{j|k})\big(\sum_{j'} \exp(v_{j'|k})\big) - \beta \cdot \exp(v_{j|k})^2}{\big(\sum_{j'} \exp(v_{j'|k})\big)^2} \tag{2.14}$$

We illustrate the partial effects in Figure 2.11. To measure the conditional marginal effects we do the following: we consider the actual destination zone j to be the treated zone (i.e., $j = z_A$). Then, for a fixed LOS, we use the pricing-error rates, and calculate the partial effect for each intended destination zone k, in the order from the closest intended destination to the farthest possible destination from j. We do this for four different lengths of stay and respectively plot the effects. In this exercise, we only include destination zones k that are priced at \$3.75/hour in this exercise and ignore zones that are priced at \$1.25/hour. We do this because the conditional marginal effect is effectively 0 when the intended destination is priced the same as the treated zone.

In Figure 2.11a, we see that as the distance between the intended zone and the treated zone increases, the LOS marginal effect on choosing the treated zone (Eq. (2.13)) decreases, but it is always positive. We also note that when the duration of the transaction increases, the LOS marginal effect is more pronounced. In Figure 2.11b, we see that the distance between the actual and intended zones has a strong negative impact on choice, but weakens as the distance increases. As expected, this effect is stronger for longer LOSs.

One interpretation of these plots is that customers can be segmented into short-stay and long-stay users. Due to the cheaper rate at the treated zone j, all users gain utility if they park in j instead of k. However, the transaction cost for long-stay users is higher than it is for short-stay users, so the total savings of the cheaper rate at the treated zone j are greater – long-stay users gain more utility for parking in a treated zone. In other words, the savings for short-stay users are simply not significant enough to compensate for added cost

of walking out of their way for the cheaper rate, so they are less sensitive. In practice, if the system operator has full knowledge of a customer's duration and intended destination before the transaction, they can tailor personalized discounts. In particular, users with longer LOS and closer intended destinations are more sensitive to discounted rates.

2.2.4.4 Robustness Checks

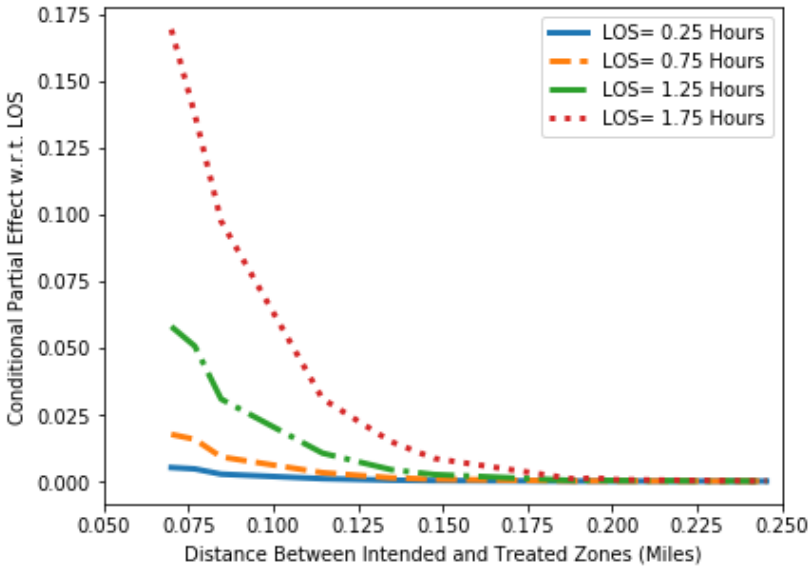
To ensure that our assumptions and data-processing methods are sound, we complete several robustness checks. We observe that, in general, our results and insights are preserved across the various scenarios considered, but the magnitudes of the estimates do change. The detailed results to these are available in the Online Appendix.

- Alternative Mixing Distributions (Tables C.4-C.5): Instead of computing the mixing distribution λ_k^i via Bayesian smoothing and exogenously providing this to the MLE procedure, we consider two other distributions over intended zones.

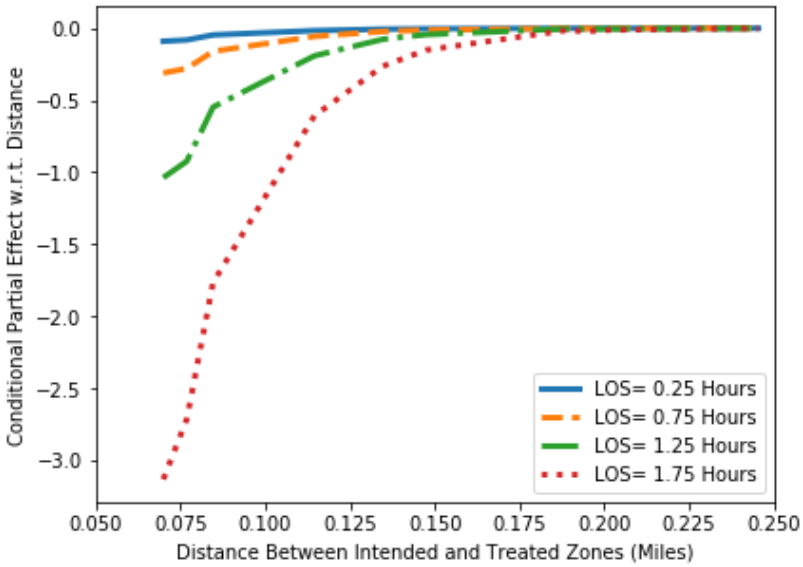
  In the first, we use the count-based, empirical distribution over intended zones, i.e. $\lambda_k^i = n_{ik}/n_i$. The key difference between the two approaches is that the empirical, count-based distribution can be quite sparse. Specifically, many zones k can have $\lambda_k^i = 0$ because a user i could have no pre-exposure transactions in zone k. As a result of this change the spatial elasticity weakens from \$80.84 per mile to \$51.76 per mile.

  In the second, we retain the learning transaction. In Section 2.2.3, we describe how we remove the "learning transaction," for each user and do not include it in either the *BeforeExposure* and *AfterExposure* datasets. Since we use the *BeforeExposure* to compute the intended destination distribution λ_k^i, not including the learning transaction can be wasteful, especially since the count-based, empirical λ_k^i are quite sparse. In this exercise, we include the learning transaction in the *BeforeExposure* dataset and then compute λ_k^i via Bayesian Smoothing. This shifts more mass to either λ_{i,z_A} or λ_{i,z_B}, depending on which zone customer i's learning transaction occurs in, and shifts mass

Figure 2.11: Conditional Partial Effect with Respect to (a) Length of Stay and (b) Distance at Treated Zone #1



(a) Length of Stay Conditional Partial Effect



(b) Distance Conditional Partial Effect

away from all other zones. The implication of this is that the mixing distribution reflects a stronger affinity for treated zones. As such, we would expect that users would be more sensitive to distance after learning about the pricing error, which is exactly what we see in the results. The spatial elasticity moves from \$80.84 per mile to \$87.12 per mile, meaning that users require a larger cost savings to walk the same distance to their intended destination.

- Neighborhood Size (Tables C.6-C.8): We vary the neighborhood size to include 0.10, 0.15, and 0.20-mile radius around the treated zones. When the catchment area radius is respectively 0.20 miles and 0.15 miles, the spatial elasticity estimates weakens by 12.7% and 3.5%, but when the radius is 0.10 miles, the estimate does not change from the baseline. Overall, the estimates that include rain seem unreliable as the catchment region shrinks, as the sign of β_{rain} is not consistent across day, $\pm$ 30 minutes, and $\pm$ 60 minutes. This is direct consequence of having less transactions affected by rain, which reduces the estimate's power.

- Distance Norm (Tables C.9-C.10): In addition to using Haversine distance for measuring the separation between zones (d_{jk}), we use Manhattan distance and the walking distance, which we query using the Google's Direction and Distance Matrix APIs. These distances are both longer than the Haversine distance. For both norms, the magnitude of the spatial elasticity estimates across all utility specifications decreases by approximately 20%. Since these measures likely reflect more accurate walking distances over the Haversine distances, one could claim that these estimates are more realistic estimates of spatial elasticity.

- User Inclusion Criteria (Table C.11): We perform the estimation with a second, less restrictive subset of the population. This group includes all individuals who were exposed to the pricing error. The main difference with the original cohort is that these users may not have any transactions prior to the pricing error or not have any

transaction in the treated zones. In total, there are 2,003 parkers that satisfy this criteria. For these individuals, we use the population's distribution over intended zones, i.e., λ_k. We find that spatial elasticity drops from approximately \$80/hour to \$64/hour when we relax the inclusion criteria. Moreover, the effect of rain vanishes but the morning-hour estimates hold. It is possible that assigning the population's distribution to individuals without pre-treatment data is not effective because there is no reason to believe that the distribution of the whole population's preferences align with an individual parker's.

- Extended Service Sessions (Table C.12): We notice that there are instances when users have multiple, consecutive transactions on the same zone. For instance, when the maximum stay is 2 hours, a customer could pay for 2 hours, and then when their time expires, they immediately can buy x hours of additional time. This appears as two separate transactions in the data, where the first transaction is 2 hours and the second is x hours. While this is illegal and users who engage in this behavior can receive parking citations, it is quite common in the data. We refer to this situation as a "re-up" and note that one could collapse a re-up chain into a single transaction with a duration of $2 + x$ hours. In this check, we identify a re-up when the end and start of two consecutive transactions occurred within 30 minutes of each other. Before any data processing, the dataset has over 609,000 transactions in the catchment area, and of these, there over 69,000 re-up chains. After collapsing the transactions in a chain into a single transaction and updating the duration and cost to reflect the combined values, over 89,000 transactions are removed. The resulting dataset has approximately 519,000 transactions, and this is what we use to repeat all of our estimate exercises outlined in this paper beginning in Section 2.2.2. Note that we recompute λ_k^i before completing the estimation. We find that the magnitude of the estimates increases by 10%.

  Similar to the case of β_{rain} estimates in the previous exercise with smaller neighborhood

sizes, the sign of β_{rain} here also lacks consistency between the day, $\pm$ 30 minutes, and $\pm$ 60 minutes. Since collapsing re-ups results in less transactions, we also have fewer transactions that are impacted by rain, so the power for estimating β_{rain} decreases.

- Alternative Utility Specification (Table C.13): We also consider a slightly simpler utility model that only depends on the rate $rate_{jt}$ but not on the length of stay LOS_t– i.e., it only considers the rate, not the total transaction cost. We find that the estimates do not vary much and our insights hold.

2.2.5 Numerical Study

In Section 2.2.4.2, our results indicate that customers are more spatially sensitive than our baseline, perfect foresight model indicates. Ultimately, this suggests that price is less important than distance in making parking decisions. In transportation settings with competition, multi-homing across platforms, and a dearth of brand loyalty, such as with dockless electric scooters, this spatial elasticity is an important consideration that should influence how systems manage and price their network. In this section, we unpack this and demonstrate the benefit of spatially-aware pricing policies.

To highlight the importance of spatial elasticity in this pricing context, we perform a numerical study where we evaluate two pricing policies, one that considers spatial elasticity and one that does not, in a setting where customers are spatially sensitive. Since parking spots are a reusable resource, we leverage the existing literature on pricing reusable products, such as Rusmevichientong, Sumida, and Topaloglu (2020); Lei and Jasin (2020), and Besbes, Elmachtoub, and Sun (2021). We compute the optimal, time-varying policy using the methodology proposed by Owen and Simchi-Levi (2018). Under this policy, in each time period t, arriving customers are presented with one of several pricing configurations over zones with some probability. Since a stochastic pricing policy does not make sense in a parking setting, since all customers in time period t should see the same prices, we convert

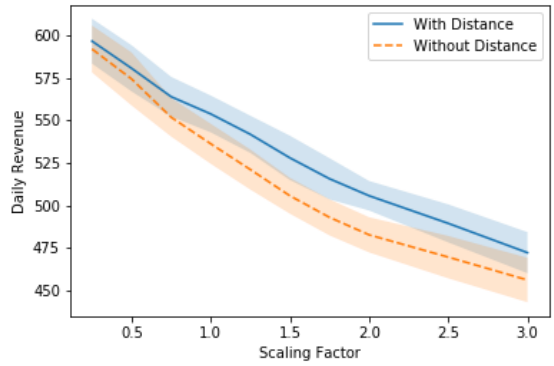
this policy to a deterministic policy. In each time period t, we offer all arriving customers the pricing configuration that occurs with the highest probability in that time period.

We consider a simple system with 3 parking zones, i.e. $\mathcal{Z} = \{A, B, C\}$, and a capacity of 10 spaces in each zone. We choose the optimal prices from the discrete set [\$0.25, \$0.50, ..., \$10.00] for each time period over a finite horizon of 10 hours, where the time is discretized into 30-minute periods. This gives a total of 20 time windows in the horizon, and we use t to represent an arbitrary time block. We set the distance between each zone as follows: $d_{AB} = 0.20, d_{AC} = 0.25, d_{BC} = 0.10$. Length of stay is independent of t and exponentially distributed with a mean of 1.5 hours. Time-varying Poisson arrivals are randomly generated for each zone and period t as follows: $\Lambda_{A,t} = 0.5 + \mathcal{U}(0, 1)$ per hour, $\Lambda_{B,t} = 4 + \mathcal{U}(0, 2)$ per hour, $\Lambda_{C,t} = 9 + \mathcal{U}(0, 2)$ per hour, where $\mathcal{U}(0, x)$ represents the uniform random variable between 0 and x. Due to the choices of distances, we use a spatial elasticity of \$5 per mile and our baseline utility specification in the first row of Table 2.8.

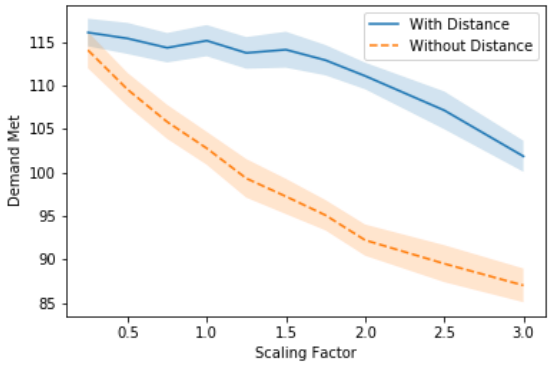
To measure how the policies perform in different settings, we generate a variety of system configurations by scaling either the arrival rates or the distance upwards or downwards. Then, for each configuration, we create 30 instances by randomly generating $\Lambda_{A,t}, \Lambda_{B,t}$, and $\Lambda_{C,t}$. For each instance, we compute two optimal policies: *WithDistance*, which incorporates spatial elasticity (i.e., $\beta > 0$), and *WithoutDistance*, which incorrectly assumes customers are not spatially sensitive (i.e., $\beta = 0$). For each instance, we simulate the system for 30 days, using the same stream of spatially-sensitive arrivals to evaluate each policy. We track the unmet demand, revenue generated, and several other key metrics. Experiments are run in Python 2.7 on a Dell Inspiron 13 with 16GB of RAM and an Intel Core i7 1.8GHz processor. We present the results in Figure 2.12, where the line and shaded area respectively represent the average and standard deviation over 30 random instances. As we scale the distance and arrival rate upwards from the base scenario, the spatially-aware policy generates more revenue and meets more demand. When we scale the distance and arrival rate down, the two policies perform similarly. Ultimately, this exercise shows that when customers are spatially

sensitive, then using a pricing policy that captures this behavior, i.e. the *WithDistance* policy, is beneficial. For cities with large demand, ignoring spatial elasticity can significantly reduce a system's performance. The results also indicate that regardless of how close or far spread out a city may be, incorporating spatial elasticity can improve key operational metrics.

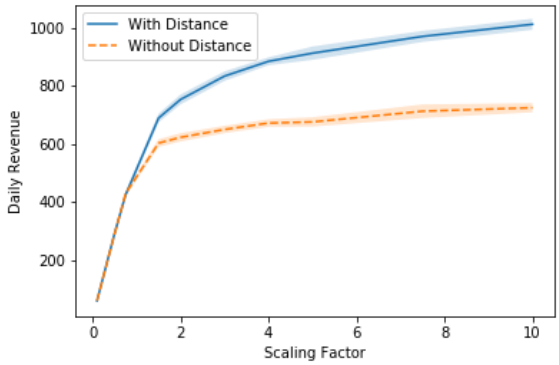
Figure 2.12: Simulation Results from Dynamic Pricing With and Without Spatial Elasticity



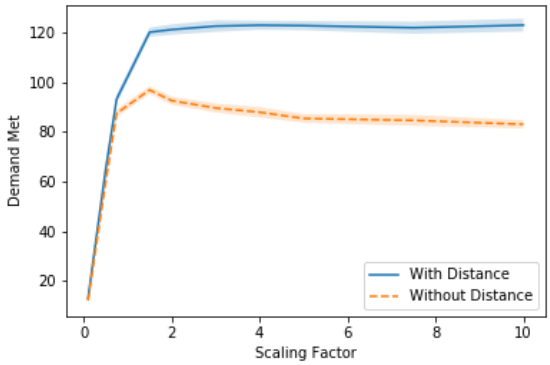
(a) Daily Revenue vs. Scaled Distance



(b) Demand Met vs. Scaled Distance



(c) Daily Revenue vs. Scaled Arrival Rate



(d) Demand Met vs. Scaled Arrival Rate

2.2.6 Conclusion and Future Work

Using nearly two years of transactions from a large metropolitan city's mobile parking application, we measure spatial elasticity, a trade-off between walking distance and cost. Normally, estimating this would require a large-scale pricing experiment, but during this period there was a pricing error on two, adjacent, downtown blocks, giving us the pricing variation that we can do our estimation.

In a parking-specific context, spatial elasticity is critical for cities that want to develop dynamic, traffic-minimizing, pricing policies for on-street, metered parking. Understanding spatial elasticity can allow city planners and policy makers to set prices in a way that incentivizes users to park on less-congested blocks that are away from popular areas. And while our data comes from a parking setting, these estimates can be insightful in many urban mobility settings. For instance, privately-held, venture-backed micro-mobility systems and ride-sharing platforms need to understand how users value walking distance if they want to encourage users to start and end their journeys at locations that might be a short distance away from their users' actual origins and intended destinations. Beyond mobility and transportation, these estimates can be used in other settings where walking is a mode of transportation, such as metropolitan facility location problems where the goal is to optimally place schools and transit centers.

Our empirical approach involves using the LC-MNL discrete choice model to capture customer's parking behavior. We use data before users were exposed to the pricing error to estimate a discrete distribution over customers' true destination preferences. In the traditional LC-MNL context, these intended destination preferences are incorporated into the model as class probabilities, then we use the transaction data after users were exposed to the pricing error to measure the spatial elasticty using maximum likelihood estimation. We add in additional factors, such as if rain was present at the time of the transaction or if the transaction was during the morning hours, and we find that both of these increase the

spatial elasticity.

One area for future work is to test if our estimates are robust to the initial transportation mode. In our setting, customers travel in their personal vehicle, but if the initial transportation mode is a bus, a shared vehicle (i.e., bike or electric scooter), or ride share, then it is possible that the spatial elasticities can be more or less sensitive. Another interesting direction involves a thorough examination of demographic and time factors that affect spatial elasticity, such as gender and time of day. For instance, during dusk or night time hours, when more crime occurs, spatial elasticity may be higher.

One important question that we did not explore deals with the counter-factual scenario after the pricing error was corrected. We did not have access to post-treatment data to measure if customers revert to pre-treatment behavior after the pricing change is correct. If customers do, that would give stronger veracity to our estimates.

It is important to acknowledge that the estimates we report are unique to the features of our city and demographics of the population. For instance, our data comes from a neighborhood in one of the most highly educated cities in America and with one of the highest average salaries in the country. Moreover, this specific vicinity is home to high-end boutiques and restaurants. Because of these characteristics, we suspect that our estimates are quite high since many residents and visitors are affluent and busy young professionals, so they are more time sensitive. We hypothesize that the estimates will likely change depending on the characteristics, realities, features of each city. For instance, in areas surrounding college campuses, we would expect the estimates to be much lower as students are much more price sensitive and would be much more likely to change their behavior in response to price discounts.

Finally, we use pre-treatment data to create a distribution over each customer's intended destination, but it may be possible to gather better estimates. For instance, Zhang et al. (2019) use a Wi-Fi-based tracking technology to determine if phones were inside a specific store. Merging a dataset like this into the panel dataset we have would allow for more

accurate intended destination estimates.

CHAPTER 3

Price Optimization

In this chapter, we study how to optimally price reusable resources (or products) as a function of the remaining inventory. The elasticities we estimate in Chapter 2 can be fed into the models we develop in this section. We focus on the single-zone problem, where a single zone can be thought of as a single resource with finite inventory. For example, a single block with several parking spots on the block or a single cloud-based server with several processors. Then, in future work, we will consider the multi-zone problem, where we want to price a network of several reusable resources, each with a finite capacity, where there are substitution effects between products. The pricing policies are a function of the current state of the system, the remaining time in the horizon, and the expected future demand. Continuing with the previous examples, this would be setting the price for each block in a neighborhood of multiple streets blocks, where each block contains several parking spots, or setting the price for various servers where each server has different computing power, memory, and speed. These policies can also be used for pricing storage units (i.e., Public Storage, CubeSmart) and fashion rentals (i.e., Rent The Runway, Le Tote, Bag Romance).

In this chapter, we consider a single reusable product with a finite capacity. Assuming an infinite-time horizon, price-sensitive Poisson arrivals, and price-agnostic exponential service times, we want to compute the prices as a function of the number of occupied products. The objective is to set the prices to maximize the long-run average revenue, which is also known as expected revenue per unit time. This setting can be modeled as an Erlang loss system and several researchers have studied how to compute the optimal, congestion prices under

stationary (Low, 1974) and non-stationary (Yoon and Lewis, 2004) arrival and service rates. For more information on queuing systems and Erlang loss systems, we refer the reader to Kelly (2011); Allen (2014); Ross (2014).

In the stationary setting, the long-run average revenue objective can be written as

$$R(p_0, ..., p_n) = \sum_{i=1}^{n} \pi_i(p_0, ..., p_n) \cdot \lambda(p_i) \cdot p_i \cdot \frac{1}{\mu},$$

where there are n resources (or servers), p_i is the price to charge under when $i \in \{0, ..., n\}$ resources are occupied, $\lambda(p_i)$ is the Poisson arrival rate when the price is p_i, and μ is the exponential service rate. The parameter π_i is the stationary probability of having i resources in use, which is a function of the service rate, μ, and the arrival rates $\lambda(p_1), ..., \lambda(p_n)$.

Given any price vector and the corresponding stationary distribution, for every possible state of the system, we want to know the probability that a customer paid $p_0, ...,$ or p_n. For example, consider a system that has 3 customers in service. We want to know the expected number of the 3 customers in service that entered when the system was in state 0 (and therefore paid p_0), in state 1 (and therefore paid p_1), etc. There does not exist a way to compute this metric, so we name it the *conditional entry-state distribution*, and in this chapter we develop a tractable method for computing it. We show the algorithm we develop converges to the true *conditional entry-state distribution* for any Erlang loss system.

We note that with the conditional entry-state distribution, the long-run average revenue objective function can be rewritten as

$$R(p_0, ..., p_n) = \sum_{i=1}^{n} \pi_i(p_0, ..., p_n) \sum_{j=0}^{n-1} x_{j|i} \cdot p_j,$$

where $x_{j|i}$ is the conditional entry-state distribution and represents the expected number resources that went into service when j resources were in use, given the system currently has i resources in use.

3.1 Introduction

Erlang loss systems, which are also known as finite-state, Birth-Death Markov chains or M/M/n/n queues, have a rich history in stochastic modeling and have applications to transportation, health care, and communications. We specifically consider systems where the Poisson arrival rate fluctuates with the state, or the number of busy servers. State-dependent arrivals have been used to model customer behavior related to price sensitivity and balking. This model, in its purest form, with constant, state-independent arrival and departure rates, is extensively analyzed by Takacs (1969). Brumelle (1978) performs a comparable analysis, but models both state-dependent arrivals and departures, while we only consider the former. Burman (1981) examines a similar Erlang loss system while relaxing the Markovian service time assumption. In any of these systems, the operator may wish to know what is the probability a customer entered when the system was empty, when the system had 1 customer, and so on, given the current state, or occupancy, of the system. We denote this metric as the "conditional entry-state distribution," where conditioning is on the system's current state. In this paper, we develop an algorithm to compute its value.

One application of where this metric can be valuable is described in Low (1974). The setting consists of an Erlang loss system with Poisson arrival rates that are a decreasing function of price. The author is interested in finding the optimal entry fee to charge arriving customers at each state and develops an algorithm to compute the prices that maximize the long-run average revenue per unit time. Given any feasible set of prices, the system operator may like to know *where* the revenue is coming from at each state, so the operator would like to know the expected proportion of the customers who pay the entry fee when the system is empty, has 1 customer, and so on.

In these queuing systems, many steady-state performance statistics can be computed, such as the blocking probability, the steady-state distribution, the expected number of busy servers, and more. We refer the reader to Allen (2014) for further material on computable

system metrics. To the best of our knowledge, there does not exist a tractable way to compute the metric we study.

Since this work relates to the expected state at which customers enter, or arrive, to the system, the well-known "Poisson Arrivals See Time Averages" (PASTA) property (Wolff, 1982) and the related Conditional PASTA property (Van Doorn and Regterschot, 1988) are both relevant. The latter is pertinent since our metric also involves computing steady-state, stationary values that are conditional on the system being in a particular state. However, their work does not provide any method for computing any conditional metric, so this work compliments their result.

The paper is organized as follows. Section 3.2 describes the model. The algorithm and convergence proof follows in Section 3.3. Numerical results are detailed in Section 3.4, and future directions are summarized in Section 3.5.

3.2 Model

We consider an M/M/n/n queuing system with state-dependent arrival rates. This system can be modeled as a finite-state, birth-and-death Markov chain with state-space $S = \{1, \ldots, n\}$ representing the number of busy servers. We consider the system in steady-state, denote the current state as $Z \in S$, and use the indexes i, j, k to refer to arbitrary states. Customers arrive to the system according to a Poisson process with a state-dependent rate, i.e., for $Z = k$ the arrival rate is $\lambda_k > 0$. Service times for an individual server are exponentially distributed with rate μ, so we use $\mu_k = k \cdot \mu$ to denote the service rate of the entire system in state k.

We are interested in computing the steady-state probability that an in-service customer joined the system when there were i servers busy, given the system is currently in state j. Namely,

$$Pr[\text{In-service customer entered in state } i | Z = j], \ \forall i, j \in S.$$

We refer to the above probability as the *conditional entry-state probability*. To compute it, we define $\Omega_{ij} \in \mathbb{R}_+$ to be the expected number of in-service customers (out of j currently being served) who arrived when the system was in state i. To recover the conditional entry-state probability, we simply compute Ω_{ij}/j, hence hereafter we focus on obtaining Ω_{ij}. Since $\Omega_{ij} = 0$ when either $j = 0$ or $i = n$, we focus on $i \in S \setminus \{n\}$ and $j \in S \setminus \{0\}$. Thus, for each state j, we define the column vector $\Omega_j = [\Omega_{0j}, ..., \Omega_{n-1,j}] \in \mathbb{R}_+^n$, and we note that the entries of Ω_j sum to j: $\sum_{i=0}^{n-1} \Omega_{ij} = \Omega_j^\top \mathbf{1} = j$, where $\mathbf{1}$ is the vector of all ones. Finally, we define the matrix $\Omega = [\Omega_1, ..., \Omega_j, ..., \Omega_n] \in \mathcal{M} = \{Y \in \mathbb{R}_+^{n \times n} : \sum_i Y_{ij} = j, j = 1, ..., n\}$, where Y is an arbitrary matrix in the set $\mathcal{M}$, which contains matrices with positive entries where the entries of the j-th column sum to j.

3.2.1 System Dynamics

We now describe how to analytically compute Ω_{ij} by applying the conservation flow principle that is typically used to compute steady-state probabilities in Markov chains.

Before deriving the system of equations for Ω_{ij}, we first introduce some additional probabilities. The steady-state probability of a given state j can be expressed as

$$\pi_j = p_{j,j-1} \cdot \pi_{j-1} + p_{j,j+1} \cdot \pi_{j+1} \quad \forall j \in S \setminus \{0\}$$

where $p_{i,k}$ is the probability of transitioning to i from k and π_k are the steady-state probabilities. Next, we define "$j \leftarrow i$" to mean "Enter j from i". Using Bayes' rule we can then derive

$$Pr[j \leftarrow j-1 | Z = j] = \frac{p_{j,j-1} \cdot \pi_{j-1}}{p_{j,j-1} \cdot \pi_{j-1} + p_{j,j+1} \cdot \pi_{j+1}} = \frac{\mu_j}{\lambda_j + \mu_j} \tag{3.1}$$

The last equality is obtained by making two observations. First, transitions out of state j are with probability $p_{j-1,j} = \frac{\mu_j}{\lambda_j + \mu_j}$ to state $j-1$ if a departure occurs, or with probability $p_{j+1,j} = \frac{\lambda_j}{\lambda_j + \mu_j}$ to state $j+1$ if an arrival happens instead. Second, using the reversibility

property of M/M/n/n queuing systems (Kelly, 2011), which implies that $p_{j,j-1} = \frac{\mu_j}{\lambda_j+\mu_j} \cdot \frac{\pi_j}{\pi_{j-1}}$ and $p_{j,j+1} = \frac{\lambda_j}{\lambda_j+\mu_j} \cdot \frac{\pi_j}{\pi_{j+1}}$. For ease of exposition we hereafter define $\alpha_j := \frac{\mu_j}{\lambda_j+\mu_j}$ and $\boldsymbol{\alpha} = [\alpha_0, ..., \alpha_n]$, and note $\alpha_0 = 0$ and $\alpha_n = 1$.

We now proceed to write the system of equations to compute $\Omega_{i,j}$. We do this by conditioning on the two events that can lead to the system reaching state j: (1) from $j-1$ when an arrival occurs or (2) from $j+1$ when a departure occurs. Thus, the expected number of customers in state j who entered in state i ($\Omega_{i,j}$) can be expressed as a combination of: (1) the expected number of existing customers in state $j-1$ who first arrived when the system was in state i ($\Omega_{i,j-1}$) and the expected number of new customers into j who enter the system in state i ($\mathbb{I}[i = j-1]$) and (2) the expected number of existing customers in state $j+1$ who first arrived when the system was in state i ($\Omega_{i,j+1}$) minus the departing customers from state $j+1$ who first arrived when the system was in state i ($\frac{1}{j+1} \cdot \Omega_{i,j+1}$, where $\frac{1}{j+1}$ is the probability that any of the existing customers depart from the system, which is due to the memoryless property of exponential service times). Namely,

$$\begin{aligned}\Omega_{ij} =& (\Omega_{i,j-1} + \mathbb{I}[i = j-1]) \cdot Pr[j \leftarrow j-1 | Z = j] \\ &+ \left(\Omega_{i,j+1} - \frac{1}{j+1} \cdot \Omega_{i,j+1}\right) \cdot Pr[j \leftarrow j+1 | Z = j] \\ =& (\Omega_{i,j-1} + \mathbb{I}[i = j-1]) \cdot \alpha_j + \left(\Omega_{i,j+1} - \frac{1}{j+1} \cdot \Omega_{i,j+1}\right) \cdot (1 - \alpha_j) \end{aligned} \tag{3.2}$$

We extend this flow conservation between $\Omega_{ij}, \Omega_{i,j-1}$, and $\Omega_{i,j+1}$ to matrix form in Eq. (3.3) with the function $g : \mathcal{M} \mapsto \mathcal{M}$. We define e_j as the unit vector with a 1 in the j-th component, and use it to capture an arrival from state $j-1$ so the system evolves

from having $j-1$ to j servers busy.

$$g(Y) = \begin{cases} \alpha_1(e_1) + \frac{1-\alpha_1}{2} Y_2 & \text{if } j = 1 \\ \alpha_j(Y_{j-1} + e_j) + (1-\alpha_j)\frac{j}{j+1} Y_{j+1} & \text{if } 1 < j < n \\ Y_{n-1} + e_n & \text{if } j = n \end{cases} \tag{3.3}$$

To understand the intuition in g, consider the general case of $1 < j < n$: the function mixes the distribution at $j-1$ while considering an arrival into j (i.e., $Y_{j-1} + e_j$), and the distribution at $j+1$ considering the potential departures (i.e., $\frac{j}{j+1} Y_{j+1}$). We note g can be applied to any $Y \in \mathcal{M}$ and for each column Y_j, g computes a convex combination of two vectors that sum to j. This means $g(Y)$ maps to an element of $\mathcal{M}$.

3.3 An Algorithm to Compute the Conditional Expected Entry State

Next, we present an algorithm to compute Ω. To simplify notation, we define the function $f(Y) = g(g(Y)) = Y'$ and note that $f : \mathcal{M} \mapsto \mathcal{M}$. The output $Y' = [Y'_1, ..., Y'_j, ..., Y'_n]$ has several unique, structural properties, such as its columns Y'_j are weighted, linear sum of the columns of Y, where the coefficients are polynomial functions of $\boldsymbol{\alpha}$ (Property D.1.1). Lemma 3.3.1 states a key property that f is a contraction mapping. For readability, all properties and proofs are available in Appendix D.

Lemma 3.3.1. *The function $f : \mathcal{M} \mapsto \mathcal{M}$ is a contraction mapping.*

We next present a simple algorithm that repeatedly applies f to any matrix in $\mathcal{M}$ until the desired tolerance ϵ is reached. Then, in Theorem 3.3.1 we show that the algorithm converges to the correct value of Ω, and in Theorem 3.3.2, we give an upper bound on the algorithm's iteration limit. The proofs for both theorems are available in Appendices D.3-D.4.

Theorem 3.3.1. *For all $Y \in \mathcal{M}$, as $\epsilon \to 0$, the algorithm converges to Ω.*

Algorithm 1: Computing the Conditional Expected Entry State

Input: $\epsilon > 0, Y \in \mathcal{M}$

Compute $Y' = f(Y)$

while $d(Y, Y') = ||Y - Y'||_\infty = \max_{i,j}\{|Y_{ij} - Y'_{ij}|\} > \epsilon$ **do**

  Set $Y = Y'$

  Compute $Y' = f(Y)$

end

Output: Y'

Theorem 3.3.2. *For all $\epsilon \in (0,1)$ and $Y = [Y_1, ..., Y_n] \in \mathcal{M}$, where $Y_j = \frac{j}{n} \cdot \mathbf{1}$, the algorithm terminates in at most $\log_A(\epsilon) + 1$ iterations. The parameter $A = \max_{j=1,...,n}\{H_j(\boldsymbol{\alpha})\}$, where $H_j(\boldsymbol{\alpha})$, defined in Table D.1, is the maximum absolute difference in the estimate of Ω_j after one iteration.*

Theorem 3.3.2 presents an upper bound on the number of iterations the algorithm runs until convergence. However, we note that the bound is not tight. This is because A, the upper bound on the rate at which the distance d decreases in each iteration, is bounded above by $\frac{n-1}{n}$ and approaches 1 when n becomes large, so $\log_A(\epsilon)$ becomes large. When A does equal its upper bound of $\frac{n-1}{n}$, the term $\log_{\frac{n-1}{n}}(\epsilon)$ is concave increasing in n. In the next section, we observe that the numerical performance is much better.

3.4 Numerical Experiments

In this section we numerically demonstrate the speed and efficacy of our algorithm in computing the metric of interest Ω. For several system sizes $n \in \{2, 5, 10, 25, 50, 100, 250\}$, we generate 50 random instances where $\mu \sim \mathcal{U}(0, 5)$ and $\lambda_k \sim \mathcal{U}(0, 10)$ for $k = 0, ..., n$. For each instance, we run the algorithm with $\epsilon = 10^{-12}$ and initial point $Y = [Y_1, ..., Y_k, ..., Y_n] \in \mathcal{M}$, where $Y_k = \mathbf{1} \cdot \frac{k}{n}$, and we record the number of iterations and time to convergence. We also simulate the system for 25,000 arrivals and compute Ω using the second half of the simulation, when the system is in steady-state. We denote this value Ω^{sim}. Using the output of

the algorithm, Ω, and Ω^{sim}, we compute two metrics: the maximum and the mean absolute deviation between the Ω and Ω^{sim}. Experiments are run in Python 2.7 on a Dell Inspiron 13 with 16GB of RAM and an Intel Core i7 1.8GHz processor.

For each value of n, we report the average and standard deviation over the 50 random instances in Table 3.1. These results show the algorithm and simulation return values very close to each other. However, as n increases, the simulation rarely reaches some states in steady-state, so reliably computing Ω_{ij}^{sim} is challenging (and impossible when state j is never reached). This is evident in the last two columns of Table 3.1 where the deviation between Ω and Ω^{sim} increases with n because Ω^{sim}. The algorithm is never victim to this issue and always converges to the true value of Ω, which is why we can run the algorithm for all n but can only simulate for $n \leq 25$.

Table 3.1 also shows that the algorithm computes Ω faster than the simulation method. To capture the algorithm's convergence speed, in Figure 3.1 we report the algorithm's average distance by iteration over 50 random instances. The figure shows that the algorithm converges rapidly for several values of n. We note that there is a "kink" and speed-up around the $n/2$-th iteration.

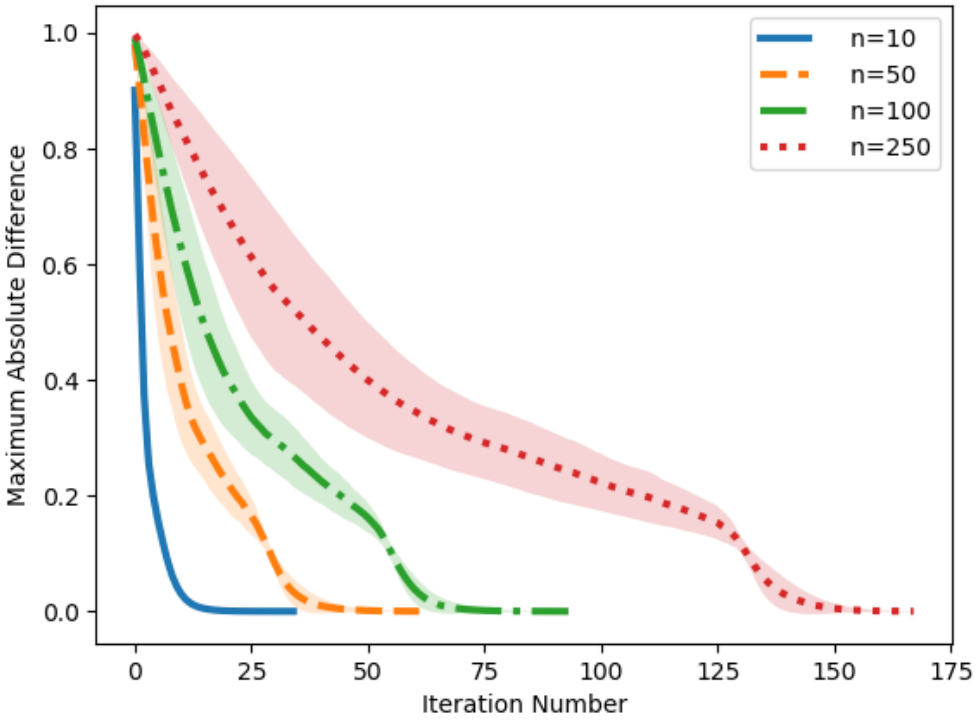
One potential interpretation is related to the structure of $Y' = f(Y)$. Per Property D.1.1, Y'_j has two components: a weighted sum of the columns of Y and a vector C_j. With each iteration, the weights on the columns of Y approach 0, so the influence of Y effectively vanishes after many iterations. The vector C_j is a function of $\boldsymbol{\alpha}$. After one iteration, C_j is only a function of α_{j-1}, α_j, and α_{j+1}, but after after two iterations, C_j is a function of $\alpha_{j-2}, \ldots, \alpha_{j+2}$. It takes at most $n/2$ iterations for C_j to be a function of all $\alpha_1, \ldots, \alpha_n$. Once C_j depends on all α_j, the vector finally captures the entire system's dynamics, so the convergence rate increases after this iteration.

Table 3.1: Comparison of Algorithm and Simulation

n	Iterations	Algorithm Time in Seconds	Simulation Time in Seconds	$\max_{ij} \lvert\Omega_{ij} - \Omega_{ij}^{sim}\rvert$	$\frac{1}{n^2}\sum_{ij} \lvert\Omega_{ij} - \Omega_{ij}^{sim}\rvert$
2	24.66	0.26	50.42	0.009	0.007
	(4.03)	(0.05)	(1.24)	(0.006)	(0.005)
5	57.34	1.00	67.19	0.084	0.020
	(14.28)	(0.25)	(2.37)	(0.039)	(0.006)
10	68.78	2.02	92.43	0.914	0.100
	(22.22)	(0.65)	(4.04)	(0.395)	(0.034)
25	84.44	5.45	159.92	1.247	0.275
	(27.2)	(1.76)	(7.86)	(0.505)	(0.014)
50	99.00	12.22	-	-	-
	(31.74)	(3.92)	-	-	-
250	200.66	125.79	-	-	-
	(26.57)	(16.58)	-	-	-

Note: Standard deviation values are shown in parenthesis.

Figure 3.1: Mean Convergence Rate with Shaded Standard Deviation for Various n.



Note: We report average distance values for all iterations up until the quickest random instance terminates.

3.5 Future Work and Extensions

An area for future work is to improve the bound we present for the algorithm's iteration limit. One approach worth exploring involves deriving an analytical expression to compute any value in the Cauchy sequence converging Ω_{ij}. This is different from our approach which shows a contraction modulus less than 1 exists. Two possible extensions include the cases with finite capacities (i.e. M/M/n/L queues with L > n) and with infinite capacity (i.e. $L = \infty$).

APPENDIX A

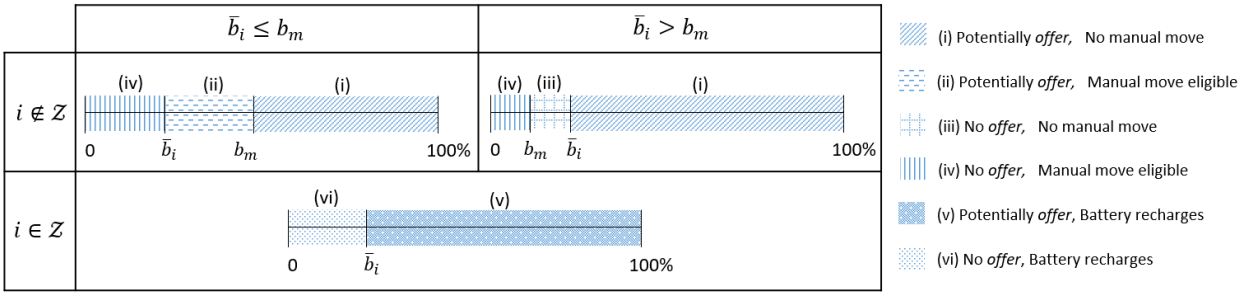
Appendix for Chapter 1

A.1 Notation Table

A.2 *1VMC* Dynamic Program for an Arbitrary Discount

The state of a vehicle, (i, w), can be categorized into 6 cases, which we outline in Figure A.1 and Table A.2. Based on each case, the value function takes a different form and in three of the cases the system operator needs to decide whether or not to offer a discounted ride. For the *1VMC* network with a general $1 - \sigma$ discount for $\sigma \in [0, 1]$, the value function for each case is provided in equations (i)-(vi).

Figure A.1: Description of Different Cases for State (i, w) in *1VMC*.



Note: For a vehicle at (i, w), there are six cases depending on if the vehicle is at a charging station (i.e., $i \notin \mathcal{Z}$), if the vehicle can reach a charging station (i.e., $w \ge \bar{b}_i$), and if the vehicle is eligible for a manual move (i.e., $w \ge b_m$).

Table A.1: Summary of Notation.

Parameter	Description
$\mathcal{R}$	Set of regions in the network, where i, j denote arbitrary locations in $\mathcal{R}$.
$\mathcal{Z} \subset \mathcal{R}$	Regions with charging stations, where z denotes an arbitrary charging station in $\mathcal{Z}$.
$\mathcal{R}(i, w) \subseteq \mathcal{R}$	Subset of regions that can be reached from location $i \in \mathcal{R}$ with battery $w \in \mathcal{W}$. Defined as $\mathcal{R}(i, w) = \{j \in \mathcal{R} : w \geq b_{ij}\}$.
$\mathcal{W}$	Set of feasible battery levels. Values are equally separated by δ, so $\mathcal{W} = [0, \delta, 2\delta, ..., 1]$.
$\delta \in [0, 1]$	Battery increment used in $\mathcal{W}$.
$\sigma \in [0, 1]$	Percentage of the full fare that the firm decides to offer. $\sigma = 1$ corresponds to no discount and $\sigma = 0$ corresponds to a 100%, or free-ride, discount.
n	Number of vehicles in the network.
m	Number of charging stations in the network, so $\lvert\mathcal{Z}\rvert = m$.
$b_m \in [0, 1]$	Battery threshold for manual move. If a parked vehicle has a battery level $w < b_m$, then this vehicle is eligible for manual repositioning.
$c_m \in [0, 1]$	Cost of a manual repositioning move.
$p_m \in [0, 1]$	Probability that a manual repositioning move occurs.
$t_m \in [0, 1]$	Mean number of periods for a manual move to be completed.
$\gamma \in \mathcal{W}$	Battery recharge rate.
$\bar{b}_i \in \mathcal{W}$	The minimum battery required to reach the nearest charging station from i. Defined as $\bar{b}_i = \min_{z \in \mathcal{Z}}\{b_{iz}\}$.
$z_i \in \mathcal{Z}$	Closest charging station to region $i \in \mathcal{R}$.
$\lambda_i \in [0, 1]$	Probability of seeing a request for a vehicle in region $i \in \mathcal{R}$.
$p_{ij} \in [0, 1]$	Probability of a ride starting at region $i \in \mathcal{R}$ and ending at $j \in \mathcal{R}$.
$b_{ij} \in \mathcal{W}$	Battery consumption of a ride starting at region $i \in \mathcal{R}$ and ending at $j \in \mathcal{R}$.
$d_{ij} \in \mathbb{R}^{+}$	Distance between regions $i, j \in \mathcal{R}$. We note that $d_{ii} = 0$ and the distance between regions does not depend on the direction, so $d_{ij} = d_{ji}$.
$f_{ij} \in \mathbb{R}^{++}$	Revenue or fare of a ride starting at region $i \in \mathcal{R}$ and ending at $i \in \mathcal{R}$.
$t_{ij} \in \mathbb{Z}^{++}$	Duration of a ride starting at region $i \in \mathcal{R}$ and ending at $j \in \mathcal{R}$.
$\beta \in (0, 1)$	Discount factor used in the dynamic programs.
$\beta_{ij} \in (0, 1)$	Adjusted discount factor of a ride starting at region $i \in \mathcal{R}$ and ending at $j \in \mathcal{R}$. Defined as $\beta_{ij} = \beta^{t_{ij}}$.
$u(d, f)$	The utility gained on a ride where d is the distance between the vehicle drop-off destination and desired destination, and f is the fare for the ride.
$\mathbb{P}(Accept_{ijz})$	The probability a customer accepts the free-ride from i to z instead of paying the full fare to go to j, the desired destination. Defined as $\mathbb{P}\big[u(d_{zj}, 0) \geq u(0, f_{ij})\big]$.
$\mathcal{DM}$	Dollar-to-Mile ratio captures the amount of money a user would take in exchange for walking 1 mile. The units are $\frac{\$}{Mile}$.

Table A.2: Description of Six Cases for State (i, w) in *1VMC*

Case	Region	Charging Station Accessible	Manual Move Eligible	Description of Case
(i)	$i \notin \mathcal{Z}$	$w \geq \bar{b}_i$	$w \geq b_m$	Do not move and potentially offer.
(ii)			$w < b_m$	Move and potentially offer.
(iii)		$w < \bar{b}_i$	$w \geq b_m$	Do not move and do not offer.
(iv)			$w < b_m$	Move and do not offer.
(v)	$i \in \mathcal{Z}$	$w \geq \bar{b}_i$	-	Battery replenishes and potentially offer.
(vi)		$w < \bar{b}_i$	-	Battery replenishes and do not offer.

Note: This table maps a vehicle's state (i, w) to one of six cases in the *1VMC* dynamic program. The first four rows are for vehicles not in charging stations stations and the last two rows are for vehicles in charging stations. The column "Charging Station Accesible" contains $w \geq \bar{b}_i$ if a vehicle at (i, w) can reach a charging station. The column "Manual Move Eligible" contains $w < b_m$ if a vehicle at (i, w) is eligible to be manually repositioned.

$$
\begin{aligned}
V(i,w) = \max_{\sigma\in[0,1]} \Bigg\{ & \lambda_i \sum_{j\in\mathcal{R}(i,w)} p_{ij} \cdot \Big(\mathbb{P}(Decline^{\sigma}_{ijz_j}) \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij}) \big) \\
& + \mathbb{P}(Accept^{\sigma}_{ijz_j}) \cdot \big(\sigma f_{ij} + \beta_{iz_j} V(z_j, w - b_{iz_j}) \big) \Big) \Bigg\} \\
& + \Big(1 - \lambda_i \sum_{j\in\mathcal{R}(i,w)} p_{ij} \Big) \cdot \beta V(i,w) \qquad \text{(i)} \\
V(i,w) = & p_m \cdot \big(-c_m + \beta_m V(z_i, w) \big) + (1 - p_m) \cdot \\
& \Bigg(\max_{\sigma\in[0,1]} \Bigg\{ \lambda_i \sum_{j\in\mathcal{R}(i,w)} p_{ij} \cdot \Big(\mathbb{P}(Decline^{\sigma}_{ijz_j}) \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij}) \big) \\
& + \mathbb{P}(Accept^{\sigma}_{ijz_j}) \cdot \big(\sigma f_{ij} + \beta_{iz_j} V(z_j, w - b_{iz_j}) \big) \Big) \Bigg\} \\
& + \Big(1 - \lambda_i \sum_{j\in\mathcal{R}(i,w)} p_{ij} \Big) \cdot \beta V(i,w) \Bigg) \qquad \text{(ii)}
\end{aligned}
$$

$$V(i,w) = \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big) + \Big(1 - \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij}\Big) \cdot \beta V(i,w) \tag{iii}$$

$$\begin{aligned} V(i,w) = & p_m \cdot (-c_m + \beta_m V(z_i, w)) + (1 - p_m) \cdot \\ & \left(\lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big) + \Big(1 - \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij}\Big) \cdot \beta V(i,w) \right) \end{aligned} \tag{iv}$$

$$\begin{aligned} V(i,w) = \max_{\sigma \in [0,1]} \Bigg\{ & \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \Big(\mathbb{P}(Decline^{\sigma}_{ijz_j}) \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big) \\ & + \mathbb{P}(Accept^{\sigma}_{ijz_j}) \cdot \big(\sigma f_{ij} + \beta_{iz_j} V(z_j, w - b_{iz_j})\big) \Big) \Bigg\} \\ & + \Big(1 - \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij}\Big) \cdot \beta V(i, \min\{w + \gamma, 1\}) \end{aligned} \tag{v}$$

$$\begin{aligned} V(i,w) = & \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij} \cdot \big(f_{ij} + \beta_{ij} V(j, w - b_{ij})\big) \\ & + \Big(1 - \lambda_i \sum_{j \in \mathcal{R}(i,w)} p_{ij}\Big) \cdot \beta V(i, \min\{w + \gamma, 1\}) \end{aligned} \tag{vi}$$

A.3 Example More Battery Being Less Lucrative

The following example illustrates that more battery can be less lucrative. Let us consider a network with three regions and one charging station, $\mathcal{R} = \{1, 2, 3\}$ and $\mathcal{Z} = \{2\}$, respectively. The arrival and transition probabilities, revenues, battery consumption, and manual repositioning parameters are defined in Table A.3. Furthermore, we assume that users accept a free ride half of the time. We plot the optimal value function at region #1 in Figure A.2 for two values of b_m, the battery threshold required for a vehicle to be manually repositioned. When $b_m = 50\%$, more battery is always lucrative, however when $b_m = 25\%$, more battery can be less lucrative ($w \leq 25\%$).

Table A.3: Parameters for Example where More Battery is Less Lucrative

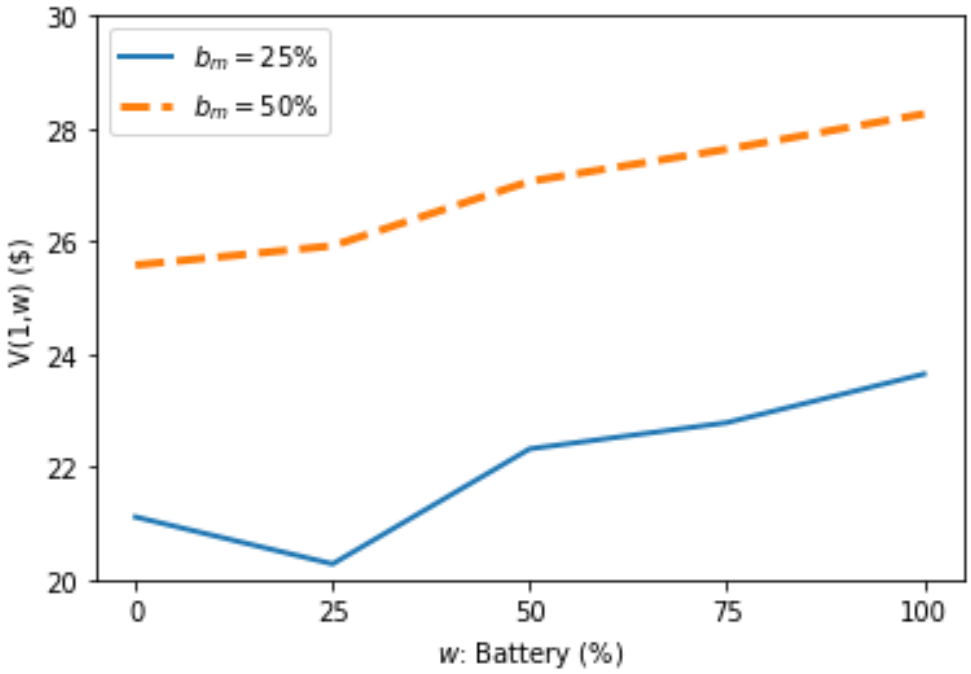
p_{ij}	1	2	3
1	1/3	1/3	1/3
2	1/3	1/3	1/3
3	1/3	1/3	1/3

f_{ij}	1	2	3
1	\$1	\$2	\$2
2	\$2	\$1	\$2
3	\$2	\$2	\$1

b_{ij}	1	2	3
1	25%	50%	75%
2	50%	25%	50%
3	75%	50%	25%

λ_1	1/3
λ_2	1/3
λ_3	1/3
p_m	0.5
b_m	25%
c_m	\$1

Figure A.2: More Battery is Not Always Lucrative



A.4 Proofs

Proof of Proposition 1.3.1.

For an arbitrary region i, we show that there exist threshold battery levels w_1^i and w_2^i such that $\tilde{x}^o(i,w) = 1$ if $w_1^i \leq w \leq w_2^i$ and $\tilde{x}^o(i,w) = 0$ otherwise. Based on our construction of $\tilde{\pi}$, this is equivalent to showing that $\tilde{\pi} \in \Pi_{SOR}$. We note that if $\sum_{w\in\mathcal{W}} \tilde{x}^o(i,w) = 0$, then the result holds trivially. Hence for the remainder of the proof, we assume that $\sum_{w\in\mathcal{W}} \tilde{x}^o(i,w) \geq 1$. First, we consider the case in which $\tilde{x}^o(i,\bar{b}_i) = 1$ and so we know that $\sum_{w\in\mathcal{W}} \tilde{y}(i,w) = 0$ by constraint (5). In this case, we claim that $\tilde{\pi}$ is the single-offer range policy where $w_1^i = \bar{b}_i$ and $w_2^i = \max\{w \in \mathcal{W} : \tilde{x}^o(i,w) = 1\}$. To see this, note that constraint (4) together with the definition of w_2^i ensure that both $\tilde{x}^o(i,w) = 1$ if $w_1^i \leq w \leq w_2^i$ and $\tilde{x}^o(i,w) = 0$ if $w > w_2^i$.

Next, we consider the case in which $\tilde{x}^o(i,\bar{b}_i) = 0$. Here, we see that constraint (5) ensures that there must exists $w' \in \mathcal{W}$ such that $\tilde{y}(i,w') = 1$. We claim that in this case, $\tilde{\pi}$ is the single-offer range policy with $w_1^i = w'$ and $w_2^i = \max\{w \in \mathcal{W} : \tilde{x}^o(i,w) = 1\}$. To show this claim, we first note that we must have $\tilde{x}^o(i,w) = 0$ for $\bar{b}_i \leq w < w'$ due to constraint (4) and the assumption that $\tilde{x}^o(i,\bar{b}_i) = 0$ and the fact that $\tilde{y}(i,w) = 0$ for a fixed region i and any $w \neq w'$. Finally, using the same argument as above, we can show that $\tilde{x}^o(i,w) = 1$ if $w_1^i \leq w \leq w_2^i$ and $\tilde{x}^o(i,w) = 0$ if $w > w_2^i$. □

Proof of Theorem 1.3.1.

We focus on the case where $w \geq \max\{\bar{b}_i, b_m\}$ for the definition of the value functions, which is the most common case, and note that the proof goes through in the case when $\bar{b}_i \leq w < b_m$. We assume that $M \geq \max\{\tilde{Z}, \max_{\pi\in\Pi_{SOR}} Z(\pi)\}$. First, we show that $\tilde{Z} \leq \max_{\pi\in\Pi_{SOR}} Z(\pi)$. Let $\tilde{V}(i,w)$ be the optimal $V(\cdot)$ decision variables in *Single Threshold*, and abusing notation slightly, we let $\tilde{V}^o(i,w)$ and $\tilde{V}^{no}(i,w)$ denote the value of $V^o(i,w)$ and $V^{no}(i,w)$ at optimality. We establish that $\tilde{Z} \leq \max_{\pi\in\Pi_{SOR}} Z(\pi)$ by showing that $\tilde{V}(i,w)$ is

a feasible solution to *LP Policy* when $\pi = \tilde{\pi}$. First, we note that at optimality $\tilde{V}(i,w) = \min\{\tilde{V}^o(i,w) + M\tilde{x}^{no}(i,w), \tilde{V}^{no}(i,w) + M\tilde{x}^o(i,w)\}$ for $i \in \mathcal{R} \setminus \mathcal{Z}, w \geq \max\{b_m, \bar{b}_i\}$. Without loss of generality, we assume $\tilde{x}^o(i,w) = 1$ and $\tilde{x}^{no}(i,w) = 0$, since the other case in symmetric. Then, we have

$$\begin{aligned}\tilde{V}(i,w) &= \min\{\tilde{V}^o(i,w), \tilde{V}^{no}(i,w) + M\} \\ &= \tilde{V}^o(i,w) \\ &= \mathbf{1}_{(i,w)\in S_{\tilde{\pi}}}\tilde{V}^o(i,w) + \mathbf{1}_{(i,w)\notin S_{\tilde{\pi}}}\tilde{V}^{no}(i,w).\end{aligned}$$

The second equality follows because $M \geq \tilde{Z} \geq \tilde{V}(i,w) = \tilde{V}^o(i,w)$ and third equality follows by definition of $S_{\tilde{\pi}}$. The other constraints are trivially satisfied and hence we get that $\tilde{V}(i,w)$ is feasible to *LP Policy* when $\pi = \tilde{\pi}$ and so $\tilde{Z} = \sum_{i\in\mathcal{R}}\sum_{w\in\mathcal{W}}\tilde{V}(i,w) \leq \max_{\pi\in\Pi_{SOR}} Z(\pi)$ since $\tilde{\pi} \in \Pi_{SOR}$ by Proposition 1.

Next, we show that $\tilde{Z} \geq \max_{\pi\in\Pi_{SOR}} Z(\pi)$. Let $\pi^*_{SOR} = \arg\max_{\pi\in\Pi_{SOR}} Z(\pi)$ be the optimal single-offer range free-ride policy and let $V_\pi(i,w)$ be the optimal decision variables to *LP Policy* when $\pi = \pi^*_{SOR}$. Further, for each region i, assume that w_1^{i*} and w_2^{i*} give the critical battery level threshold under π^*_{SOR}. To show the desired result, we construct a feasible solution $\hat{V}(i,w), \hat{x}^o(i,w), \hat{x}^{no}(i,w), \hat{y}(i,w)$ for each $i \in \mathcal{R}$ and $w \in \mathcal{W}$ to *Single Threshold* that achieves an objective of $Z(\pi^*_{SOR})$. First, we set $\hat{x}^o(i,w) = 1$ if $(i,w) \in S_{\pi^*_{SOR}}$, $\hat{x}^o(i,w) = 0$ if $(i,w) \notin S_{\pi^*_{SOR}}$. We then set $\hat{y}(i,w) = 1$ if $w = \max\{w > \bar{b}_i : (i,w) \in S_{\pi^*_{SOR}}\}$ and $\hat{y}(i,w) = 0$ if $w \neq \max\{w > \bar{b}_i : (i,w) \in S_{\pi^*_{SOR}}\}$. Finally, we set $\hat{V}(i,w) = V_\pi(i,w)$ and let $\hat{V}^o(i,w)$ and $\hat{V}^{no}(i,w)$ be the resulting values of $V^o(i,w)$ and $V^{no}(i,w)$. First, we show that this solution is feasible in *Single Threshold*. We trivially get that $\hat{x}^o(i,w) + \hat{x}^{no}(i,w) = 1$ and that constraint (5) is satisfied by construction. To show that constraint (4) is satisfied note that we have $\hat{x}^o(i,w-\delta) = \hat{x}^o(i,w)$ if $w \notin \{w_1^{i*}, w_2^{i*} - \delta\}$. If $w = w_1^{i*}$, then we get that $\hat{x}^o(i,w) - y(i,w) = 0 \leq \hat{x}^o(i,w-\delta)$ and if $w = w_2^{i*} - \delta$ then we get that $x^o(i,w-\delta) = 1$ and so the constraint must be satisfied. Hence it remains to show that $\hat{V}(i,w) \leq \min\{\hat{V}^o(i,w) +$

$M\hat{x}^{no}(i,w), \hat{V}^{no}(i,w) + M\hat{x}^{o}(i,w)\}$ when $w \geq \max\{b_m, \bar{b}_i\}$. We again assume without loss of generality that $\hat{x}^{o}(i,w) = 1$ and $\hat{x}^{no}(i,w) = 0$. We have that

$$\begin{aligned}\hat{V}(i,w) = V_\pi(i,w) &= \hat{V}^{o}(i,w) \\ &\leq \min\{\hat{V}^{o}(i,w), \hat{V}^{no}(i,w) + M\} \\ &= \min\{\hat{V}^{o}(i,w) + M\hat{x}^{no}(i,w), \hat{V}^{no}(i,w) + M\hat{x}^{o}(i,w)\},\end{aligned}$$

where the second equality follows by definition of π^*_{SOR} and the fact that we assumed that $\hat{x}^{o}(i,w) = 1$. The third equality follows because $M \geq Z(\pi^*_{SOR}) \geq V_\pi(i,w) = \hat{V}^{o}(i,w)$. Finally note that this solution attains the desired objective value of $\sum_{i\in\mathcal{R}}\sum_{w\in\mathcal{W}}\hat{V}(i,w) = Z(\pi^*_{SOR})$ and so we have established that $Z(\tilde{\pi}) = \tilde{Z} = Z(\pi^*_{SOR})$ as desired. □

A.5 Description of Discrete Event Simulation

In this section we provide additional details of our simulation and experiments to complement the description provided in Section 4 of the paper. First we describe how we use the historical data to estimate trip parameters, such as duration, battery consumption, and fare, between regions in the network. Then we describe how to solve for the optimal discount policies. And finally, we describe the dynamics of the simulation, which includes how the system is seeded, how trips are generated, and how manual moves take place.

A.5.1 Estimating Parameters

Trip Features. Using all trips in the data, we run three linear regressions to estimate a trip's mileage (m), duration (t), and battery consumption (b). We index trips from $k = 1, ..., K$ and in all regressions we use the geocoordinates of the trip's origin and destination, respectively $(i_k^{\text{lat}}, i_k^{\text{long}})$ and $(j_k^{\text{lat}}, j_k^{\text{long}})$, and the distance between the two locations, $d_{i_k j_k}$. The output to the regression in Eqs. (A.1)-(A.3) is available in Table A.4.

$$m_k = \beta_0 + \beta_{i^{\text{lat}}} \cdot i_k^{\text{lat}} + \beta_{i^{\text{long}}} \cdot i_k^{\text{long}} + \beta_{j^{\text{lat}}} \cdot j_k^{\text{lat}} + \beta_{j^{\text{long}}} \cdot j_k^{\text{long}} + \beta_d \cdot d_{i_k j_k} + \epsilon_k \tag{A.1}$$

$$t_k = \beta_0 + \beta_{i^{\text{lat}}} \cdot i_k^{\text{lat}} + \beta_{i^{\text{long}}} \cdot i_k^{\text{long}} + \beta_{j^{\text{lat}}} \cdot j_k^{\text{lat}} + \beta_{j^{\text{long}}} \cdot j_k^{\text{long}} + \beta_d \cdot d_{i_k j_k} + m_k + \epsilon_k \tag{A.2}$$

$$b_k = \beta_0 + \beta_{i^{\text{lat}}} \cdot i_k^{\text{lat}} + \beta_{i^{\text{long}}} \cdot i_k^{\text{long}} + \beta_{j^{\text{lat}}} \cdot j_k^{\text{lat}} + \beta_{j^{\text{long}}} \cdot j_k^{\text{long}} + \beta_d \cdot d_{i_k j_k} + m_k + t_k + \epsilon_k \tag{A.3}$$

We use the above regressions to generate the specific features of a ride from i to j in the simulation. Thus, using Eqs. (A.1)-(A.3) and the coefficients detailed in Table A.4, we compute each trip's predicted mileage ($\hat{m}_{ij}$), predicted duration ($\hat{t}_{ij}$), and predicted battery consumption ($\hat{b}_{ij}$). Using $\hat{t}_{ij}$, we compute the predicted fare $f_{ij} = \$1 + \$0.15 \cdot \hat{t}_{ij}$. Since the units of $\hat{t}_{ij}$ are in minutes, the value must be converted to periods (recall that 1 period equals 2.66 min) to compute the adjusted discount rate β_{ij}.

Users Utility. To compute the utility gained from a trip, we assume the utility function $u(d, f)$ takes the form $u(d, f) = -\alpha_d \cdot d - \alpha_f \cdot f$. With this structure, the utility is decreasing in both distance and price and the maximum utility is 0. We assume $\alpha_d \sim U(0, \mathcal{DM})$ and $\alpha_f \sim U(0, 1)$, and use $\mathcal{DM} = \$5$/mile as the baseline value. This value is equivalent to \$15 per hour, which will be in the minimum wage in 2023 in California, where the EVSS collaborator is based, when we assume people walk at 3 miles per hour. Under this utility model, the probability of accepting a free ride is given in Eq. (A.4). To see how $\mathbb{P}\big(Accept_{ijz_j}\big)$ fluctuates relative to the willingness to pay and walk, in Table A.5 we show the average value over all feasible origin-destination routes in historical ride data.

$$\mathbb{P}\big(Accept_{ijz_j}\big) = \begin{cases} \frac{f_{ij}}{d_{z_j j}} \cdot \frac{1}{2 \cdot \mathcal{DM}} & , \frac{f_{ij}}{d_{z_j j}} \leq \mathcal{DM} \\ 1 - \frac{\mathcal{DM}}{2} \cdot \frac{d_{z_j j}}{f_{ij}} & , \text{otherwise} \end{cases} \tag{A.4}$$

Table A.4: Regression Output.

Dependent Variable	β_0	i^{lat}	i^{long}	j^{lat}	j^{long}	d_{ij}	True Distance (Miles)	True Duration (Minutes)	R^2	σ: Standard Error of Regression
m	-1089.98	-0.79	-3.77	-1.81	-5.96	0.48	-	-	0.08	2.07
t	11309.20	-9.76	30.07	16.60	64.31	-8.76	11.21	-	0.55	21.30
b	-2529.54	15.31	23.38	5.49	-37.63	-0.23	4.39	0.01	0.63	7.53

Table A.5: Average Probability of Accepting a Free-Ride Offer for Various Dollar-to-Mile ($\mathcal{DM}$) Values.

$\mathcal{DM}$ (\$/Mile)	0.5	1	2	5	10	20
Avg. $\mathbb{P}\big(Accept_{ijz_j}\big)$	94.6%	91.9%	86.6%	72.5%	58.9%	49.3%

A.5.2 Computing Policies

Using the sets $\mathcal{R}$, $\mathcal{Z}$, and $\mathcal{W}$, and the parameters λ_i, p_{ij}, $\hat{t}_{ij}$, $\hat{b}_{ij}$, f_{ij}, β_{ij}, β, $\mathbb{P}(Accept_{ijz})$, c_m, b_m, p_m, t_m, and γ, we solve for the optimal 1VMC-SOR policy, by solving the MIP described in *Single Threshold*, and we solve for the optimal 1VMC-50 policy via linear programming. We found that value iteration and policy iteration did not scale well to large, realistic instances. Note that policies need to be recomputed anytime any of these parameters change. We provide a description of how we compute the NVMC-SALP policy in Appendix A.8.

A.5.3 Generating Trips in Simulation

A new trip is generated as follows,

- Sample inter-arrival time τ from the Beta Prime($\alpha = 0.92, \beta = 4.07$, location $= 0.01$, and scale $= 8.86$) distribution. We set the ride request's arrival time to $T + \tau$, where T is the simulation's current time.
- Sample i, the trip's origin region, from the discrete distribution of λ_i.
- Given the trip's origin region i, sample the destination region j from the discrete distribution of p_{ij}.
- Given i and j, we randomly generate the ride features (we use the tilde above m, t, and b to denote the trip specific features) by using the regression output in Table A.4 as follows:

- We sample a random error $\epsilon \sim \mathcal{N}(0, \sigma_m)$. Using the predicted mileage value $\hat{m}_{ij}$ computed using the regression in Eq. (A.1) and the corresponding coefficients in Table A.4, we set $\tilde{m}_{ij} = \hat{m}_{ij} + \epsilon$.
- Taking $\tilde{m}_{ij}$ to be the trip's true mileage, we use the regression in Eq. (A.2) and the corresponding coefficients in Table A.4 to compute the trip's duration $\tilde{t}_{ij}$. In order to ensure that the trip's duration and mileage are related, we do not sample an error term when generating the trip's duration.
- Taking $\tilde{m}_{ij}$ and $\tilde{t}_{ij}$ to be respectively the trip's true mileage and duration, we use the regression in Eq. (A.3) and the coefficients in Table A.4 to compute the trip's battery consumption $\tilde{b}_{ij}$. Again, we do not sample an error term when generating $\tilde{b}_{ij}$ to ensure that the trip's mileage, duration, and battery consumption are correlated.
- Finally, using $\tilde{t}_{ij}$, the duration of the trip in minutes, we compute the fare for the trip $f_{ij} = \$1 + \$0.15 \cdot \tilde{t}_{ij}$. This corresponds to a base fare of \$1 per trip plus a per-minute charge of \$0.15/minute.

- To generate the customer's willingness to walk and pay, we sample $\alpha_d \sim U(0, \mathcal{DM})$ and $\alpha_f \sim U(0, 1)$.

Therefore, a new trip request is characterized by the following information: ($T + \tau$, i, j, $\tilde{t}_{ij}$, $\tilde{b}_{ij}$, f_{ij}, α_d, α_f). We note that we generate ride requests dynamically in the simulation, that is, at any time T, we can generate a stream of future demand requests. This approach is in contrast to generating all demand requests in advance, which turned out to be much more computationally expensive.

A.5.4 Simulation Dynamics

Initializing System.

We track time in seconds and the simulation starts at $T = 0$. We randomly assign $n = 300$, fully-charged vehicles, to regions according to λ_i. We store the location and battery of each vehicle in a System State (SS) table. We also track in SS whether or not a vehicle is busy on a ride or being manually repositioned. Thus, vehicles status is either *idle* or *busy*. We seed the system with a single ride based on the procedure described in Appendix A.5.3, and store the request arrival time and the corresponding ride information in a Pending Events (PE) list. In PE we also keep track of in-progress rides and manual moves. We denote $Time\_Event$ as the time when the pending event is scheduled to occur in the simulation, that is, the request arrival time for a new trip, the completion time for an in-progress ride, and the completion time for an in-progress manual move. While each policy we test runs in its own simulation and has its own SS table and PE list, they all start under the same configuration and see the same stream of demand.

Run Simulation.

While $T \leq 100$ days, we sequentially perform the following six steps:

1. Find "next event" (NE). The NE is the event in the PE list with the minimum $Time\_Event$. Thus, we sort the PE list in ascending order of $Time\_Event$ and update $T' \leftarrow T$ and $T \leftarrow \min\{Time\_Event\}$, where T' is the time of the previous event. If NE occurs on the next day, then we record the status of the network and the corresponding performance measures that we track. Before we update the system based on the NE type (Steps 4-6), we update the battery of idle vehicles at charging stations (Step 2) and potentially schedule a manual move for a vehicle with remaining battery below b_m (Step_3).

2. Update battery. For idle vehicles in table SS that are at charging stations, we update the battery level at the rate of corresponding to 5 hours for a full charge for a time interval corresponding to $T - T'$.

3. Schedule manual move of vehicles. We check SS for idle vehicles not parked at charging stations that have remaining battery below the move threshold b_m. If there are any move-eligible vehicles, then a crew member will arrive with probability p_m. If there is more than one move-eligible vehicle, then we select one to move at random. The move is scheduled as an in-progress move on the PE list with $Time\_Event$ set to $T + t'_m$, where t'_m is drawn from a truncated normal distribution with a mean of 4 hours and standard deviation of 30 minutes. The selected vehicle will be moved to the nearest charging station at $Time\_Event$. The SS table is updated so the selected vehicle is shown as *busy* until the repositioning is completed.

4. The NE is a new request for a ride.

   (a) Check for available vehicles. We search the SS table for idle vehicles at the arrival region and in the regions immediately around the arrival region. If there are no available vehicles, we track the unmet demand due to not having a vehicle present. If there are vehicles, but they do not have enough battery to fulfill the ride, we track the unmet demand due to not having enough battery. Then we go to step (c).

   If there are available, idle vehicles at the arrival region, or in the neighborhood of the arrival region, that have enough battery to complete the ride, then we assume users select the highest charged vehicle. We update the status of the vehicle in the SS table to be *busy*.

   (b) Users' choice of drop-off location. Regardless of the policy in place, at the time of renting the vehicle, users weigh the utility of ending the ride at their intended destination with the utility of ending the ride at the charging station

closest to their intended destination. Based on this decision, we schedule an in-progress ride on the PE list with a completion time and end destination of either $Time\_Event = T + \tilde{t}_{ij}$ and j or $T + \tilde{t}_{iz_j}$ and z_j. We make several distinctions in how customers choose their end destination depending on the policy being tested in the simulation:

Fine-Based policy: We note that the user only has to decide on their end destination if (i) their end destination is not at a charging station and (ii) the remaining battery at the end of the ride will be less than b_m, the manual move threshold. To decide on the end destination, the user computes her utilities $u(0, f_{ij} + fine)$ and $u(d_{z_j j}, f_{ij})$ using α_d, α_f, the customer's willingness to walk and pay parameters. If $u(0, f_{ij} + fine) \leq u(d_{z_j j}, f_{ij})$, the user chooses z_j as the end destination and the system accrues f_{ij} in revenue. Otherwise, the end destination is the user's intended destination, and the system collects f_{ij} in revenue plus the fine. However, we do not count the fine as revenue since this money will be used to subsidize the cost of a manual move. Regardless of what the user ultimately decides, we track that the customer had a decision to make, whether the customer paid the fine or not, and the utility gained in the chosen action.

Free-ride policy: First we note that the user only has a decision to make if the system actually offers a free ride to a charging station. The system does this by checking if, given the vehicle's location and the vehicle's remaining battery, offering is the optimal action. In other words, the operator offers a free ride if the state describing the location and remaining battery of the vehicles to be rented $(i, w) \in S_{\tilde{\pi}}$. If a free ride is offered, then the user computes her utilities $u(f_{ij}, 0)$ and $u(0, d_{z_j j})$ using α_d, α_f, the customer's willingness to walk and pay parameters. If $u(f_{ij}, 0) \leq u(0, d_{z_j j})$, the customer chooses to accept the free ride to the charging station z_j and we update the ride's end destination to z_j *and* the fare of the ride to \$0, otherwise we leave the ride information the same. We track

that a free ride was offered, whether the customer accepted, and the utility gained in the chosen action. We note that under the 1VMC-50 policy, users also consider the utility of a 50% discounted-ride option by computing $u(0.5 \cdot f_{ij}, d_{z_j j})$. If this option results in the highest utility, then we update the ride's end destination to z_j and the fare paid to $0.5 \cdot f_{ij}$.

(c) Update PE list. First, we remove this arrival request event from the PE list. Next, we add a new arrival request onto the PE list per the process described in Appendix A.5.3.

5. The NE is the completion of an in-progress ride:

   (a) Update the SS table with the vehicle's region, remaining battery, and availability (to *idle*).

   (b) Update PE list. We remove this in-progress ride event from the PE list. We track that a ride was completed and the fare collected by the system.

6. The NE is the completion of an in-progress manual move. We update the SS table with the vehicle's new destination, which is a charging station, and allow the vehicle to be eligible for rides. We track that a manual move occurred in the metrics.

A.6 Experiment Parameter_Sensitivity Description and Results

In our second numerical experiment, Parameter_Sensitivity, we use a synthetic demand scenario where $\lambda_i = p_{ij} = \frac{1}{|\mathcal{R}|}$ and generate 8 parameter instances of an EVSS network by varying four parameters: c_m, b_m, p_m, and $\mathcal{DM}$. Table A.6 describes each instance and the parameters values we vary from the baseline. For each instance, we report the performance of four policies: the optimal free-ride, the single-offer range (1VMC-SOR), the policy where the system operator can choose between offering a free ride, a 50% discounted ride, and not offering at all (1VMC-50), the Fine-Based policy (FB), and the "never offer" policy (NO).

Tables A.7-A.10 show how each policy performed across different metrics. The values presented have been averaged over the last 30 days of the 100-day simulation. Since the network we consider is fairly large, we use the first 70 days as a warm-up period. The results provide a comparative statics analysis of each policy that shows the impact that increasing each of the 4 parameters has on several performance metrics.

Table A.6: Description of Each Instance

Instance	Varying Parameter	c_m	b_m	p_m	$\mathcal{DM}$
1	Base	\$25	0.2	20%	5
2	c_m	\$5	0.2	20%	5
3		\$50			
4	b_m	\$25	0.05	20%	5
5			0.10		
6	p_m	\$25	0.2	5%	5
7				10%	
8	$\mathcal{DM}$	\$25	0.2	20%	0.5
9					20

A.7 Experiment Demand_Sensitivity Description and Results

In our third numerical experiment, Demand_Sensitivity, we vary the demand parameters, λ_i and p_{ij} and fix the operational parameters to their baseline values. We specifically allow the arrival and transition probabilities to be either clustered close (C) to charging stations, spread uniformly (U) across the entire service area, or spread far (F) from charging stations. In total, this gives us 9 synthetic demand scenarios.

Tables A.11-A.13 show how each policy performed in each of the nine demand scenarios across several metrics. The values presented have been averaged over the last 30 days of the 100-day simulation. Since the network we consider is fairly large, we use the first 70 days as

Table A.7: Parameter_Sensitivity Results: Daily Revenue and Rides Fulfilled

		Average Daily Revenue ($)				Rides Fulfilled per Day			
#	Vary	1VMC-SOR	1VMC-50	FB	NO	1VMC-SOR	1VMC-50	FB	NO
1	Base	1,185.43	1,210.28	1,527.35	1,570.66	231.55	217.60	232.12	238.52
2	c_m	1,369.88	1,393.94	1,560.42	1,589.95	238.47	235.90	236.82	241.35
3		1,074.36	1,107.87	1,539.40	1,556.21	217.48	202.70	233.74	236.28
4	b_m	1,261.70	1,273.01	966.02	984.79	248.89	230.11	160.77	163.49
5		1,240.02	1,262.29	1,766.93	1,777.30	244.04	227.79	277.56	278.65
6	p_m	1,236.15	1,259.59	1,544.10	1,024.16	233.86	221.90	234.54	159.63
7		1,243.13	1,280.04	1,531.49	1,577.63	235.47	225.65	232.89	239.95
8	$\mathcal{DM}$	1,267.01	1,358.02	1,519.21	1,538.23	232.54	227.37	230.60	233.23
9		1,194.91	1,236.19	1,548.15	1,558.30	237.87	235.14	234.81	236.33

Table A.8: Parameter_Sensitivity Results: Unmet Demand per Day, due to Vehicle Availability and Insufficient Battery

		Unmet Demand per Day (Vehicle Availability)				Unmet Demand per Day (Insufficient Battery)				Moves per Day			
#	Vary	1VMC -SOR	1VMC -50	FB	NO	1VMC -SOR	1VMC -50	FB	NO	1VMC -SOR	1VMC -50	FB	NO
1	Base	196.8	212.8	189.4	183.2	4.3	2.3	11.1	11.0	4.84	2.95	4.90	47.79
2	c_m	184.8	191.6	185.1	180.3	9.3	5.1	10.7	11.0	18.53	8.72	4.96	48.12
3		212.5	228.6	188.1	185.8	3.1	1.8	11.3	11.0	3.09	2.04	4.86	47.47
4	b_m	141.8	175.8	39.1	39.6	39.0	23.7	229.8	226.6	2.99	1.70	13.16	27.85
5		175.3	196.9	84.3	84.3	12.1	6.8	69.6	68.5	3.00	1.64	11.99	48.21
6	p_m	192.3	207.4	186.7	118.4	6.0	2.8	10.8	154.0	7.05	3.89	4.82	31.32
7		191.7	204.5	189.6	180.0	5.9	2.9	10.6	13.1	7.02	3.82	4.72	47.53
8	$\mathcal{DM}$	191.2	197.7	189.6	187.2	7.5	6.2	11.1	10.9	8.50	5.59	3.58	47.18
9		189.3	192.2	186.2	184.6	4.7	4.4	10.8	10.9	12.05	11.11	9.67	47.40

Table A.9: Parameter_Sensitivity Results: Offers per Day, Accepts per Day, and Average Utility per Offer

		Offers per Day				Accepts Per Day				Average Utilty per Offer			
#	Vary	1VMC -SOR	1VMC -50	FB	NO	1VMC -SOR	1VMC -50	FB	NO	1VMC -SOR	1VMC -50	FB	NO
1	Base	63.70	85.78	51.10	0.00	53.88	62.75	46.20	0.00	-0.69	-2.09	-5.09	0.00
2	c_m	40.21	72.50	51.83	0.00	33.51	51.35	46.87	0.00	-0.67	-2.19	-5.08	0.00
3		65.82	87.28	51.58	0.00	55.99	64.09	46.71	0.00	-0.69	-2.10	-5.06	0.00
4	b_m	68.72	90.43	30.14	0.00	58.03	66.18	16.98	0.00	-0.69	-2.07	-7.84	0.00
5		68.26	89.82	52.41	0.00	57.49	65.95	40.42	0.00	-0.69	-2.07	-6.01	0.00
6	p_m	57.73	81.61	50.84	0.00	48.41	59.08	45.99	0.00	-0.69	-2.12	-5.07	0.00
7		58.34	83.09	50.64	0.00	49.09	60.36	45.91	0.00	-0.68	-2.13	-5.05	0.00
8	$\mathcal{DM}$	45.58	50.50	51.19	0.00	44.83	48.85	47.62	0.00	-0.08	-1.64	-4.49	0.00
9		102.01	154.05	51.58	0.00	54.00	63.36	41.88	0.00	-1.78	-2.49	-6.98	0.00

Table A.10: Parameter_Sensitivity Results: Average Charge of Fleet and Proportion of Fleet at Charging Stations.

		Average Battery (w.o. vehicles at Charging Stations)				Proportion of Fleet at Charging Stations			
#	Vary	1VMC-SOR	1VMC-50	FB	NO	1VMC-SOR	1VMC-50	FB	NO
1	Base	0.55	0.61	0.49	0.44	0.78	0.81	0.76	0.71
2	c_m	0.48	0.54	0.49	0.44	0.74	0.76	0.75	0.71
3		0.58	0.64	0.48	0.44	0.81	0.84	0.75	0.72
4	b_m	0.43	0.52	0.11	0.11	0.67	0.75	0.06	0.06
5		0.52	0.59	0.31	0.30	0.73	0.79	0.42	0.40
6	p_m	0.53	0.60	0.49	0.09	0.76	0.80	0.75	0.11
7		0.53	0.60	0.49	0.44	0.76	0.79	0.75	0.70
8	$\mathcal{DM}$	0.51	0.52	0.49	0.44	0.76	0.77	0.76	0.72
9		0.55	0.57	0.48	0.44	0.76	0.78	0.75	0.72

a warm-up period. In this experiment we only test the 1VMC-SOR policy since the previous experiment indicated the 1VMC-SOR and 1VMC-50 policy perform similarly. We present the results to Demand_Sensitivity in Tables A.11-A.13.

A.8 NVMC-SALP: Description of Basis Functions and Weight Vector

In Table A.14, we provide a description of the 10 basis functions we use for solving NVMC-SALP. Each basis function captures valuable information about the system state and relates to the percentage of the fleet that is available, the geographic dispersion of vehicles, and the average charge of the fleet.

Next, we specify how we sample states and choose the constraint violation budget when solving the NVMC-SALP. Solving the resulting linear program yields a vector of basis function weights, which we provide in Table A.14. It is these weights that we ultimately use to approximate the value function for a multi-vehicle network. Under the NVMC-SALP policy in the simulation, we use the approximate value functions to compute the expected discounted revenue of the $NoOffer$ and $Offer$ actions and choose the revenue-maximizing decision. In this sense, we say that the policy is greedy with respect to the value function approximations.

To generate the NVMC-SALP policy, we sample 500 states. In sampling a single state, we assume that each of the n vehicles are randomly chosen to be either busy or idle. If a vehicle is idle, then it is located in region i with probability λ_i. We assume that each vehicle has a remaining battery that is randomly drawn from $\mathcal{W}$. If a vehicle is busy, we sample the remaining time busy from a triangular distribution with mean 53 minutes and support [1 minutes, 180 minutes], the minimum and maximum trip duration in the data.

The violation budget restricts the expected constraint violation over all 500 states. Per the guidance of Desai, Farias, and Moallemi (2012), we test several budget values ranging

Table A.11: Demand_Sensitivity Results: Daily Revenue and Rides Fulfilled

		Average Daily Revenue ($)			Rides Fulfilled per Day			Moves per Day		
λ_i	p_{ij}	1VMC-SOR	FB	NO	1VMC-SOR	FB	NO	1VMC-SOR	FB	NO
C		1,333.34	1,791.93	1,802.19	264.69	271.90	273.08	5.68	5.82	56.96
U	C	1,168.79	1,529.69	1,552.73	227.61	232.16	235.48	5.07	4.88	47.24
F		721.29	855.68	859.14	129.39	129.73	130.18	5.92	2.43	24.28
C		1,286.68	1,753.53	1,760.54	259.93	266.63	267.64	4.75	5.74	56.05
U	U	1,142.77	1,535.49	1,563.68	226.35	233.10	237.25	4.17	4.88	47.70
F		696.30	845.88	857.82	127.98	128.33	130.17	4.20	2.31	24.18
C		1,229.33	1,767.13	1,797.49	254.52	268.56	273.03	3.40	5.91	56.94
U	F	1,081.99	1,560.62	1,552.91	219.61	237.22	235.93	3.00	4.86	47.58
F		398.11	534.27	550.09	79.53	82.05	84.32	2.18	2.43	19.28

Table A.12: Demand_Sensitivity Results: Unmet Demand per Day (due to Vehicle Availability and Insufficient Battery) and Proportion of Fleet at Charging Stations

		Unmet Demand per Day (Vehicle Availability)			Unmet Demand per Day (Insufficient Battery)			Proportion of Fleet at $\mathcal{Z}$		
λ_i	p_{ij}	1VMC-SOR	FB	NO	1VMC-SOR	FB	NO	1VMC-SOR	FB	NO
C		162.90	147.84	146.84	4.95	12.81	12.62	0.56	0.52	0.47
U	C	199.83	188.76	185.40	4.41	10.93	10.98	0.78	0.75	0.72
F		297.92	295.82	295.36	3.90	5.67	5.69	0.85	0.84	0.83
C		166.88	151.60	150.58	4.21	12.80	12.81	0.58	0.52	0.47
U	U	200.99	187.42	183.11	3.89	10.71	10.88	0.79	0.75	0.71
F		299.76	297.13	294.81	3.14	5.43	5.90	0.85	0.85	0.83
C		174.56	150.85	146.46	3.16	12.83	12.75	0.63	0.54	0.48
U	F	209.63	183.81	185.03	2.88	11.08	11.15	0.80	0.75	0.72
F		349.54	343.91	341.09	1.79	4.89	5.44	0.94	0.94	0.92

Table A.13: Demand_Sensitivity Results: Offers per Day, Accepts per Day, and Average Utility per Offer

		Offers per Day			Accepts Per Day			Average Utilty per Offer		
λ_i	p_{ij}	1VMC -SOR	FB	NO	1VMC -SOR	FB	NO	1VMC -SOR	FB	NO
C		76.66	62.54	0.00	64.81	56.73	0.00	-0.69	-5.06	0.00
U	C	62.18	51.16	0.00	52.38	46.29	0.00	-0.69	-5.08	0.00
F		25.80	26.27	0.00	21.59	23.83	0.00	-0.69	-5.09	0.00
C		78.68	61.41	0.00	66.37	55.66	0.00	-0.69	-5.07	0.00
U	U	64.43	51.37	0.00	54.52	46.49	0.00	-0.69	-5.11	0.00
F		28.28	25.95	0.00	23.80	23.63	0.00	-0.68	-5.05	0.00
C		82.14	61.51	0.00	69.38	55.61	0.00	-0.69	-5.09	0.00
U	F	67.21	52.33	0.00	56.92	47.46	0.00	-0.69	-5.04	0.00
F		23.99	19.25	0.00	19.06	16.80	0.00	-0.90	-5.50	0.00

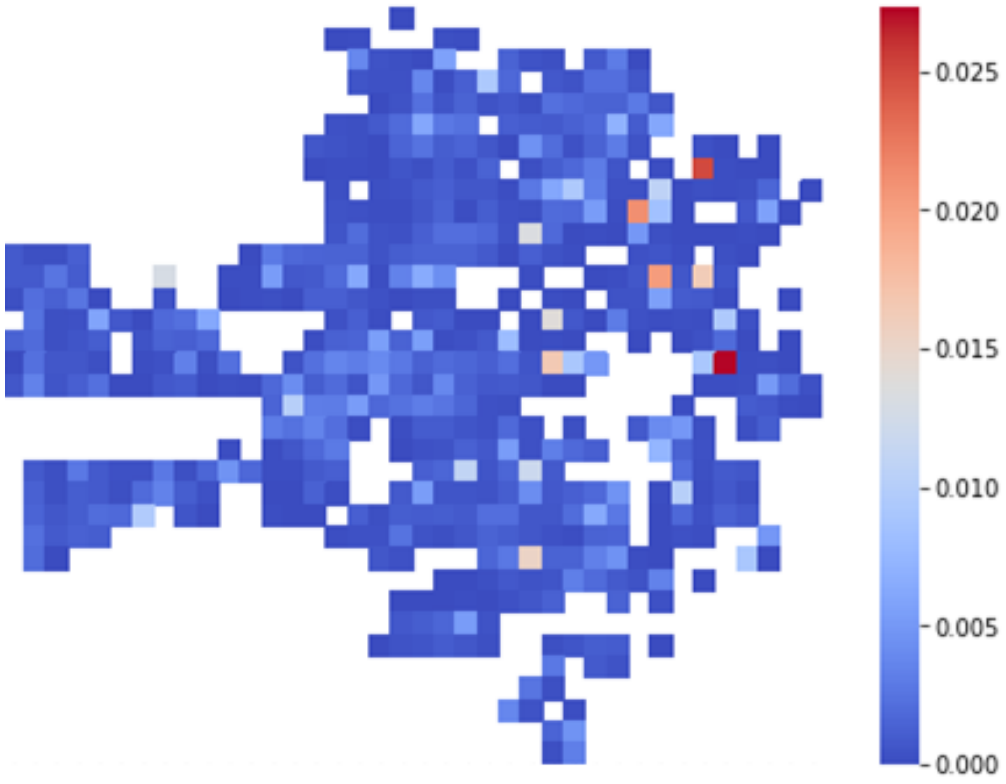
from 0 to 0.05 and then test the performance out of sample and in the simulation. For extreme values, the resulting policy would offer too liberally, resulting in a policy that forgoes too much revenue by offering free rides, and poor dispersion of vehicles since there would be a glut of vehicles at charging stations.

Table A.14: Description of Basis Functions used in NVMC-SALP and the Corresponding Weight Vectors.

Basis Function	Description	**r**
1	% of fleet on trips busy on rides	-1846.83
2	% of busy vehicles available in 1 period	403.41
3	Mean fleet charge (available and non-available vehicles)	5219.46
4	Mean fleet charge (available and non-available vehicles) without vehicles located in $\mathcal{Z}$	-4679.12
5	% of available vehicles with $\lvert\mathcal{R}(i,w)\rvert = 0$	-132.896
6	% of available vehicles with $\lvert\mathcal{R}(i,w)\rvert > 0, w < \bar{b}_i$	408.34
7	% of available vehicles with $\lvert\mathcal{R}(i,w)\rvert > 0, w \geq \bar{b}_i$	639.047
8	% of regions with available vehicles	1729.18
9	% of available vehicles located in $\mathcal{Z}$	188.071
10	% of available vehicles eligible for a move (with battery $< b_m$)	2863.42

A.9 Heatmap of Historical Arrival Probability

Figure A.3: Heatmap of Historical Arrival Probability.



APPENDIX B

Appendix for Chapter 2.1

B.1 Robustness Checks – Estimation with Various Values of Δ

In this section, we present the estimation results for the models described in Eqs. 2.1-2.4 for $\Delta \in \{16, 20, 24, 28, 32, 36, 40, 44\}$, where Δ is in weeks. The results are in Tables B.1-B.8.

We also have the results using the data from customers who transact at least once before and after W_{PC} (i.e., $\Delta = 0$), but we vary the data range in $W_{PC} \pm \Delta'$ for $\Delta' \in \{12, 16, 20, 24, 28, 32, 36, 40, 44\}$, where Δ' is in weeks. The results are in Tables B.9-B.17.

Table B.1: Combined Results: PIDs with $\Delta = 16$ Weeks; Data Range: 16 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	0.22	0.89	0.90	0.20
$Week\,(\beta_{z=41,1})$	2.42^{**}	2.42^{***}	2.42	2.42^{***}
$Week\,(\beta_{z=42,1})$	3.12^{***}	3.12^{***}	3.65^{**}	3.12^{***}
$Week\,(\beta_{z=43,1})$	3.11^{***}	3.11^{***}	4.50^{**}	3.11^{***}
$Week\,(\beta_{z=45,1})$	2.05^{**}	2.05^{**}	2.96	2.05^{**}
$Week\,(\beta_{z=46,1})$	1.26	1.26	0.56	1.26
$Week\,(\beta_{z=48,1})$	3.41^{***}	3.41^{***}	1.23	3.41^{***}
$Week\,(\beta_2)$	-1.54^{***}	-4.17^{***}	-4.17^{***}	
W_{PC} or $Week\,(\beta_3)$				-4.08^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	224	224	224	224
R^2	0.67	0.69	0.70	0.68
Residual Std. Error	34.72	34.05	34.01	34.16
F Statistic	30.80^{***}	30.50^{***}	22.15^{***}	32.30^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 58,462 transactions and 6,406 parkers.			

Table B.2: Combined Results: PIDs with $\Delta = 20$ Weeks; Data Range: 20 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	0.02	0.54	0.90	0.16
$Week\,(\beta_{z=41,1})$	1.62^{**}	1.62^{**}	0.91	1.62^{**}
$Week\,(\beta_{z=42,1})$	2.43^{***}	2.43^{***}	2.30^{*}	2.43^{***}
$Week\,(\beta_{z=43,1})$	2.39^{***}	2.39^{***}	2.79^{**}	2.39^{***}
$Week\,(\beta_{z=45,1})$	1.48^{**}	1.48^{**}	1.30	1.48^{**}
$Week\,(\beta_{z=46,1})$	1.30^{**}	1.30^{**}	1.13	1.30^{**}
$Week\,(\beta_{z=48,1})$	3.19^{***}	3.19^{***}	1.45	3.19^{***}
$Week\,(\beta_2)$	-1.03^{***}	-3.03^{***}	-3.03^{***}	
W_{PC} or $Week\,(\beta_3)$				-2.99^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	280	280	280	280
R^2	0.67	0.69	0.69	0.69
Residual Std. Error	33.02	32.37	32.44	32.40
F Statistic	39.17^{***}	38.82^{***}	27.84^{***}	41.40^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 67,039 transactions and 6,105 parkers.			

Table B.3: Combined Results: PIDs with $\Delta = 24$ Weeks; Data Range: 24 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	0.19	0.58	1.16^{*}	0.41
$Week\,(\beta_{z=41,1})$	1.14^{**}	1.14^{**}	0.68	1.14^{**}
$Week\,(\beta_{z=42,1})$	1.29^{***}	1.29^{***}	0.62	1.29^{***}
$Week\,(\beta_{z=43,1})$	1.50^{***}	1.50^{***}	1.19	1.50^{***}
$Week\,(\beta_{z=45,1})$	0.83^{*}	0.83^{*}	0.56	0.83^{*}
$Week\,(\beta_{z=46,1})$	0.73	0.73	0.43	0.73
$Week\,(\beta_{z=48,1})$	1.63^{***}	1.63^{***}	-0.45	1.63^{***}
$Week\,(\beta_2)$	-0.66^{***}	-2.19^{***}	-2.19^{***}	
W_{PC} or $Week\,(\beta_3)$				-2.17^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	336	336	336	336
R^2	0.65	0.66	0.67	0.66
Residual Std. Error	30.96	30.37	30.20	30.36
F Statistic	41.86^{***}	41.48^{***}	30.43^{***}	44.45^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 74,899 transactions and 5,716 parkers.			

Table B.4: Combined Results: PIDs with $\Delta = 28$ Weeks; Data Range: 28 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	0.18	0.40	1.16^{**}	0.23
$Week\,(\beta_{z=41,1})$	0.95^{***}	0.95^{***}	0.12	0.95^{***}
$Week\,(\beta_{z=42,1})$	1.03^{***}	1.03^{***}	0.18	1.03^{***}
$Week\,(\beta_{z=43,1})$	1.57^{***}	1.57^{***}	1.10^{*}	1.57^{***}
$Week\,(\beta_{z=45,1})$	0.69^{**}	0.69^{**}	0.11	0.69^{**}
$Week\,(\beta_{z=46,1})$	0.64^{*}	0.64^{**}	0.08	0.64^{**}
$Week\,(\beta_{z=48,1})$	1.05^{***}	1.05^{***}	-0.93	1.05^{***}
$Week\,(\beta_2)$	-0.49^{***}	-1.39^{***}	-1.39^{***}	
W_{PC} or $Week\,(\beta_3)$				-1.37^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	392	392	392	392
R^2	0.66	0.67	0.68	0.67
Residual Std. Error	28.20	27.92	27.62	27.93
F Statistic	52.20^{***}	50.30^{***}	37.39^{***}	53.76^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 80,839 transactions and 5,205 parkers.			

Table B.5: Combined Results: PIDs with $\Delta = 32$ Weeks; Data Range: 32 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	0.55^{**}	0.69^{***}	1.35^{***}	0.47^{**}
$Week\,(\beta_{z=41,1})$	0.45^{*}	0.45^{*}	-0.23	0.45^{*}
$Week\,(\beta_{z=42,1})$	0.57^{**}	0.57^{**}	-0.12	0.57^{**}
$Week\,(\beta_{z=43,1})$	1.22^{***}	1.22^{***}	0.58	1.22^{***}
$Week\,(\beta_{z=45,1})$	0.40	0.40	0.01	0.40
$Week\,(\beta_{z=46,1})$	0.23	0.23	-0.35	0.23
$Week\,(\beta_{z=48,1})$	0.61^{**}	0.61^{**}	-1.03^{**}	0.61^{**}
$Week\,(\beta_2)$	-0.46^{***}	-1.04^{***}	-1.04^{***}	
W_{PC} or $Week\,(\beta_3)$				-1.02^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	448	448	448	448
R^2	0.69	0.69	0.70	0.69
Residual Std. Error	25.42	25.25	24.93	25.33
F Statistic	67.48^{***}	64.25^{***}	47.93^{***}	68.12^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 84,335 transactions and 4,594 parkers.			

Table B.6: Combined Results: PIDs with $\Delta = 36$ Weeks; Data Range: 36 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	0.62^{***}	0.82^{***}	1.45^{***}	0.66^{***}
$Week\,(\beta_{z=41,1})$	0.45^{**}	0.45^{**}	-0.13	0.45^{**}
$Week\,(\beta_{z=42,1})$	0.45^{**}	0.45^{**}	-0.17	0.45^{**}
$Week\,(\beta_{z=43,1})$	0.97^{***}	0.97^{***}	0.30	0.97^{***}
$Week\,(\beta_{z=45,1})$	0.32	0.32	-0.20	0.32
$Week\,(\beta_{z=46,1})$	0.17	0.17	-0.39	0.17
$Week\,(\beta_{z=48,1})$	0.34^{*}	0.34^{*}	-1.14^{***}	0.34^{*}
$Week\,(\beta_2)$	-0.45^{***}	-1.25^{***}	-1.25^{***}	
W_{PC} or $Week\,(\beta_3)$				-1.24^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	504	504	504	504
R^2	0.68	0.70	0.71	0.69
Residual Std. Error	24.68	24.22	23.86	24.27
F Statistic	75.35^{***}	74.35^{***}	55.74^{***}	79.14^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 84,187 transactions and 3,964 parkers.			

Table B.7: Combined Results: PIDs with $\Delta = 40$ Weeks; Data Range: 40 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	0.54^{***}	0.76^{***}	1.19^{***}	0.70^{***}
$Week\,(\beta_{z=41,1})$	0.34^{**}	0.34^{**}	-0.06	0.34^{**}
$Week\,(\beta_{z=42,1})$	0.32^{*}	0.32^{*}	-0.14	0.32^{*}
$Week\,(\beta_{z=43,1})$	0.71^{***}	0.71^{***}	0.15	0.71^{***}
$Week\,(\beta_{z=45,1})$	0.08	0.08	-0.21	0.08
$Week\,(\beta_{z=46,1})$	0.12	0.12	-0.12	0.12
$Week\,(\beta_{z=48,1})$	0.24	0.24	-0.84^{***}	0.24
$Week\,(\beta_2)$	-0.34^{***}	-1.20^{***}	-1.20^{***}	
W_{PC} or $Week\,(\beta_3)$				-1.20^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	560	560	560	560
R^2	0.63	0.65	0.66	0.65
Residual Std. Error	24.38	23.69	23.45	23.68
F Statistic	65.25^{***}	66.66^{***}	49.47^{***}	71.45^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 80,764 transactions and 3,188 parkers.			

Table B.8: Combined Results: PIDs with $\Delta = 44$ Weeks; Data Range: 44 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	0.61^{***}	0.70^{***}	1.01^{***}	0.65^{***}
$Week\,(\beta_{z=41,1})$	-0.02	-0.02	-0.27	-0.02
$Week\,(\beta_{z=42,1})$	0.04	0.04	-0.30	0.04
$Week\,(\beta_{z=43,1})$	0.33^{***}	0.33^{***}	-0.13	0.33^{***}
$Week\,(\beta_{z=45,1})$	-0.14	-0.14	-0.37	-0.14
$Week\,(\beta_{z=46,1})$	-0.15	-0.15	-0.31	-0.15
$Week\,(\beta_{z=48,1})$	-0.07	-0.07	-0.78^{***}	-0.07
$Week\,(\beta_2)$	-0.18^{***}	-0.57^{***}	-0.57^{***}	
W_{PC} or $Week\,(\beta_3)$				-0.57^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	616	616	616	616
R^2	0.66	0.67	0.67	0.67
Residual Std. Error	19.73	19.54	19.38	19.54
F Statistic	82.94^{***}	79.78^{***}	58.68^{***}	85.44^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 70,405 transactions and 2,143 parkers.			

Table B.9: Combined Results: PIDs with $\Delta = 0$ Weeks; Data Range: 12 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	2.12	4.07^{***}	2.72	2.64^{**}
$Week\,(\beta_{z=41,1})$	3.14^{**}	3.14^{**}	4.72^{*}	3.14^{**}
$Week\,(\beta_{z=42,1})$	5.10^{***}	5.10^{***}	7.52^{***}	5.10^{***}
$Week\,(\beta_{z=43,1})$	3.56^{**}	3.56^{***}	7.42^{***}	3.56^{***}
$Week\,(\beta_{z=45,1})$	2.27	2.27^{*}	3.59	2.27^{*}
$Week\,(\beta_{z=46,1})$	2.11	2.11	2.40	2.11
$Week\,(\beta_{z=48,1})$	5.79^{***}	5.79^{***}	5.76^{**}	5.79^{***}
$Week\,(\beta_2)$	-3.55^{***}	-11.04^{***}	-11.04^{***}	
W_{PC} or $Week\,(\beta_3)$				-10.82^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	168	168	168	168
R^2	0.80	0.84	0.85	0.84
Residual Std. Error	35.50	32.04	32.20	32.41
F Statistic	44.44^{***}	53.28^{***}	37.91^{***}	55.49^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 60,624 transactions and 9,371 parkers.			

Table B.10: Combined Results: PIDs with $\Delta = 0$ Weeks; Data Range: 16 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	3.74^{***}	5.80^{***}	5.77^{***}	4.66^{***}
$Week\,(\beta_{z=41,1})$	1.96	1.96^{*}	2.10	1.96^{*}
$Week\,(\beta_{z=42,1})$	3.50^{***}	3.50^{***}	3.80^{*}	3.50^{***}
$Week\,(\beta_{z=43,1})$	2.94^{**}	2.94^{***}	4.76^{**}	2.94^{**}
$Week\,(\beta_{z=45,1})$	1.80	1.80	2.39	1.80
$Week\,(\beta_{z=46,1})$	0.66	0.66	-0.14	0.66
$Week\,(\beta_{z=48,1})$	3.24^{**}	3.24^{***}	1.38	3.24^{***}
$Week\,(\beta_2)$	-3.54^{***}	-11.51^{***}	-11.51^{***}	
W_{PC} or $Week\,(\beta_3)$				-11.38^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	224	224	224	224
R^2	0.68	0.75	0.76	0.75
Residual Std. Error	46.95	41.67	41.87	41.99
F Statistic	31.94^{***}	41.66^{***}	29.67^{***}	43.66^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 77,846 transactions and 10,266 parkers.			

Table B.11: Combined Results: PIDs with $\Delta = 0$ Weeks; Data Range: 20 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	3.65^{***}	5.42^{***}	6.00^{***}	4.64^{***}
$Week\,(\beta_{z=41,1})$	0.89	0.89	0.10	0.89
$Week\,(\beta_{z=42,1})$	2.33^{**}	2.33^{***}	1.50	2.33^{***}
$Week\,(\beta_{z=43,1})$	2.23^{**}	2.23^{***}	2.76^{*}	2.23^{***}
$Week\,(\beta_{z=45,1})$	0.65	0.65	0.10	0.65
$Week\,(\beta_{z=46,1})$	0.47	0.47	-0.25	0.47
$Week\,(\beta_{z=48,1})$	2.74^{***}	2.74^{***}	1.05	2.74^{***}
$Week\,(\beta_2)$	-2.91^{***}	-9.81^{***}	-9.81^{***}	
W_{PC} or $Week\,(\beta_3)$				-9.74^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	280	280	280	280
R^2	0.66	0.74	0.74	0.74
Residual Std. Error	47.88	41.82	42.06	42.04
F Statistic	36.69^{***}	50.47^{***}	35.77^{***}	53.21^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 95,085 transactions and 10,864 parkers.			

Table B.12: Combined Results: PIDs with $\Delta = 0$ Weeks; Data Range: 24 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	3.29^{***}	4.94^{***}	5.88^{***}	4.59^{***}
$Week\,(\beta_{z=41,1})$	0.61	0.61	-0.18	0.61
$Week\,(\beta_{z=42,1})$	1.37^{*}	1.37^{**}	-0.03	1.37^{**}
$Week\,(\beta_{z=43,1})$	1.40^{*}	1.40^{**}	1.10	1.40^{**}
$Week\,(\beta_{z=45,1})$	0.27	0.27	-0.46	0.27
$Week\,(\beta_{z=46,1})$	0.14	0.14	-0.69	0.14
$Week\,(\beta_{z=48,1})$	1.09	1.09^{*}	-1.48	1.09^{*}
$Week\,(\beta_2)$	-2.33^{***}	-8.81^{***}	-8.81^{***}	
W_{PC} or $Week\,(\beta_3)$				-8.78^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	336	336	336	336
R^2	0.62	0.73	0.73	0.72
Residual Std. Error	49.55	42.02	41.93	42.04
F Statistic	37.09^{***}	56.56^{***}	40.93^{***}	60.44^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 111,562 transactions and 11,307 parkers.			

Table B.13: Combined Results: PIDs with $\Delta = 0$ Weeks; Data Range: 28 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	3.11^{***}	4.50^{***}	5.46^{***}	4.26^{***}
$Week\,(\beta_{z=41,1})$	0.24	0.24	-0.66	0.24
$Week\,(\beta_{z=42,1})$	0.92	0.92^{*}	-0.46	0.92^{*}
$Week\,(\beta_{z=43,1})$	1.42^{**}	1.42^{***}	1.23	1.42^{***}
$Week\,(\beta_{z=45,1})$	-0.03	-0.03	-0.81	-0.03
$Week\,(\beta_{z=46,1})$	-0.25	-0.25	-1.19	-0.25
$Week\,(\beta_{z=48,1})$	0.35	0.35	-2.16^{**}	0.35
$Week\,(\beta_2)$	-1.94^{***}	-7.41^{***}	-7.41^{***}	
W_{PC} or $Week\,(\beta_3)$				-7.39^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	392	392	392	392
R^2	0.62	0.73	0.74	0.73
Residual Std. Error	48.32	40.85	40.51	40.85
F Statistic	43.94^{***}	67.51^{***}	49.60^{***}	72.22^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 128,175 transactions and 11,613 parkers.			

Table B.14: Combined Results: PIDs with $\Delta = 0$ Weeks; Data Range: 32 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	3.06^{***}	4.35^{***}	5.13^{***}	4.13^{***}
$Week\,(\beta_{z=41,1})$	0.21	0.21	-0.49	0.21
$Week\,(\beta_{z=42,1})$	0.72	0.72^{*}	-0.50	0.72^{*}
$Week\,(\beta_{z=43,1})$	1.07^{**}	1.07^{***}	0.63	1.07^{***}
$Week\,(\beta_{z=45,1})$	0.15	0.15	-0.27	0.15
$Week\,(\beta_{z=46,1})$	-0.21	-0.21	-0.90	-0.21
$Week\,(\beta_{z=48,1})$	0.28	0.28	-1.69^{**}	0.28
$Week\,(\beta_2)$	-1.82^{***}	-6.91^{***}	-6.91^{***}	
W_{PC} or $Week\,(\beta_3)$				-6.89^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	448	448	448	448
R^2	0.61	0.74	0.75	0.74
Residual Std. Error	48.34	39.81	39.56	39.84
F Statistic	49.28^{***}	81.53^{***}	59.54^{***}	87.15^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 143,842 transactions and 11,890 parkers.			

Table B.15: Combined Results: PIDs with $\Delta = 0$ Weeks; Data Range: 36 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	2.92^{***}	4.27^{***}	5.00^{***}	4.20^{***}
$Week\,(\beta_{z=41,1})$	0.15	0.15	-0.45	0.15
$Week\,(\beta_{z=42,1})$	0.46	0.46	-0.74	0.46
$Week\,(\beta_{z=43,1})$	0.87^{**}	0.87^{***}	0.35	0.87^{***}
$Week\,(\beta_{z=45,1})$	0.13	0.13	-0.24	0.13
$Week\,(\beta_{z=46,1})$	-0.30	-0.30	-0.92	-0.30
$Week\,(\beta_{z=48,1})$	0.04	0.04	-1.74^{***}	0.04
$Week\,(\beta_2)$	-1.71^{***}	-7.04^{***}	-7.04^{***}	
W_{PC} or $Week\,(\beta_3)$				-7.03^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	504	504	504	504
R^2	0.57	0.74	0.74	0.74
Residual Std. Error	51.83	40.57	40.28	40.54
F Statistic	45.83^{***}	90.45^{***}	66.18^{***}	97.06^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 158,142 transactions and 12,068 parkers.			

Table B.16: Combined Results: PIDs with $\Delta = 0$ Weeks; Data Range: 40 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	2.70^{***}	4.11^{***}	4.73^{***}	4.18^{***}
$Week\,(\beta_{z=41,1})$	0.25	0.25	-0.16	0.25
$Week\,(\beta_{z=42,1})$	0.31	0.31	-0.80	0.31
$Week\,(\beta_{z=43,1})$	0.68^{*}	0.68^{**}	0.08	0.68^{**}
$Week\,(\beta_{z=45,1})$	0.02	0.02	-0.37	0.02
$Week\,(\beta_{z=46,1})$	-0.15	-0.15	-0.51	-0.15
$Week\,(\beta_{z=48,1})$	0.04	0.04	-1.41^{**}	0.04
$Week\,(\beta_2)$	-1.63^{***}	-7.19^{***}	-7.19^{***}	
W_{PC} or $Week\,(\beta_3)$				-7.19^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	560	560	560	560
R^2	0.51	0.72	0.73	0.72
Residual Std. Error	56.67	42.55	42.33	42.52
F Statistic	40.00^{***}	94.40^{***}	68.69^{***}	101.26^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01 Underlying data has 171,541 transactions and 12,250 parkers.			

Table B.17: Combined Results: PIDs with $\Delta = 0$ Weeks; Data Range: 44 Weeks Pre/Post Price Change

	Dependent Variable: $Y_{z,Week}$			
	Basic	Two Intercepts	Two Intercepts with $INT_{z,2}$	Hinge
$Week\,(\beta_1)$	2.52^{***}	3.85^{***}	4.39^{***}	3.96^{***}
$Week\,(\beta_{z=41,1})$	0.27	0.27	-0.05	0.27
$Week\,(\beta_{z=42,1})$	0.19	0.19	-0.84^{*}	0.19
$Week\,(\beta_{z=43,1})$	0.53	0.53^{**}	-0.11	0.53^{**}
$Week\,(\beta_{z=45,1})$	0.06	0.06	-0.23	0.06
$Week\,(\beta_{z=46,1})$	-0.08	-0.08	-0.33	-0.08
$Week\,(\beta_{z=48,1})$	-0.05	-0.05	-1.34^{***}	-0.05
$Week\,(\beta_2)$	-1.50^{***}	-6.76^{***}	-6.76^{***}	
W_{PC} or $Week\,(\beta_3)$				-6.77^{***}
Zone FE (INT_z)	Yes	Yes	Yes	Yes
Zone Coefficient ($\beta_{z,1}$)	Yes	Yes	Yes	Yes
Zone FE ($INT_{z,2}$)	–	–	Yes	–
Observations	616	616	616	616
R^2	0.49	0.72	0.73	0.72
Residual Std. Error	57.77	42.71	42.44	42.70
F Statistic	40.75^{***}	102.91^{***}	75.08^{***}	110.25^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01			
	Underlying data has 185,303 transactions and 12,352 parkers.			

APPENDIX C

Appendix for Chapter 2.2

This appendix is divided into three sections. In Section C.1 we provide the Proof to Lemma 2.2.1. In Section C.2, we provide additional descriptive statistics on our dataset, especially for the slices of the data that appear in the robustness checks. In Section C.3 we present a series of tables with the results to our robustness checks described in Section 2.2.4.4 of the main paper.

C.1 Proof of Lemma 2.2.1

Let $\mathcal{D}$ represent the set of during-treatment transactions used in the MLE estimation. From $\mathcal{D}$, we can compute the binary indicator S^i_{jt} that takes value 1 if customer i parked in zone j during transaction t, and we can compute $m^i_j = \sum_t S^i_{jt}$ and $m^i = \sum_t \sum_j S^i_{jt}$, respectively the number of transactions by i in zone k and the total number of transactions by i.

First, we refer to the LC-MNL likelihood and log-likelihood functions in Eqs. (2.11)-(2.12). For simplicity in this section, we define

$$\boldsymbol{q} = (q_{11|1}, ..., q_{jt|k}, ..., q_{ZT|Z}), \boldsymbol{\lambda} = (\lambda^1_1, ..., \lambda^i_k, ..., \lambda^I_Z), Z = |\mathcal{Z}|, I = |\mathcal{I}|, T = \sum_i |\mathcal{T}_i|.$$

We note that β is embedded in conditional choice probability $q_{jt|k}$ and that the log-likelihood function can be re-written as Eq. (C.1), where, for ease of exposition, we drop the index t

and replace S^i_{jt}.

$$\hat{l}(\boldsymbol{q}, \boldsymbol{\lambda}|\mathcal{D}) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{Z}} m^i_j \cdot \ln\Big(\sum_k \lambda^i_k \cdot q_{j|k}\Big) \tag{C.1}$$

To find MLE estimates, we need solve $\max_{\boldsymbol{q},\boldsymbol{\lambda} \geq 0} \left\{\hat{l}(\boldsymbol{q}, \boldsymbol{\lambda}|\mathcal{D})\right\}$ subject to two constraints: $\sum_k \lambda^i_k = 1 \; \forall \, i$ and $\sum_j q_{j|k} = 1 \; \forall \, i, k$. To find a local maxima, we formulate the associated Lagrangian function $\mathbf{L}(\boldsymbol{q}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \boldsymbol{\mu})$ in Eq. (C.2), where $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_i, ...\alpha_I)$ and $\boldsymbol{\mu} = (\mu_1, ..., \mu_k, ...\mu_Z)$ are the multipliers associated with each constraint. Then we can derive the corresponding KKT conditions. We provide the first-order, stationarity conditions in Eqs. (C.3)-(C.4). The remaining conditions are as follows: $\sum_k \lambda^i_k - 1 = 0 \; \forall \, i$ and $\sum_j q_{j|k} - 1 = 0 \; \forall \, k$ (Primal Feasibility); $\alpha_i \geq 0 \; \forall \, i$ and $\mu_k \geq 0 \; \forall \, k$ (Dual Feasibility); $\sum_i \alpha_i \cdot \Big(\sum_k \lambda^i_k - 1\Big) = 0$ and $\sum_k \mu_k \cdot \Big(\sum_j q_{j|k} - 1\Big) = 0$ (Complementary Slackness).

$$\begin{aligned}\mathbf{L}(\boldsymbol{q}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \sum_i \sum_j m^i_j \cdot \log\Big(\sum_k \lambda^i_k \cdot q_{j|k}\Big) \\ - \sum_i \alpha_i \cdot \Big(\sum_k \lambda^i_k - 1\Big) - \sum_k \mu_k \cdot \Big(\sum_j q_{j|k} - 1\Big)\end{aligned} \tag{C.2}$$

$$\frac{\partial}{\partial q_{j|k}} \mathbf{L}(\boldsymbol{q}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \sum_i \sum_j m^i_j \cdot \frac{1}{\sum_z \lambda^i_z \cdot q_{j|z}} \cdot \lambda^i_k - \mu_k \qquad = 0 \qquad \forall \; j, k \tag{C.3}$$

$$\frac{\partial}{\partial \lambda^i_k} \mathbf{L}(\boldsymbol{q}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \sum_j m^i_j \cdot \frac{1}{\sum_z \lambda^i_z \cdot q_{j|z}} \cdot q_{j|k} - \alpha_i \qquad = 0 \qquad \forall \; i, k \tag{C.4}$$

Next, we observe that when we set $\lambda^i_k = \frac{m^i_k}{m^i}$ and $q_{j|k} = 1$ if $j = k$ and $q_{j|k} = 0$ if $j \neq k$, and we use multipliers with the value $\alpha_i = m_i \; \forall \; i$ and $\mu_k = Z \cdot \sum_i m^i_k \; \forall \; k$, the KKT conditions are satisfied. Therefore, this point is a feasible MLE estimate. If we compute the determinant of the Hessian at this point the determinant is equal to 0. However, as $\beta \to -\infty$, the determinant of the Hessian is approaching zero from below, which shows

that the function is concave at the point and the point does not occur at a saddle point or inflection point, but at a local maxima.

Finally, we show that when $\beta \to -\infty$, $q_{j|k}$ naturally takes 1 if $j = k$ and 0 if $j \neq k$, which proves that as $\beta \to -\infty$ and $\lambda_k^i = \frac{m_k^i}{m^i}$ are feasible MLE estimates. Recall from Eq. (2.8) that $q_{j|k} = \frac{\exp(v_{j|k})}{\sum_{j' \in \mathcal{Z}} \exp(v_{j'k})}$. We begin by observing a key property of the utility function $v_{j|k}$: as $\beta \to -\infty$, $\exp(v_{j|k}) \to 0$ if $j \neq k$ and $\exp(v_{j|k}) = \exp(-LOS \cdot rate_j)$ if $j = k$. This is because $d_{jk} > 0$ if $j \neq k$ and $d_{jk} = 0$ if $j = k$. These observations imply that, as $\beta \to -\infty$, $q_{j|k} \approx 0$ if $j \neq k$ and $q_{j|k} \approx 1$ if $j = k$. With these values of $q_{j|k}$, we saw in the previous paragraph that the LC-MNL likehood function attains a local maxima when the class probabilities λ_k^i equal the exact proportion of transactions by i in zone k in the estimation dataset $\mathcal{D}$. □

C.2 Data Pre-Processing and Descriptive Statistics

For each neighborhood radius, ranging from 0.25 miles, which we use in the main paper, down to 0.10 miles, which we use as a robustness check of our results, Table C.1 contains the number of transactions that occur during the morning hours, and when precipitation is present, at the daily level, and within 30 or 60 minutes. The bottom section of Table C.1 reports this same breakdown but for the larger subset of the population, which is also used as a robustness check. Furthermore, for each neighborhood size, Table C.2 contains a summary of the number of zones and the number of unique parkers in the dataset.

C.3 Robustness Checks

This section describes the results to all of the robustness checks described in Section 2.2.4.4. Table C.3 serves as a "Table of Contents" with a description of each robustness check, relative to the original, baseline case that we present in the main paper. Each row of the table represents a different configuration or change that we make to test as a robustness check.

Table C.1: Data Description: Number of Transactions as a Function of Neighborhood Size

Population Size	Neighborhood Radius (Miles)	Description	Number of Transactions	Percentage of Dataset
Original	0.25	Total Transactions	6,477	100%
		Precipitation: Daily	2,234	34.5%
		Precipitation: ± 30 Minutes	413	6.4%
		Precipitation: ± 60 Minutes	499	7.7%
		Morning: First 60 Minutes	976	15.1%
		Morning: First 120 Minutes	1,810	27.9%
	0.20	Total Transactions	5,777	100%
		Precipitation: Daily	2,005	34.7%
		Precipitation: ± 30 Minutes	367	6.4%
		Precipitation: ± 60 Minutes	444	7.7%
		Morning: First 60 Minutes	825	14.3%
		Morning: First 120 Minutes	1,600	27.7%
	0.15	Total Transactions	4,047	100%
		Precipitation: Daily	1,385	34.2%
		Precipitation: ± 30 Minutes	256	6.3%
		Precipitation: ± 60 Minutes	306	7.6%
		Morning: First 60 Minutes	551	13.6%
		Morning: First 120 Minutes	1,082	26.7%
	0.10	Total Transactions	2,718	100%
		Precipitation: Daily	943	34.7%
		Precipitation: ± 30 Minutes	178	6.5%
		Precipitation: ± 60 Minutes	217	8.0%
		Morning: First 60 Minutes	308	11.3%
		Morning: First 120 Minutes	697	25.6%
Large Population	0.25	Total Transactions	29,012	100%
		Precipitation: Daily	10,109	34.8%
		Precipitation: ± 30 Minutes	1,971	6.8%
		Precipitation: ± 60 Minutes	2,333	8.0%
		Morning: First 60 Minutes	4,406	15.2%
		Morning: First 120 Minutes	7,339	25.3%

Table C.2: Data Description: Neighborhood Size

Neighborhood Radius (Miles)	Number of Zones	Number of Zones at Original Hourly Rate of $3.75	Number of Transac-tions	Number of Parkers
0.25	43	24	6,477	180
0.20	29	16	5,777	177
0.15	17	11	4,047	173
0.10	9	4	2,718	160

Any entry that contains ✓means the value takes the same value as the baseline configuration, which is described in detail in the second row in the table. The second column of Table C.3 contains the Table number that houses the results of the corresponding estimation.

Table C.3: Tables for the Estimation Outputs Associated with Each Robustness Check

Difference from Baseline	Table	Distance	Population	Intended Destination Probability	Catchment Area Radius	Re-Ups
Baseline	2.9	Haversine	Baseline	Bayesian Smoothing	0.25-Mile	None
Count-Based Intended Destination Probability	C.4	✓	✓	Count-Based	✓	✓
Keep Learning Transaction	C.5	✓	✓	Keep Learning Transaction	✓	✓
0.20-Mile Neighborhood Size	C.6	✓	✓	✓	0.20-Mile	✓
0.15-Mile Neighborhood Size	C.7	✓	✓	✓	0.15-Mile	✓
0.10-Mile Neighborhood Size	C.8	✓	✓	✓	0.10-Mile	✓
Google Walking API Distance	C.9	Google Walking API	✓	✓	✓	✓
Manhattan Distance	C.10	Manhattan	✓	✓	✓	✓
Less Restrictive Population	C.11	✓	Less Restrictive	✓	✓	✓
Combine 30-Minute Re-Up Transactions	C.12	✓	✓	✓	✓	30 Minutes
Rate-Only Utility	C.13	Haversine	Baseline	Bayesian Smoothing	0.25-Mile	None

Table C.4: Count-Based Mixing Distribution – Regression Results

	Utility Specification								
	Baseline	Rain Effect			Morning Effect		Relaxing Perfect Foresight		
		± 30 Min.	± 60 Min.	Day	60 Min.	120 Min.	LOS-II	LOS-III	LOS-IV
β	-51.76^{***}	-51.48^{***}	-51.45^{***}	-50.95^{***}	-49.97^{***}	-49.62^{***}	-30.04^{***}	-33.00^{***}	-30.10^{***}
	(2.89)	(2.60)	(3.51)	(3.64)	(3.55)	(3.61)	(1.71)	(1.82)	(2.02)
β_{rain}		-4.63^{***}	-4.31^{***}	-2.43^{***}					
		(1.39)	(0.95)	(0.79)					
β_{AM}					-11.22^{***}	-7.79^{***}			
					(1.45)	(0.92)			
β_{rate}								1.34^{***}	0.16^{**}
								(0.03)	(0.08)
β_{LOS}							-0.13^{***}		-0.23^{***}
							(0.02)		(0.03)
Num PIDS	180	180	180	180	180	180	180	180	180
Num Txns	6477	6477	6477	6477	6477	6477	6477	6477	6477
Log Likelihood	-21179.7	-21176.5	-21176.4	-21176.2	-21136.1	-21147.7	-18519.6	-18768.7	-18515.4
AIC	42361.4	42356.9	42356.8	42356.4	42276.1	42299.4	37043.3	37541.4	37036.8
BIC	42368.2	42370.5	42370.3	42369.9	42289.7	42312	37056.8	37555	37057.1
Note:									*p<0.1; **p<0.05; ***p<0.01

Table C.5: Keeping "Learning Transaction" – Regression Results

	Utility Specification								
	Baseline	Rain Effect			Morning Effect		Relaxing Perfect Foresight		
		± 30 Min.	± 60 Min.	Day	60 Min.	120 Min.	LOS-II	LOS-III	LOS-IV
β	-87.12^{***}	-86.55^{***}	-87.10^{***}	-85.46^{***}	-82.66^{***}	-83.26^{***}	-41.02^{***}	-45.80^{***}	-41.11^{***}
	(8.94)	(8.05)	(7.46)	(8.07)	(7.89)	(7.90)	(3.29)	(3.95)	(3.77)
β_{rain}		-15.15^{***}	-0.24	-5.08^{***}					
		(4.55)	(2.64)	(1.21)					
β_{AM}					-27.67^{***}	-12.73^{**}			
					(5.13)	(6.37)			
β_{rate}								1.35^{***}	0.28^{***}
								(0.04)	(0.10)
β_{LOS}							-0.09^{**}		-0.26^{***}
							(0.04)		(0.05)
Num PIDS	180	180	180	180	180	180	180	180	180
Num Txns	6477	6477	6477	6477	6477	6477	6477	6477	6477
Log Likelihood	-19002	-19000.5	-19002	-19000.8	-18980.4	-18994.5	-18052.3	-18151.5	-18046.9
AIC	38006.1	38005.1	38008.1	38005.6	37964.9	37993	36108.6	36307.1	36099.8
BIC	38012.8	38018.6	38021.6	38019.1	37978.4	38006.6	36122.1	36320.6	36120.1
Note:									*p<0.1; **p<0.05; ***p<0.01

Table C.6: Catchment Area 0.20 Miles – Estimation Results

	Utility Specification								
	Baseline	Rain Effect			Morning Effect		Relaxing Perfect Foresight		
		± 30 Min.	± 60 Min.	Day	60 Min.	120 Min.	LOS-II	LOS-III	LOS-IV
β	-70.57^{***}	-70.10^{***}	-70.29^{***}	-70.19^{***}	-66.39^{***}	-67.62^{***}	-42.42^{***}	-45.90^{***}	-42.42^{***}
	(6.61)	(5.97)	(6.23)	(6.95)	(5.47)	(7.30)	(4.00)	(3.79)	(4.22)
β_{rain}		-7.72^{***}	-3.83^{***}	-1.22^{*}					
		(1.42)	(1.36)	(0.66)					
β_{AM}					-38.61^{***}	-10.52^{***}			
					(2.77)	(1.97)			
β_{rate}								1.01^{***}	0.02
								(0.03)	(0.12)
β_{LOS}							-0.36^{***}		-0.37^{***}
							(0.01)		(0.04)
Num PIDS	177	177	177	177	177	177	177	177	177
Num Txns	5777	5777	5777	5777	5777	5777	5777	5777	5777
Log Likelihood	-14692.5	-14691.4	-14692.2	-14692.4	-14648.6	-14684.9	-14343.8	-14389.4	-14343.8
AIC	29387	29386.8	29388.3	29388.8	29301.3	29373.9	28691.7	28782.8	28693.7
BIC	29393.7	29400.1	29401.6	29402.1	29314.6	29387.2	28705	28796.1	28713.6
Note:									*p<0.1; **p<0.05; ***p<0.01

Table C.7: Catchment Area 0.15 Miles – Regression Results

	Utility Specification								
	Baseline	Rain Effect			Morning Effect		Relaxing Perfect Foresight		
		± 30 Min.	± 60 Min.	Day	60 Min.	120 Min.	LOS-II	LOS-III	LOS-IV
β	-77.17^{***}	-77.03^{***}	-77.27^{***}	-76.22^{***}	-72.45^{***}	-74.25^{***}	-42.28^{***}	-47.55^{***}	-42.30^{***}
	(4.71)	(4.25)	(4.05)	(3.23)	(4.00)	(3.49)	(3.05)	(2.65)	(3.61)
β_{rain}		-2.78^{**}	1.41	-2.98^{***}					
		(1.35)	(1.01)	(0.68)					
β_{AM}					-35.30^{***}	-9.24^{***}			
					(2.77)	(2.11)			
β_{rate}								1.07^{***}	0.05
								(0.03)	(0.17)
β_{LOS}							-0.29^{***}		-0.32^{***}
							(0.01)		(0.05)
Num PIDS	173	173	173	173	173	173	173	173	173
Num Txns	4047	4047	4047	4047	4047	4047	4047	4047	4047
Log Likelihood	-8910	-8909.9	-8909.9	-8909.4	-8873.2	-8904.3	-8632.3	-8677.1	-8632.2
AIC	17821.9	17823.7	17823.9	17822.8	17750.5	17812.6	17268.6	17358.3	17270.4
BIC	17828.3	17836.3	17836.5	17835.4	17763.1	17825.2	17281.2	17370.9	17289.3
Note:									*p<0.1; **p<0.05; ***p<0.01

Table C.8: Catchment Area 0.10 Miles – Regression Results

	Utility Specification								
	Baseline	Rain Effect			Morning Effect		Relaxing Perfect Foresight		
		± 30 Min.	± 60 Min.	Day	60 Min.	120 Min.	LOS-II	LOS-III	LOS-IV
β	-83.85^{***}	-83.85^{***}	-84.02^{***}	-82.39^{***}	-79.04^{***}	-81.47^{***}	-44.18^{***}	-53.15^{***}	-44.16^{***}
	(3.83)	(3.22)	(3.71)	(3.67)	(3.58)	(3.70)	(3.08)	(3.12)	(4.33)
β_{rain}		0.04	2.89^{***}	-4.55^{***}					
		(1.26)	(0.94)	(0.67)					
β_{AM}					-39.44^{***}	-7.26^{***}			
					(3.37)	(1.91)			
β_{rate}								0.99^{***}	-0.05
								(0.02)	(0.17)
β_{LOS}							-0.31^{***}		-0.28^{***}
							(0.02)		(0.05)
Num PIDS	160	160	160	160	160	160	160	160	160
Num Txns	2718	2718	2718	2718	2718	2718	2718	2718	2718
Log Likelihood	-4504.4	-4504.4	-4504.3	-4503.2	-4471.2	-4501.4	-4383	-4414.3	-4382.9
AIC	9010.8	9012.8	9012.5	9010.5	8946.5	9006.8	8769.9	8832.6	8771.8
BIC	9016.7	9024.6	9024.3	9022.3	8958.3	9018.7	8781.7	8844.4	8789.5
Note:									*p<0.1; **p<0.05; ***p<0.01

Table C.9: Google Walking Distance – Regression Results

	Utility Specification								
	Baseline	Rain Effect			Morning Effect		Relaxing Perfect Foresight		
		± 30 Min.	± 60 Min.	Day	60 Min.	120 Min.	LOS-II	LOS-III	LOS-IV
β	-64.81^{***}	-64.48^{***}	-64.75^{***}	-64.18^{***}	-62.29^{***}	-63.10^{***}	-29.28^{***}	-34.00^{***}	-29.32^{***}
	(4.38)	(4.21)	(3.94)	(4.75)	(3.65)	(4.00)	(2.11)	(2.60)	(2.43)
β_{rain}		-7.44^{**}	-0.86	-2.02^{***}					
		(3.26)	(2.22)	(0.75)					
β_{AM}					-17.42^{***}	-5.46^{***}			
					(2.94)	(1.73)			
β_{rate}								1.28^{***}	0.20^{***}
								(0.01)	(0.07)
β_{LOS}		-0.15^{***}		-0.26^{***}					
		(0.01)		(0.03)					
Num PIDS	180	180	180	180	180	180	180	180	180
Num Txns	6477	6477	6477	6477	6477	6477	6477	6477	6477
Log Likelihood	-18933.6	-18932.1	-18933.6	-18933	-18911.9	-18929	-17972	-18107	-17967.9
AIC	37869.2	37868.2	37871.1	37869.9	37827.9	37862	35948	36217.9	35941.8
BIC	37876	37881.8	37884.7	37883.5	37841.4	37875.5	35961.5	36231.5	35962.1
Note:									*p<0.1; **p<0.05; ***p<0.01

Table C.10: Manhattan Distance – Regression Results

	Utility Specification								
	Baseline	Rain Effect			Morning Effect		Relaxing Perfect Foresight		
		± 30 Min.	± 60 Min.	Day	60 Min.	120 Min.	LOS-II	LOS-III	LOS-IV
β	-64.40^{***}	-64.12^{***}	-64.38^{***}	-63.50^{***}	-61.82^{***}	-62.41^{***}	-31.70^{***}	-35.57^{***}	-31.73^{***}
	(4.05)	(3.66)	(3.90)	(4.46)	(3.29)	(3.40)	(2.42)	(2.48)	(2.70)
β_{rain}		-5.89	-0.38	-2.83^{***}					
		(3.63)	(1.92)	(0.75)					
β_{AM}					-25.96^{***}	-7.43^{***}			
					(4.68)	(2.58)			
β_{rate}								1.24^{***}	0.17^{**}
								(0.01)	(0.07)
β_{LOS}		-0.18^{***}		-0.28^{***}					
		(0.01)		(0.03)					
Num PIDS	180	180	180	180	180	180	180	180	180
Num Txns	6477	6477	6477	6477	6477	6477	6477	6477	6477
Log Likelihood	-19001.6	-19000.8	-19001.6	-19000.6	-18973.4	-18995.5	-18091	-18210.7	-18088.6
AIC	38005.2	38005.5	38007.2	38005.2	37950.8	37995	36186.1	36425.3	36183.1
BIC	38012	38019.1	38020.8	38018.8	37964.3	38008.6	36199.6	36438.9	36203.5
Note:									*p<0.1; **p<0.05; ***p<0.01

Table C.11: Larger Subset of the Population – Regression Results

	Utility Specification								
	Baseline	Rain Effect			Morning Effect		Relaxing Perfect Foresight		
		± 30 Min.	± 60 Min.	Day	60 Min.	120 Min.	LOS-II	LOS-III	LOS-IV
β	-64.39^{***}	-64.39^{***}	-64.49^{***}	-65.10^{***}	-61.68^{***}	-61.88^{***}	-29.32^{***}	-33.07^{***}	-29.66^{***}
	(2.05)	(2.47)	(2.50)	(2.81)	(2.33)	(2.11)	(1.63)	(1.42)	(1.57)
β_{rain}		0.05	1.18	1.99^{***}					
		(0.85)	(0.72)	(0.38)					
β_{AM}					-41.77^{***}	-11.77^{***}			
					(3.59)	(1.46)			
β_{rate}								1.25^{***}	-0.29^{***}
								(0.02)	(0.08)
β_{LOS}							-0.23^{***}		-0.07^{***}
							(0.02)		(0.02)
Num PIDS	2003	2003	2003	2003	2003	2003	2003	2003	2003
Num Txns	29012	29012	29012	29012	29012	29012	29012	29012	29012
Log Likelihood	-92874.9	-92874.9	-92874.3	-92869.9	-92520.1	-92759	-86614	-88010.1	-86552.9
AIC	185751.9	185753.8	185752.7	185743.8	185044.1	185522	173231.9	176024.1	173111.7
BIC	185760.1	185770.4	185769.2	185760.3	185060.7	185538.6	173248.5	176040.7	173136.6

Note: *p<0.1; **p<0.05; ***p<0.01

Table C.12: Combining "30-Minute Re-Up" Transactions – Regression Results

	Utility Specification								
	Baseline	Rain Effect			Morning Effect		Relaxing Perfect Foresight		
		± 30 Min.	± 60 Min.	Day	60 Min.	120 Min.	LOS-II	LOS-III	LOS-IV
β	-89.33^{***}	-90.16^{***}	-90.80^{***}	-86.63^{***}	-83.26^{***}	-82.24^{***}	-39.04^{***}	-43.20^{***}	-39.06^{***}
	(6.42)	(5.63)	(6.17)	(6.69)	(6.05)	(5.25)	(2.94)	(3.64)	(3.09)
β_{rain}		11.66^{***}	17.53^{***}	-8.94^{***}					
		(1.40)	(3.23)	(2.13)					
β_{AM}					-34.10^{***}	-19.72^{***}			
					(4.39)	(4.33)			
β_{rate}								1.46^{***}	0.35^{***}
								(0.02)	(0.09)
β_{LOS}							-0.13^{***}		-0.32^{***}
							(0.02)		(0.03)
Num PIDS	179	179	179	179	179	179	179	179	179
Num Txns	5416	5416	5416	5416	5416	5416	5416	5416	5416
Log Likelihood	-16240.3	-16238.2	-16234.2	-16237.1	-16211.6	-16222.8	-15309.7	-15407	-15301.7
AIC	32482.5	32480.4	32472.5	32478.2	32427.2	32449.6	30623.3	30817.9	30609.4
BIC	32489.1	32493.6	32485.7	32491.4	32440.4	32462.8	30636.5	30831.1	30629.2
Note:									*p<0.1; **p<0.05; ***p<0.01

Table C.13: Rate-Only Utility Specification – Regression Results: Baseline, Rain, and Morning

	Utility Specification					
	Baseline	With Rain			With Morning	
		Day	± 30 Min.	± 60 Min.	60 Min.	120 Min.
β	-52.10^{***}	-51.65^{***}	-51.88^{***}	-51.94^{***}	-50.90^{***}	-51.56^{***}
	(2.69)	(3.49)	(3.44)	(3.66)	(3.46)	(3.32)
β_{rain}		-1.33^{***}	-4.32^{***}	-2.54^{**}		
		(0.38)	(1.35)	(1.04)		
β_{AM}					-9.88^{***}	-2.03^{***}
					(1.38)	(0.78)
Num PIDS	180	180	180	180	180	180
Num Txns	6477	6477	6477	6477	6477	6477
Log Likelihood	-18432.6	-18432.2	-18431.7	-18432.2	-18423.4	-18431.8
AIC	36867.2	36868.4	36867.4	36868.4	36850.8	36867.6
BIC	36874	36881.9	36881	36882	36864.4	36881.1
Note:						*p<0.1; **p<0.05; ***p<0.01

APPENDIX D

Appendix for Chapter 3.1

D.1 Properties of f

Property D.1.1. *Let $X \in \mathcal{M}$ and $X' = f(X)$.*

1. *(Affine Structure) Each column in X' has the form: $X'_j = C_j + \sum_{l=1}^{n} b_{j,l} \cdot X_l$, where $C_j \in \mathbb{R}_+^n$ and $b_{j,l} \in \mathbb{R}_+$ are polynomial functions of $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_n]$.*

D.2 Proof of Lemma 3.3.1.

For all $X, Y \in \mathcal{M}$, we first define the distance metric $\Delta = d(X, Y) = \max_{i,j}\{|X_{ij} - Y_{ij}|\} = ||X - Y||_\infty$. We note that $(\mathcal{M}, d)$ is a non-empty complete metric space. Equivalently, $\Delta = \max_j\{\Delta_j\}$ where $\Delta_j = d(X_j, Y_j)$. Let $X' = f(X), Y' = f(Y)$, and $\Delta' = d(X', Y')$. To show that f is a contraction on $\mathcal{M}$, we show that for all $X, Y \in \mathcal{M}, \exists\, 0 \leq q < 1$ s.t. $\Delta' \leq q \cdot \Delta$. Since $\Delta' = \max_j\{\Delta'_j\}$, we will show that for all $j, \exists\, 0 \leq q < 1$ s.t. $\Delta'_j \leq q \cdot \Delta$.

The proof has four steps: (1) we upper bound Δ'_j by a constant times Δ. (2) we show this constant is a function of $\boldsymbol{\alpha}$. In (3) and (4), we show this constant is always strictly less than 1, proving f is a contraction.

(1) Upperbound on Δ'_j.

For an arbitrary j, we can find an upperbound on Δ'_j as follows:

$$\begin{aligned}
\Delta'_j = \max_i |X'_{ij} - Y'_{ij}| &= \max_i |C_{ij} + \sum_{l=1}^{n} b_{j,l} \cdot X_{il} - C_{ij} - \sum_{l=1}^{n} b_{j,l} \cdot Y_{il}| \\
&= \max_i |\sum_{l=1}^{n} b_{j,l} \cdot (X_{il} - Y_{il})| \\
&\leq \max_i \sum_{l=1}^{n} |b_{j,l}| \cdot |X_{il} - Y_{il}| \\
&\leq \sum_{l=1}^{n} |b_{j,l}| \cdot \max_i |X_{il} - Y_{il}| \\
&= \sum_{l=1}^{n} b_{j,l} \cdot \Delta_l \\
&\leq \sum_{l=1}^{n} b_{j,l} \cdot \Delta
\end{aligned}$$

The second and last equality in the above equation come from Property D.1.1. This shows that $\Delta'_j \leq \sum_{l=1}^{n} b_{j,l} \cdot \Delta$, where $\sum_{l=1}^{n} b_{j,l} \in \mathbb{R}_+$. If $\sum_{l=1}^{n} b_{j,l} < 1$ for all j, then f is a contraction.

(2) Expressing $\sum_{l=1}^{n} b_{j,l}$ as a Function of $\boldsymbol{\alpha}$.

For the general case when $n \geq 5$, the columns in the output matrix $X' = f(X)$ for an arbitrary $X \in \mathcal{M}$ fall into one of five categories based on the column number j: left columns ($j = 1, 2$), right columns ($j = n-1, n$), and interior columns ($2 < j < n-1$). Using Eq. (3.3) and f, we derive the expressions for X'_j in terms of $\boldsymbol{\alpha}$ and present them in Table D.1. We also define the function $H_j = \sum_{l=1}^{n} b_{j,l}$ as a function of $\boldsymbol{\alpha}$. We note when $n = 2$, we use columns X'_1 and X'_n. When $n = 3$, we use columns X'_1, X'_{n-1}, X'_n. For $n = 4$, we use columns $X'_1, X'_2, X'_{n-1}, X'_n$.

Table D.1: Expressions for X'_j and $H_j(\boldsymbol{\alpha})$

j	X'_j	$H_j(\boldsymbol{\alpha})$
1	$\alpha_1 \cdot e_1 + \frac{1}{2}(1-\alpha_1)\alpha_2 \cdot e_2 + \frac{1}{2}(1-\alpha_1)\alpha_2 \cdot X_1 + \frac{1}{3}(1-\alpha_1)(1-\alpha_2) \cdot X_3$	$\frac{1}{2}(1-\alpha_1)\alpha_2 + \frac{1}{3}(1-\alpha_1)(1-\alpha_2)$
2	$\alpha_1\alpha_2 \cdot e_1 + \frac{2}{3}(1-\alpha_2)\alpha_3 \cdot e_3 + \left(\frac{1}{2}(1-\alpha_1)\alpha_2 + \frac{2}{3}(1-\alpha_2)\alpha_3\right) \cdot X_2 + \frac{1}{2}(1-\alpha_2)(1-\alpha_3) \cdot X_4$	$\frac{1}{2}(1-\alpha_1)\alpha_2 + \frac{2}{3}(1-\alpha_2)\alpha_3 + \frac{1}{2}(1-\alpha_2)(1-\alpha_3)$
$2 < j < n-1$	$\alpha_j\alpha_{j-1} \cdot e_{j-1} + \left(\frac{j}{j+1}(1-\alpha_j)\alpha_{j+1} + \alpha_j\right) \cdot e_j + \alpha_j\alpha_{j-1} \cdot X_{j-2} + \left(\frac{j-1}{j}(1-\alpha_{j-1})\alpha_j + \frac{j}{j+1}(1-\alpha_j)\alpha_{j+1}\right) \cdot X_j + \frac{j}{j+2}(1-\alpha_j)(1-\alpha_{j+1}) \cdot X_{j+2}$	$\alpha_j\alpha_{j-1} + \frac{j-1}{j}(1-\alpha_{j-1})\alpha_j + \frac{j}{j+1}(1-\alpha_j)\alpha_{j+1} + \frac{j}{j+2}(1-\alpha_j)(1-\alpha_{j+1})$
$n-1$	$\alpha_{n-2}\alpha_{n-1} \cdot e_{n-2} + \alpha_{n-1} \cdot e_{n-1} + \frac{n-1}{n}(1-\alpha_{n-1}) \cdot e_n + \alpha_{n-2}\alpha_{n-1} \cdot X_{n-3} + \left(\frac{n-2}{n-1}(1-\alpha_{n-2})\alpha_{n-1} + \frac{n-1}{n}(1-\alpha_{n-1})\right) \cdot X_{n-1}$	$\alpha_{n-2}\alpha_{n-1} + \frac{n-2}{n-1}(1-\alpha_{n-2})\alpha_{n-1} + \frac{n-1}{n}(1-\alpha_{n-1})$
n	$\alpha_{n-1} \cdot e_{n-1} + e_n + \alpha_{n-1} \cdot X_{n-2} + \frac{n-1}{n}(1-\alpha_{n-1}) \cdot X_n$	$\alpha_{n-1} + \frac{n-1}{n}(1-\alpha_{n-1})$

Note: To illustrate how we derive X'_j, consider the case $j = 1$: Let $g(X)_j$ be the value of the j-th column after one application of g. With this, $X'_1 = f(X)_1 = g(g(X))_j$. By Eq. (3.3), $X'_1 = \alpha_1(e_1) + \frac{1-\alpha_1}{2}g(X)_2$, and $g(X)_2 = \alpha_2(X_1 + e_2) + (1-\alpha_2)\frac{2}{3}X_3$. Combining this, we get $X'_1 = \alpha_1(e_1) + \frac{1-\alpha_1}{2}(\alpha_2(X_1 + e_2) + (1-\alpha_2)\frac{2}{3}X_3)$, the expression above.

(3) Showing Maximum Value of H_j is Strictly Less Than 1.

For the five cases described in Table D.1, we want to show $\max_{\boldsymbol{\alpha}\in[0,1]^n:0<\alpha_k<1}\{H_j(\boldsymbol{\alpha})\} < 1$. We refer to this as Problem A and denote z_A its optimal value and $\boldsymbol{\alpha}_A^*$ its maximizer. However, problem A's feasible region is not a compact set due to the strict inequalities on $\boldsymbol{\alpha}$. Relaxing the strict inequalities in A, we have $\max_{\boldsymbol{\alpha}\in[0,1]^n:0\leq\alpha_k\leq1}\{H_j(\boldsymbol{\alpha})\}$, which we refer to problem B. We define z_B and $\boldsymbol{\alpha}_B^*$ similarly. Since A's feasible region is strictly contained within B's, we have $z_A \leq z_B$, and since B's feasible region is compact, we can solve for z_B and $\boldsymbol{\alpha}_B^*$. If we show that $z_B < 1$ or $z_A < z_B = 1$ for all j, then we have shown f is a contraction. Below, we compute z_B and α_B^* for each case.

$j = 1$**:** $z_B = \max_{0\leq\alpha_1,\alpha_2\leq1}\left\{H_1(\alpha_1,\alpha_2)\right\} = \frac{1}{2}$ since $\frac{\partial H_1}{\partial\alpha_2} = \frac{1-\alpha_1}{6} \geq 0$, so $\alpha_B^* = (0,1)$ and $z_B = \frac{1}{2}$.

$j = 2$**:** $z_B = \max_{0\leq\alpha_1,\alpha_2,\alpha_3\leq1}\left\{H_2(\alpha_1,\alpha_2,\alpha_3)\right\} = \frac{2}{3}$ since $\frac{\partial H_2'}{\partial\alpha_1} = \frac{-\alpha_2}{2} \leq 0$, $\frac{\partial H_2'}{\partial\alpha_2} = \frac{-\alpha_1}{2} - \frac{\alpha_3}{6} \leq 0$, and $\frac{\partial H_2'}{\partial\alpha_3} = \frac{1-\alpha_2}{6} \geq 0$. Therefore, $\alpha_B^* = (0,0,1)$ and $z_B = \frac{2}{3}$.

$2 < j < n-1$**:** $z_B = \max_{0\leq\alpha_{j-1},\alpha_j,\alpha_{j+1}\leq1}\left\{H_j(\alpha_{j-1},\alpha_j,\alpha_{j+1})\right\} = 1$ since $\frac{\partial H_j}{\partial\alpha_{j-1}} = \frac{\alpha_j}{j} \geq 0$ and $\frac{\partial H_j}{\partial\alpha_{j+1}} = \frac{j(1-\alpha_j)}{(j+1)(j+2)} \geq 0$, so $\alpha_{j-1,B}^* = \alpha_{j+1,B}^* = 1$. Then $\frac{\partial H_j}{\partial\alpha_j} = \frac{j-2}{j(j+2)} + \frac{\alpha_{j-1}}{j} - \frac{\alpha_{j+1}\cdot j}{(j+2)(j+1)} > 0$, so $\alpha_B^* = (1,1,1)$ and $z_B = 1$.

$j = n-1$**:** $z_B = \max_{0\leq\alpha_{n-2},\alpha_{n-1}\leq1}\left\{H_{n-1}(\alpha_{n-2},\alpha_{n-1})\right\} = 1$ since $\frac{\partial H_{n-1}}{\partial\alpha_{n-2}} = \frac{1}{n}\alpha_{n-1} \geq 0$, which implies $\alpha_{n-2,B}^* = 1$. When $\alpha_{n-2} > \frac{1}{n}$, we have $\frac{\partial H_{n-1}}{\partial\alpha_{n-1}} = \frac{\alpha_{n-2}}{n-1} - \frac{1}{n(n-1)} > 0$. Therefore, $\alpha^* = (1,1)$ and $z_B = 1$.

$j = n$**:** $z_B = \max_{0\leq\alpha_{n-1}\leq1}\left\{H_n(\alpha_{n-1})\right\} = 1$ since $\frac{\partial H_n(\alpha_{n-1})}{\partial\alpha_{n-1}} = \frac{1}{n} \geq 0$, so $\alpha_{n-1,B}^* = 1$ and $z_B = 1$.

(4) Final Step.

For $j = 1, 2$, $z_B < 1$, so we are done for those cases. However, for the cases $j > 2$, $z_B = 1$. All of the solutions in these cases are corner solutions where the elements of $\boldsymbol{\alpha}_B^*$ take either 0 or 1. Since problem A's feasible region is strictly contained in B's, the corner solution $\boldsymbol{\alpha}_B^*$ cannot be reached in A. Therefore, $z_A < z_B = 1$ for all $j \in \{1, ..., n\}$ and f is a contraction. □

D.3 Proof of Theorem 3.3.1.

From Lemma 3.3.1 and the Banach fixed-point theorem, we know there exists a unique $Y^* \in \mathcal{M}$ such that $Y^* = f(Y^*)$ and $d(Y^*, f(Y^*)) = ||Y^* - f(Y^*)||_\infty = 0$. We also know $d(Y, Y') \geq d(Y', Y'') \geq ... > 0$, where $Y^{(k)} = f^{\circ k}(X)$ for $k \in \mathbb{Z}_+$. For each application of f, or as $k \to \infty$, the distance d successively decreases. Therefore as $\epsilon \to 0$, the algorithm converges to Y^*.

Next, we show $Y^* = \Omega$. Assume that $Y^* \neq \Omega$. Embedded within f is g, which uses steady-state transition probabilities to model the flow between states, so we know $g(\Omega) = \Omega$, which means that $f(\Omega) = g(g(\Omega)) = \Omega$. If $Y^* \neq \Omega$, then $f(Y^*) \neq Y^*$ and $d(Y^*, f(Y^*)) > 0$, which is impossible since Y^* is the fixed point of the contraction mapping f. Therefore $Y^* = \Omega$ and the algorithm converges to the correct Ω as $\epsilon \to 0$. □

D.4 Proof of Theorem 3.3.2.

If d_1 is the distance after the first iteration of the algorithm and q is the number of subsequent iterations until the distance reaches the stopping tolerance ϵ, then the algorithm terminates when $d_1 \cdot A^q < \epsilon$, where $A = \max_{j=1,...,n}\{H_j(\boldsymbol{\alpha})\} \in (0, 1)$. Note that $H_j(\boldsymbol{\alpha})$ (see Table D.1) is the upper bound on the rate at which the distance decreases in successive iterations. Rearranging the expression, we have $q < \log_A(\epsilon/d_1)$, so the algorithm terminates in, at most, $\log_A(\epsilon/d_1) + 1$ iterations. When the input matrix is $Y = [Y_1, ..., Y_j, ..., Y_n] \in \mathcal{M}$, where

$Y_j = \frac{j}{n} \cdot \mathbf{1}$, then the distance $d_1 = d(Y, Y') \to 1$ as $n \to \infty$. Applying this insight, we know the algorithm terminates in at most $\log_A(\epsilon) + 1$ iterations. $\square$

Bibliography

Aguiar, Luis and Joel Waldfogel (2018), "Platforms, Promotion, and Product Discovery: Evidence from Spotify Playlists," Technical report, National Bureau of Economic Research.

Albert, Gila and David Mahalel (2006), "Congestion Tolls and Parking Fees: A Comparison of the Potential Effect on Travel Behavior," *Transport Policy*, 13 (6), 496–502.

Albuquerque, Paulo and Bart J Bronnenberg (2012), "Measuring the Impact of Negative Demand Shocks on Car Dealer Networks," *Marketing Science*, 31 (1), 4–23.

Allcott, Hunt and Todd Rogers (2014), "The Short-Run and Long-Run Effects of Behavioral Interventions: Experimental Evidence from Energy Conservation," *American Economic Review*, 104 (10), 3003–37.

Allen, Arnold O (2014), *Probability, Statistics, and Queueing Theory*, Academic Press.

Amazon (2020), "Prime Now," URL `https://primenow.amazon.com/`.

Arnott, Richard and John Rowse (1999), "Modeling Parking," *Journal of urban economics*, 45 (1), 97–124.

Avci, Buket, Karan Girotra, and Serguei Netessine (2014), "Electric Vehicles with a Battery Switching Station: Adoption and Environmental Impact," *Management Science*, 61 (4), 772–794.

Balachander, Subramanian and Sanjoy Ghose (2003), "Reciprocal Spillover Effects: A Strategic Benefit of Brand Extensions," *Journal of Marketing*, 67 (1), 4–13.

Banerjee, Siddhartha, Daniel Freund, and Thodoris Lykouris (2016), "Pricing and Optimization in Shared Vehicle Systems: An Approximation Framework," *CoRR abs/1608.06819*.

Barrowman, Nicholas J and Ransom A Myers (2000), "Still More Spawner-Recruitment Curves: The Hockey Stick and its Generalizations," *Canadian Journal of Fisheries and Aquatic Sciences*, 57 (4), 665–676.

BBC (2020), "Why we have a love-hate relationship with electric scooters," URL `https://www.bbc.com/future/article/20200608-how-sustainable-are-electric-scooters`.

Beath, Ken J (2017), "randomLCA: An R Package for Latent Class with Random Effects Analysis," *Journal of Statistical Software*, 81 (13), 1–25.

Bellos, Ioannis, Mark Ferguson, and L Beril Toktay (2017), "The car sharing economy: Interaction of business model choice and product line design," *Manufacturing & Service Operations Management*, 19 (2), 185–201.

Besbes, Omar, Adam N Elmachtoub, and Yunjie Sun (2021), "Static Pricing: Universal Guarantees for Reusable Resources," *Operations Research*.

Boyacı, Burak, Konstantinos G Zografos, and Nikolas Geroliminis (2015), "An Optimization Framework for the Development of Efficient One-way Car-sharing Systems," *European Journal of Operational Research*, 240 (3), 718–733.

Brandon, Alec, Paul J Ferraro, John A List, Robert D Metcalfe, Michael K Price, and Florian Rundhammer (2017), "Do the Effects of Social Nudges Persist? Theory and Evidence from 38 Natural Field Experiments," Technical report, National Bureau of Economic Research.

Brandstätter, Georg, Michael Kahr, and Markus Leitner (2017), "Determining Optimal Locations for Charging Stations of Electric Car-sharing Systems Under Stochastic Demand," *Transportation Research Part B: Methodological*, 104, 17–35.

Bront, Juan José Miranda, Isabel Méndez-Díaz, and Gustavo Vulcano (2009), "A Column Generation Algorithm for Choice-Based Network Revenue Management," *Operations Research*, 57 (3), 769–784.

Browning, Raymond C, Emily A Baker, Jessica A Herron, and Rodger Kram (2006), "Effects of obesity and sex on the energetic cost and preferred speed of walking," *Journal of Applied Physiology*, 100 (2), 390–398.

Bruglieri, Maurizio, Al berto Colorni, and Alessandro Luè (2014), "The Relocation Problem for the One-way Electric Vehicle Sharing," *Networks*, 64 (4), 292–305.

Brumelle, Shelby L (1978), "A Generalization of Erlang's Loss System to State Dependent Arrival and Service Rates," *Mathematics of Operations Research*, 3 (1), 10–16.

Bucklin, Randolph E, Sivaramakrishnan Siddarth, and Jorge M Silva-Risso (2008), "Distribution Intensity and New Car Choice," *Journal of Marketing Research*, 45 (4), 473–486.

Burman, David Y (1981), "Insensitivity in Queueing Systems," *Advances in Applied Probability*, 846–859.

Chang, Joy, Miao Yu, Siqian Shen, and Ming Xu (2017), "Location Design and Relocation of a Mixed Car-sharing Fleet with a CO2 Emission Constraint," *Service Science*, 9 (3), 205–218.

Chemla, Daniel, Frédéric Meunier, Thomas Pradeau, Roberto Wolfler Calvo, and Houssame Yahiaoui (2013), "Self-service bike sharing systems: simulation, repositioning, pricing," *Working Paper*.

Christensen, Peter and Adam Osman (2021), "The Demand for Mobility: Evidence from an Experiment with Uber Riders," *Available at SSRN 3803723*.

Chung, Hangil, Daniel Freund, and David B Shmoys (2018), "Bike Angels: An Analysis of Citi Bike's Incentive Program," in "Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies," 1–9.

CNN (2017), "Cities warm up to designated Uber, Lyft pick-up spots," URL `https://money.cnn.com/2017/11/16/technology/uber-lyft-designated-pickup-spots/index.html`.

Cohen, Maxime, Michael-David Fiszer, and Baek Jung Kim (2018), "Frustration-Based Promotions: Field Experiments in Ride-sharing," *Available at SSRN 3129717*.

Croissant, Yves (2020), "Estimation of Random Utility Models in R: The mlogit Package," *Journal of Statistical Software*, 95 (1), 1–41.

DellaVigna, Stefano and Ethan Kaplan (2007), "The Fox News Effect: Media Bias and Voting," *The Quarterly Journal of Economics*, 122 (3), 1187–1234.

Dell'Olio, Luigi, Angel Ibeas, Juan de Ona, and Rocio de Ona (2017), *Public Transportation Quality of Service: Factors, Models, and Applications*, Elsevier.

Desai, Vijay V, Vivek F Farias, and Ciamac C Moallemi (2012), "Approximate Dynamic Programming via a Smoothed Linear Program," *Operations Research*, 60 (3), 655–674.

Deutsch, Yael and Boaz Golany (2018), "A Parcel Locker Network as a Solution to the Logistics Last Mile Problem," *International Journal of Production Research*, 56 (1-2), 251–261.

Efron, Bradley and Robert Tibshirani (1986), "Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy," *Statistical Science*, 54–75.

Fabusuyi, Tayo and Robert C Hampshire (2018), "Rethinking Performance Based Parking Pricing: A Case Study of SFpark," *Transportation Research Part A: Policy and Practice*, 115, 90–101.

Feldman, Jacob and Huseyin Topaloglu (2015), "Bounding Optimal Expected Revenues for Assortment Optimization Under Mixtures of Multinomial Logits," *Production and Operations Management*, 24 (10), 1598–1620.

Feldman, Pnina, Jun Li, and Hsin-Tien Tsai (2020), "Welfare Implications of Congestion Pricing: Evidence from SFpark," *Manufacturing & Service Operations Management*.

Fielbaum, Andres, Xiaoshan Bai, and Javier Alonso-Mora (2021), "On-demand Ridesharing with Optimized Pick-up and Drop-off Walking Locations," *Transportation Research Part C: Emerging Technologies*, 126, 103061.

Fisher, Marshall, Santiago Gallino, and Jun Li (2018), "Competition-Based Dynamic Pricing in Online Retailing: A Methodology Validated with Field Experiments," *Management Science*, 64 (6), 2496–2514.

Fishman, Elliot, Simon Washington, and Narelle Haworth (2014), "Bike Share's Impact on Car Use: Evidence from the United States, Great Britain, and Australia," *Transportation Research Part D: Transport and Environment*, 31, 13–20.

Freund, Daniel, Shane G Henderson, and David B Shmoys (2017), "Minimizing Multimodular Functions and Allocating Capacity in Bike-sharing Systems," in "International Conference on Integer Programming and Combinatorial Optimization," Springer, 186–198.

Frey, Erin and Todd Rogers (2014), "Persistence: How Treatment Effects Persist After Interventions Stop," *Policy Insights from the Behavioral and Brain Sciences*, 1 (1), 172–179.

Fricker, Christine and Nicolas Gast (2014), "Incentives and Redistribution in Homogeneous Bike-sharing Systems with Stations of Finite Capacity," *EURO Journal on Transportation and Logistics*, 1–31.

Gillen, David W (1978), "Parking Policy, Parking Location Decisions and the Distribution of Congestion," *Transportation*, 7 (1), 69–85.

Hampshire, Robert C and Donald Shoup (2018), "What Share of Traffic is Cruising for Parking?" *Journal of Transport Economics and Policy (JTEP)*, 52 (3), 184–201.

He, Long, Ho-Yin Mak, Ying Rong, and Zuo-Jun Max Shen (2017), "Service Region Design for Urban Electric Vehicle Sharing Systems," *Manufacturing & Service Operations Management*, 19 (2), 309–327.

Hensher, David A and William H Greene (2003), "The Mixed Logit Model: The State of Practice," *Transportation*, 30 (2), 133–176.

Hess, Stephane, Moshe Ben-Akiva, Dinesh Gopinath, and Joan Walker (2011), "Advantages of Latent Class over Continuous Mixture of Logit Models," *Institute for Transport Studies, University of Leeds. Working paper*.

Hudson, Derek J (1966), "Fitting Segmented Curves Whose Join Points Have to be Estimated," *Journal of the American Statistical Association*, 61 (316), 1097–1129.

Hui, Sam K, J Jeffrey Inman, Yanliu Huang, and Jacob Suher (2013), "The Effect of In-store Travel Distance on Unplanned Spending: Applications to Mobile Promotion Strategies," *Journal of Marketing*, 77 (2), 1–16.

Imbens, Guido W and Thomas Lemieux (2008), "Regression Discontinuity Designs: A Guide to Practice," *Journal of Econometrics*, 142 (2), 615–635.

Jorge, Diana and Gonçalo Correia (2013), "Carsharing Systems Demand Estimation and Defined Operations: A Literature Review," *European Journal of Transport and Infrastructure Research*, 13 (3), 201–220.

Kabra, Ashish, Elena Belavina, and Karan Girotra (2020), "Bike-Share Systems: Accessibility and Availability," *Management Science*, 66 (9), 3803–3824.

Kamakura, Wagner A and Gary J Russell (1989), "A Probabilistic Choice Model for Market Segmentation and Elasticity Structure," *Journal of marketing research*, 26 (4), 379–390.

Kanafani, Adib and Lawrence H Lan (1988), "Development of Pricing Strategies for Airport Parking—A Case Study at San Francisco Airport," *International Journal of Transport Economics/Rivista internazionale di economia dei trasporti*, 55–76.

Kelly, Frank P (2011), *Reversibility and Stochastic Networks*, Cambridge University Press.

Kelly, J Andrew and J Peter Clinch (2009), "Temporal Variance of Revealed Preference On-street Parking Price Elasticity," *Transport Policy*, 16 (4), 193–199.

Lehner, Stephan and Stefanie Peer (2019), "The Price Elasticity of Parking: A Meta-Analysis," *Transportation Research Part A: Policy and Practice*, 121, 177–191.

Lei, Yanzhe and Stefanus Jasin (2020), "Real-Time Dynamic Pricing for Revenue Management with Reusable Resources, Advance Reservation, and Deterministic Service Time Requirements," *Operations Research*, 68 (3), 676–685.

Li, Hongmin, Scott Webster, Nicholas Mason, and Karl Kempf (2019), "Product-Line Pricing Under Discrete Mixed Multinomial Logit Demand: Winner—2017 M&SOM Practice-Based Research Competition," *Manufacturing & Service Operations Management*, 21 (1), 14–28.

Li, Jun, Nelson Granados, and Serguei Netessine (2014), "Are Consumers Strategic? Structural Estimation from the Air-Travel Industry," *Management Science*, 60 (9), 2114–2137.

Lim, Michael K, Ho-Yin Mak, and Ying Rong (2014), "Toward Mass Adoption of Electric Vehicles: Impact of the Range and Resale Anxieties," *Manufacturing & Service Operations Management*, 17 (1), 101–119.

Lin, Jenn-Rong and Ta-Hui Yang (2011), "Strategic Design of Public Bicycle Sharing Systems with Service Level Constraints," *Transportation Research Part E: Logistics and Transportation Review*, 47 (2), 284–294.

Low, David W (1974), "Optimal Dynamic Pricing Policies for an M/M/s Queue," *Operations Research*, 22 (3), 545–561.

Lu, Mengshi, Zhihao Chen, and Siqian Shen (2017), "Optimizing the Profitability and Quality of Service in Carshare Systems Under Demand Uncertainty," *Manufacturing & Service Operations Management*.

Mak, Ho-Yin, Ying Rong, and Zuo-Jun Max Shen (2013), "Infrastructure Planning for Electric Vehicles with Battery Swapping," *Management Science*, 59 (7), 1557–1575.

Manning, Christopher D, Hinrich Schütze, and Prabhakar Raghavan (2008), *Introduction to Information Retrieval*, Cambridge University Press.

Marecek, Jakub, Robert Shorten, and Jia Yuan Yu (2016), "Pricing Vehicle Sharing with Proximity Information," *arXiv preprint arXiv:1601.06672*.

McFadden, Daniel and Kenneth Train (2000), "Mixed MNL Models for Discrete Response," *Journal of Applied Econometrics*, 15 (5), 447–470.

Millard-Ball, Adam, Rachel Weinberger, and Robert Hampshire (2013), "Comment on Pierce and Shoup: Evaluating the Impacts of Performance-Based Parking," *Journal of the American Planning Association*, 79 (4), 330–336.

Millard-Ball, Adam, Rachel R Weinberger, and Robert C Hampshire (2014), "Is the Curb 80% Full or 20% Empty? Assessing the Impacts of San Francisco's Parking Pricing Experiment," *Transportation Research Part A: Policy and Practice*, 63, 76–92.

Nair, Rahul and Elise Miller-Hooks (2011), "Fleet Management for Vehicle Sharing Operations," *Transportation Science*, 45 (4), 524–540.

Navigant Research (2016a), "Carsharing Programs," URL http://bit.ly/2JKpHD5.

— (2016b), "Carsharing Services Will Surpass 12 Million Members Worldwide by 2020 Navigant Research," URL http://bit.ly/2w3uZS9.

Nyotta, Bobby, Fernanda Bravo, and Jacob Feldman (2019), "Free Rides in Dockless, Electric Vehicle Sharing Systems," *Working Paper*.

Observer (2020), "E-scooters Have Doubled in Popularity and Tripled in Injuries," URL https://observer.com/2020/01/e-scooters-popularity-injuries-us-millennials/.

O'Mahony, Eoin and David Shmoys (2015), "Data Analysis and Optimization for (Citi) Bike Sharing," in "Proceedings of the AAAI Conference on Artificial Intelligence," volume 29.

Ottosson, Dadi Baldur, Cynthia Chen, Tingting Wang, and Haiyun Lin (2013), "The Sensitivity of On-street Parking Demand in Response to Price Changes: A Case Study in Seattle, WA," *Transport Policy*, 25, 222–232.

Owen, Susan Hesse and Mark S Daskin (1998), "Strategic facility location: A Review," 111 (3), 423–447.

Owen, Zachary and David Simchi-Levi (2018), "Price and Assortment Optimization for Reusable Resources," *Available at SSRN 3070625*.

ParkMobile (2021), "ParkMobile," URL https://parkmobile.io/.

Pay By Phone (2021), "Pay By Phone," URL https://www.paybyphone.com/how-it-works.

Pfrommer, Julius, Joseph Warrington, Georg Schildbach, and Manfred Morari (2014), "Dynamic Vehicle Redistribution and Online Price Incentives in Shared Mobility Systems," *IEEE Transactions on Intelligent Transportation Systems*, 15 (4), 1567–1578.

Pierce, Gregory and Donald Shoup (2013), "Getting the Prices Right: An Evaluation of Pricing Parking by Demand in San Francisco," *Journal of the American Planning Association*, 79 (1), 67–81.

Quandt, Richard E (1960), "Tests of the Hypothesis that a Linear Regression System Obeys Two Separate Regimes," *Journal of the American Statistical Association*, 55 (290), 324–330.

Raviv, Tal, Michal Tzur, and Iris A Forma (2013), "Static Repositioning in a Bike-sharing System: Models and Solution Approaches," *EURO Journal on Transportation and Logistics*, 2 (3), 187–229.

Rennhoff, Adam D and Mark F Owens (2012), "Competition and the Strategic Choices of Churches," *American Economic Journal: Microeconomics*, 4 (3), 152–70.

Ross, Sheldon M (2014), *Introduction to Probability Models*, Academic Press.

Roth, Gabriel Joseph (1965), *Paying for Parking*, volume 33, Institute of Economic Affairs London.

Rusmevichientong, Paat, Mika Sumida, and Huseyin Topaloglu (2020), "Dynamic Assortment Optimization for Reusable Products with Random Usage Durations," *Management Science*.

Santa Monica Daily Press (2019), "Prices spike for e-scooter rentals," URL `https://www.smdp.com/prices-spike-for-e-scooter-rentals/177117`.

Sarrias, Mauricio and Ricardo Daziano (2017), "Multinomial Logit Models with Continuous and Discrete Individual Heterogeneity in R: The gmnl Package," *Journal of Statistical Software*, 79 (2), 1–46.

Sayarshad, Hamidreza, Sepideh Tavassoli, and Fang Zhao (2012), "A Multi-periodic Optimization Formulation for Bike Planning and Bike Utilization," *Applied Mathematical Modelling*, 36 (10), 4944–4951.

Schuijbroek, Jasper, Robert C Hampshire, and W-J Van Hoeve (2017), "Inventory Rebalancing and Vehicle Routing in Bike Sharing Systems," *European Journal of Operational Research*, 257 (3), 992–1004.

Shoup, Donald (2017), *The High Cost of Free Parking: Updated Edition*, Routledge.

Shoup, Donald C (2006), "Cruising for Parking," *Transport Policy*, 13 (6), 479–486.

Simeonova, Emilia, Janet Currie, Peter Nilsson, and Reed Walker (2019), "Congestion Pricing, Air Pollution, and Children's Health," *Journal of Human Resources*, 0218–9363R2.

Simon, Dan and Stephen A Spiller (2016), "The Elasticity of Preferences," *Psychological Science*, 27 (12), 1588–1599.

Singla, Adish, Marco Santoni, Gábor Bartók, Pratik Mukerji, Moritz Meenen, and Andreas Krause (2015), "Incentivizing Users for Balancing Bike Sharing Systems." in "AAAI," 723–729.

Sprent, P (1961), "Some Hypotheses Concerning Two Phase Regression Lines," *Biometrics*, 17 (4), 634–645.

Stamatopoulos, Ioannis, Achal Bassamboo, and Antonio Moreno (2020), "The Effects of Menu Costs on Retail Performance: Evidence from Adoption of the Electronic Shelf Label Technology," *Management Science*.

Takacs, Lajos (1969), "On Erlang's Formula," *The Annals of Mathematical Statistics*, 40 (1), 71–78.

The San Diego Union-Tribune (2016), "Car2go ceases San Diego operations," URL `http://bit.ly/2EditnB`.

Train, Kenneth E (2009), *Discrete Choice Methods with Simulation*, Cambridge university press.

U.S. Department of Energy (2016), "Alternative Fuels Data Center: Emissions from Hybrid and Plug-In Electric Vehicles," URL `http://bit.ly/2Edjs7h`.

Van Doorn, Erik A and GJK Regterschot (1988), "Conditional PASTA," *Operations Research Letters*, 7 (5), 229–232.

Vickrey, William (1955), "Some Implications of Marginal Cost Pricing for Public Utilities," *The American Economic Review*, 45 (2), 605–620.

Vieth, Elisabeth (1989), "Fitting Piecewise Linear Regression Functions to Biological Responses," *Journal of Applied Physiology*, 67 (1), 390–396.

Waserhole, Ariel and Vincent Jost (2016), "Pricing in Vehicle Sharing Systems: Optimization in Queuing Networks with Product Forms," *EURO Journal on Transportation and Logistics*, 5 (3), 293–320.

Weikl, Simone and Klaus Bogenberger (2013), "Relocation Strategies and Algorithms for Free-Floating Car Sharing Systems," *IEEE Intelligent Transportation Systems Magazine*, 5 (4), 100–111.

Witten, Ian H and Timothy C Bell (1991), "The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression," *IEEE Transactions on Information Theory*, 37 (4), 1085–1094.

Wolff, Ronald W (1982), "Poisson Arrivals See Time Averages," *Operations Research*, 30 (2), 223–231.

Wooldridge, Jeffrey M (2010), *Econometric Analysis of Cross Section and Panel Data*, MIT press.

Yan, Xiang, Jonathan Levine, and Robert Marans (2019), "The Effectiveness of Parking Policies to Reduce Parking Demand Pressure and Car Use," *Transport Policy*, 73, 41–50.

Yoon, Seunghwan and Mark E Lewis (2004), "Optimal pricing and admission control in a queueing system with periodically varying parameters," *Queueing Systems*, 47 (3), 177–199.

Zhang, Dennis J, Hengchen Dai, Lingxiu Dong, Qian Wu, Lifan Guo, and Xiaofei Liu (2019), "The Value of Pop-Up Stores on Retailing Platforms: Evidence from a Field Experiment with Alibaba," *Management Science*, 65 (11), 5142–5151.