Title
A Reinforcement Learning Approach for Crowdshipping in Food Delivery: Role of Pricing Decisions

Author
Pourrahmani, Elham

Publication Date
2022

A Reinforcement Learning Approach for Crowdshipping in Food Delivery: Role of Pricing Decisions

By

ELHAM POURRAHMANI

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Civil and Environmental Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Miguel Jaller, Chair

Susan Handy

Ilias Tagkopoulos

Committee in Charge

2022

ABSTRACT

The recent significant increase in the online food delivery market both in the US and many other countries and the emergence of transformative mobility services have brought both challenges and opportunities. While their evolution has provided consumers with a variety of options, existing services lack transparency in how they operate and their fee structure. Moreover, their long-term impact on human health, the environment, and social welfare is either unknown or unexplored. This study formulated and solved a simulation and optimization of a pricing model based on reinforcement learning techniques to establish dynamic, zone-based pricing for online food delivery services. A sample of restaurant or other food outlet dining experiences in San Francisco was used as a case study to test the model's performance. The designed pricing method outperformed the alternative static and myopic pricing strategies. The results also reveal the importance of platform providers' decisions regarding total profit and deliveries. Optimal surge pricing coupled with the efficient distribution of drivers in delivery regions is an important means of improving a service's impact on the environment and social welfare. Microanalysis has concluded that the change in general cost per customer relies heavily on the value they place on the time they save using online food delivery services as an alternative to dining out. However, the elimination of active transport time among those who used to bike or walk to access food outlets for their meals negatively impacts their health in the long term. This could potentially be mitigated by ordering healthy food or engaging in other forms of physical activity during the time saved ordering food online.

Contents

ABSTRACT ii

List of Figures v

List of Tables vi

1. Introduction 1

1.1 Context and Motivation 1

1.2 Focus and Scope 4

1.3 Relevance and Importance 7

1.4 Questions and Objectives 9

1.5 Overview of the Structure 11

2. Literature Review: Crowdshipping in Research 12

2.1 Survey and Data Analysis 12

2.2 Matching and Routing Optimization 19

2.3 Research Gaps and Limits 27

2.4 Summary 30

3. Background 32

3.1 Pricing in Online Food Delivery 32

3.2 Delivery Pricing in the Research 37

3.2.1 Empirical and Experimental 39

3.2.2 Analytical 41

3.3 Conjoint Analysis 44

3.4 Dynamic Programming 47

3.4.1 Description 47

3.4.2 Formulation 48

3.5 Reinforcement Learning 51

3.5.1 Elements 52

3.5.2 Algorithms 56

3.6 Summary 61

4. Data Collection and Analysis 63

4.1 Food Delivery Data Collection and Analysis 63

4.1.1 Menu Cost 65

4.1.2 Delivery fee 67

4.1.3 Other fees 69

4.2 Conjoint Analysis Process 71
4.2.1 Implementation 72
4.2.2 Results 76
4.3 Summary 80
5. Materials and Methods 81
5.1 Problem Definition 82
5.1.1 Platform 83
5.1.2 Requesters 84
5.1.3 Drivers 85
5.2 Dynamic Programming 87
5.3 Reinforcement Learning 89
5.3.1 Value Function Approximation and Algorithm 90
5.3.2 Computational complexity 97
5.3.3 Implementation 98
5.4 Summary 99
6. Results and Analysis 100
6.1 Case Study Description 100
6.2 Algorithm Parameter Settings 103
6.3 Food Delivery Scenarios 107
6.3.1 Platform 108
6.3.2 Drivers 113
6.3.3 Requesters 116
6.3.4 System and Environment 121
6.4 Summary 122
7. Conclusion 125
7.1 Summary 129
7.1.1 Platform provider 129
7.1.2 Drivers 130
7.1.3 Requesters 131
7.1.4 Environment and health 131

List of Figures

Figure 1 Share of each item in the pricing scheme for each platform (Kinetic 2019) ... 36
Figure 2 Fees as % of food menu price for each platform (Kinetic 2019) ... 36
Figure 3 Generalized policy iteration algorithm (Sutton and Barto 2018) ... 50
Figure 4 Sample RL episodes ... 53
Figure 5 RL diagram ... 53
Figure 6 SARSA (on-policy TD method) ... 59
Figure 7 Q-learning (off-policy TD method) ... 59
Figure 8 Average delivery fee variation per day for different food types and apps ... 68
Figure 9 Average share of fees for a typical order from (a) Subway (regular oven-roasted turkey sandwich, foot long) and (b) Chipotle (chicken burrito bowl) ... 71
Figure 10 Conceptual model ... 72
Figure 11 Conjoint rating-based table ... 76
Figure 12 summary of the final sample ... 77
Figure 13 relative importance of attributes ... 79
Figure 14 Pricing model implementation overview ... 82
Figure 15 Solution approach step 1 ... 89
Figure 16 Solution approach step 2 ... 90
Figure 17 MC-based SGD pseudocode (Sutton and Barto 2018) ... 94
Figure 18 MC-based SGD algorithm pseudocode ... 95
Figure 19 ES policy improvement algorithm pseudocode ... 97
Figure 20 RL implementation in Python ... 99
Figure 21 Eat out trip distribution description ... 101
Figure 22 Eat-out trip destination distribution ... 103
Figure 23 Algorithm convergence behavior ... 104
Figure 24 Policy value progress using the proposed policy improvement algorithm (VFA+PI), memory replay, concurrent environments, and random policy (VFA) ... 106
Figure 25 Total earning and deliveries for (a) Pr and b) SW scenarios ... 111
Figure 26 Average temporal price deviation per zone ... 112
Figure 27 Surge multiplier vs. delivery distance for matched requesters and drivers ... 113
Figure 28 Drivers' profit gain and loss in (a) Pr and (b) SW scenarios ... 114
Figure 29 Drivers relocation distances ... 116
Figure 30 Portions of requests and deliveries ... 117
Figure 31 Requesters' change in general cost after switching from eat-out trip to online food delivery. 119
Figure 32 Change in general cost versus value of time for requesters ... 120
Figure 33 Change in total driving distance, active travel time, and total saved time by switching to online food delivery in a) Profit and b) Social-Welfare ... 121
Figure 34 Total drivers' profit, platform's commission, and requesters' change in general cost across fleet sizes for a) Pr and b) SW ... 122

List of Tables

Table 1 Survey and data analysis review summary 17
Table 2 Matching and routing optimization review summary 24
Table 3 Restaurant information 64
Table 4 Food menu price fluctuation for various foods, apps, and cities 66
Table 5 Other fees information for all apps and cities 69
Table 6 Survey attributes and levels 73
Table 7 Part-worth utilities for all variables estimated, including all the respondents 78
Table 8 Binomial logit model summary 79
Table 9 Abbreviations and definitions 107
Table 10 Total deliveries for different fleet size, drivers' distribution, and pricing schemes 109
Table 11 Total profit for different fleet size, drivers' distribution, and pricing schemes 109
Table 12 Delivery requesters experience 120
Table 1 Increase in deliveries by dynamic pricing versus alternative pricing strategies in SW scenarios 129
Table 2 Increase in profit by dynamic pricing versus alternative pricing strategies in Pr scenarios 129
Table 15 T test results for number of deliveries resulted from different pricing strategies with random distribution 132
Table 16 T test results for number of deliveries resulted from different pricing strategies with weighted distribution 133
Table 17 T test results for number of deliveries resulted from different pricing strategies with weighted-No choice distribution 134
Table 18 T test results for earnings resulted from different pricing strategies with random distribution 135
Table 19 T test results for earnings resulted from different pricing strategies with weighted distribution 136
Table 20 T test results for earnings resulted from different pricing strategies with weighted-No choice distribution 137
Table 21 T test results for number of deliveries resulted from weighted and weighted-no choice distribution 138

1. Introduction

1.1 Context and Motivation

According to the U.S. Department of Commerce, total US ecommerce sales reached $759.47 billion in 2020, a 31.73 percent year-over-year increase from 2019's $576.53 billion. The growth in e-commerce has been accompanied by increased customer expectations, with demands for faster and cheaper deliveries. As a result, the logistics behind those transactions, and especially the Last-Mile Delivery (LMD) needs to be fast, inexpensive and reliable (Gdowska, Viana et al. 2018). LMD refers to the last stage of the supply chain, where items are delivered to their destination (e.g., businesses, stores, or residences). For carriers, this is the most inefficient portion of deliveries, and accounts for more than 40% of the total supply chain costs in the U.S. (Joerss, Schröder et al. 2016), excluding pickup, line-haul, and sorting costs. Moreover, within the e-retailer supply chain, the LMD transport component represents about 40% of the used energy and generated emissions (Thompson 2015, Generation IM, Preston et al. 2020). LMD is also responsible for traffic externalities in urban and suburban areas such as congestion, emissions, accidents, wear-and-tear on road infrastructure, and parking issues (Rodrigue and Dablanc 2011), as increased demand has pushed more trucks on to the roads (Ibbetson 2019, New York City Department of Transportation 2019).

To reduce the externalities of LMD, package delivery companies are investing in such solutions as optimizing vehicle tours (Ursani, Essam et al. 2011, Kritzinger, Doerner et al. 2012, Ruan, Lin et al. 2012, Xiao and Konak 2016), use of electric vehicles (Cavadas, de

Almeida Correia et al. 2015, Roberti and Wen 2016, Jaller, Pineda et al. 2019) or introducing policy restrictions on delivery time (Jaller and Holguín-Veras 2013, Holguín-Veras and Aros-Vera 2014, Jaller, Penagos et al. 2016, Holguín-Veras, Hodge et al. 2018), shipping routes (Holguín-Veras, Xu et al. 2015), the size and weight of packages (Holguín-Veras, Jaller et al. 2013), low emission zone definitions (Giuliano 2013) and consolidation strategies (Morana, Gonzalez-Feliu et al. 2014, Jaller and Pahwa 2020, Tiwari, Wee et al. 2021).

Additionally, transforming the LMD business model has captured increased attention from both researchers and stakeholders in recent years (Devari, Nikolaev et al. 2017). Among the newly proposed LMD models, there is one where professional freight (PF) delivery is supplemented or replaced by crowds of ordinary individuals willing to integrate shipment tasks with their own itinerary (Jaller, Otero-Palencia et al. 2020). These shippers are assigned delivery tasks and get a small compensation fee that reimburses them for additional costs. The widespread use of smartphones and the recent emergence of mobility apps increase the opportunity for this strategy. There are many names associated with this new shipping, all used interchangeably: crowdshipping, crowd logistics, crowd-sourced delivery, cargo hitching, and collaborative logistics (Rai, Verlinde et al. 2017). This study uses the term crowdshipping for most of the content.

Few definitions exist for crowdshipping in the literature (Mehmann, Frehe et al. 2015, Rai, Verlinde et al. 2017, Rześny-Cieplińska and Szmelter-Jarosz 2019). Based on the research background, this study defines crowdshipping as the outsourcing of delivery tasks to crowds of on-demand ordinary agents to finish the last mile delivery and receive a small compensation fee in return. The service is usually supported by an online platform with various features (e.g., matching, routing, pricing). Crowds' transport mode and time shift are determinant factors in size and type of delivery items. While the original objective is to decrease delivery costs for retailers, it might (but not necessarily) lead to environmental benefits if it happens by zero-emission means of transport or through already existing trips with a minimal detour. This is due to the fact that crowdshipping could support sustainability efforts by using excess transportation capacity for freight deliveries without adding a new trip to the network. In a successful implementation, crowdshipping benefits society by reducing the number of freight delivery trucks in urban/suburban areas. It also benefits companies by helping them reduce their delivery costs while maintaining the same level of service. This novel design also creates an opportunity for social collaboration, for non-professional individuals to be involved in LMD practices and play a role in a social activity that might ultimately lead to a more sustainable community. If these benefits materialize, crowdshipping could lead to a more environmentally friendly and economically efficient system that ultimately improves everyone's quality of life.

However, these positive outcomes are based on optimistic assumptions that may not necessarily happen in practice. For instance, some crowdshipping services operate in similar ways to other courier or transport services, where the carriers (crowds of individuals in this case) exclusively conduct transport delivery activities, and not as part of their regular travel needs. Nevertheless, during the COVID-19 pandemic, crowdshipping services offered an important social benefit by enabling some of the increased shipments and home deliveries that are necessitated by quarantined households, minimizing their risk.

Crowdshipping is challenging due to its novelty, as well as its lack of operational uniformity and real-world systems that disseminate data. Implementing an effective and efficient system requires developing a better understanding of the involved stakeholders, as well as the environmental, and economic impacts of the strategy. Pourrahmani and Jaller (2021) conducted a state-of-practice and research review on crowdshipping. The authors extracted critical characteristics of existing platforms in the industry, proposed a typology to categorize current crowdshipping practices based on platform type, delivery type, mode, pricing strategies; and identified potential unintended consequences, challenges, and opportunities associated with the service. According to the findings, insufficient details are available on delivery pricing criteria and algorithms in practice. In almost all the operational studies, pricing was simply treated as either being fixed or incremental in distance/time, although the concept received more thorough attention in the passenger ridesharing market.

1.2 Focus and Scope

Pricing affects both demand and supply in crowdshipping. It plays an important role in maintaining the supply in the system where couriers may participate in several other on-demand services, including passenger rideshare markets. The importance of pricing on demand is emphasized more in (Rougès and Montreuil 2014) that point to the higher cost of unfulfilled requests in delivery compared to the passenger market, where other modes of transportation are readily available. According to this, pricing for delivery is the focus of this study.

Overall, three types of pricing and payment schemes were identified in crowdshipping (Pourrahmani and Jaller 2021).

1. **Hourly or per time block.** Payment is made for an entire delivery time shift or per hour, and does not change, whether the delivery task takes longer or finishes earlier. The rate varies depending on the region, time of the day, or weather conditions. There are limits set on the maximum number of deliveries per shift. Amazon Flex, Deliv, and Shipt have implemented this pricing strategy, whereby couriers choose their time shift and are paid per shift/hour of delivery.

2. **Fixed per task with possible incremental charges.** This type of payment starts with a flat fee and can increase depending on the order or parcel's weight, size, delivery urgency, time, and distance (e.g., New Dada and Hummer crowdshipping). This pricing scheme works well in attracting new customers; however, it might not be profitable in later phases without incremental charges. Many crowdshipping platforms apply this strategy. Some charge a fixed rate per task (e.g., Doordash and Instacart), while others also include incremental charges (e.g., Postmates and Deliv). Uber implements both strategies alternatively in different regions.

3. **Negotiation and biding.** In this scheme, the sender/platform suggests an initial flexible price and potential couriers bid, then the best offer is accepted. Nimber and DHL MyWays are among the platforms that apply this strategy.

Customer tips, service fees, and surge (busy time) pricing might impact the initially determined price of each strategy. For more information about pricing and payment schemes in the selected crowdshipping platforms, it is referred to (Pourrahmani and Jaller 2021).

Online food delivery services which is considered in this study represents a major application for crowdshipping. Food delivery services are growing popular in the U.S. as two-thirds of Americans now use food delivery apps to order food online. According to Morgan Stanley Research in November 2019, 65% of a survey respondents have ordered food for delivery online in the last six months. Meanwhile, ordering food delivery by phone decreased by 3% per year through 2025. Annual online food delivery sales in the U.S. could grow by 18%, from 2018 to 2025 (Morgan Stanley 2020, Noah Lichtenstein 2020).

Food delivery has existed since 1889 when a pizza was delivered in Naples, Italy. It has evolved until 1995, when the first online restaurant delivery service, Waiter.com, was launched in the Bay Area, featuring more than 60 different restaurants. The current modern form of the food delivery services initiated in 2004 when GrubHub was founded in Chicago (Caitlyn Hitt 2020). Nowadays, the food delivery digital platforms are increasing in number as they become more popular. Some platforms deliver almost everything including, food and fresh items. For example, Amazon Flex drivers deliver various Amazon products, including Amazon Fresh and Amazon Restaurants (Carter 2019). Postmates delivers a wide range of products, with food deliveries being the most significant portion. Others might exclusively deliver food from restaurants (e.g., GrubHub) or grocery stores (e.g.,Instacart).

Recently, restaurants and food providers have responded to difficulties by increasing the menu prices to cover the extra costs associated with the third-party delivery platforms or services. These additional fees have become a burden to customers paying more than the original price for the food (João Pedreda 2020). On the other hand, drivers complain about long wait times and low compensation fees. Moreover, there is a lack of clarity about factors affecting delivery fees. For instance, the delivery fee might be more expensive depending on other factors rather than only the distance from the customer. Hence, platforms have recently focused on new ways to charge customers to reduce their burden and maintain profitability for themselves and for drivers.

Setting delivery fees proportional to delivery distance is psychologically more appealing to users and mitigates pricing confusion. Introducing variation in delivery fees due to change in demand across spatial zones and time of the day potentially enhances the service performance in terms of deliveries and profitability. Coupling pricing with effective demand prediction and signaling the level of pricing to couriers in real-time can increase their profit. This happens by efficiently repositioning the couriers to zones where demand exists to satisfy their expected earnings and reducing their relocation operation cost. Eventually, the traffic and environmental food prints are mitigated, particularly for crowdshipping systems operating by motor vehicles.

1.3 Relevance and Importance

Given the importance of pricing for delivery services and the need for innovation and research on food delivery pricing, this study addresses the gap in the empirical analysis for food delivery pricing. Next, it conducts a dynamic zone-based pricing simulation-optimization model for online food delivery with crowds of drivers.

First, the study collects empirical data from four food giant delivery apps in three Californian cities. The data includes details on delivery pricing for various times of the day, food types, and delivery distances. The analysis of the results provides insightful information for hidden costs of delivery in different apps and its variations across region and time. Although the empirical data analysis by (Kinetic 2019) and (Noah Lichtenstein 2020) was eye-opening, the authors were limited to comparing total pricing for various apps. More insight is needed into the temporal change in delivery fees and the impact of delivery region on price in various food apps.

Second, the study formulates and solves dynamic zone-based pricing for an online food delivery platform using reinforcement learning simulation and optimization techniques. This platform is assumed to be of type b (Pourrahmani and Jaller 2021), the most common delivery platform featuring tasks such as matching drivers, requesters, and senders, facilitating navigation, tracking deliveries, and setting prices. Drivers are on-demand crowds willing to fulfill delivery tasks and receive a small compensation in return. Requesters place a food delivery order given the offered price and wait time. The dynamic pricing model finds the optimal delivery fees with respect to requesters' elasticities and drivers' preferences by maximizing profits and social welfare. Delivery fees represent a key item in food delivery total cost and affect requesters' payment and the revenue of drivers and platform providers.

Dynamic delivery pricing is rarely studied in the literature, although it has captured researchers' attention in the passenger rideshare market (Banerjee, Riquelme et al. 2015, Cachon, Daniels et al. 2017, Guda and Subramanian 2017, Guda and Subramanian 2019, Garg and Nazerzadeh 2021). Most of these studies are highly analytical and are not applied. They rarely consider supply and demand elasticities with pricing and matching decisions. Furthermore, they lack the flexibility of simulation models that enable the study of different actors' behavior in the system and environmental impacts under different settings.

This study contributes to the growing literature in several ways: 1) empirical data collection and analysis for food delivery pricing in California; 2) formulating a dynamic zone-based pricing model for a many-to-many food delivery problem considering variations in drivers' preferences and requesters' elasticities; 3) estimating an elasticity function for food delivery requesters given the time of day, delivery price, and wait time; 4) building an agent-based macrosimulation model based on reinforcement learning to train and simulate the pricing model; 5) designing multiple scenarios considering various drivers distribution strategies as well as model objectives; 6) evaluating the service operation and its impact on platforms, drivers, requesters, and environment.

1.4 Questions and Objectives

The main objective of this study is to explore crowdshipping operation in food delivery and evaluate its impacts on the system in which pricing decisions play the key role. To do so, empirical data is collected from a sample of food delivery requests and a dynamic zone-based pricing simulation model is built. This main objective can further be divided into secondary objectives as follows:

a) Collecting food delivery pricing data from different online food delivery apps and studying the factors affecting food delivery pricing

b) Conducting conjoint analysis survey to build statistical utility functions and estimate individuals' elasticity functions for food delivery

c) Formulating dynamic zone-based pricing and presenting a solution framework based on simulation-optimization techniques

d) Evaluating the performance of dynamic zone-based pricing against static and myopic pricing

e) Designing various scenarios in terms of pricing objectives as well as drivers' fleet size, distribution, and preferences; and determining performance indicators to evaluate the pricing strategy

f) Conducting a series of simulation scenarios to measure the impacts of pricing decisions on platform profitability and social welfare, drivers' revenue, requesters' experience, and environment in various contexts

In particular, the study attempts to answer the following questions:

1) Where are the influential factors in food delivery pricing? Does the pricing scheme vary by the platform provider, city, or region? Is there substantial evidence of temporal surge pricing in food delivery pricing?

2) How do delivery fees, wait times, and time of the day affect requesters' delivery decisions? Are these significant in their elasticity value?

3) Is dynamic zone-based pricing better than static and myopic pricing strategies? Does it increase platform profitability and deliveries?

4) How different are the social welfare and profit maximization scenarios regarding the number of deliveries, total profit, and price reliability for the platform? Are various settings for crowds of drivers affecting the system?

5) How much is the change in general cost and time saved by switching from eat-out trips to online food delivery for different mode users (walkers, bikers, and drivers)?

1.5 Overview of the Structure

This dissertation comprises seven chapters, each of which includes several sections and ends with a summary wherever appropriate. A broad literature review on crowdshipping is organized and provided in the next chapter, followed by an outline of research gaps and future directions. Chapter 3 focuses mainly on providing enough background information to support the methodology and materials used in the following chapters. It starts with a detailed description of current food delivery pricing and reviews the empirical data analysis and practices of other researchers in this field. Then, the dynamic pricing concept and foundations are clarified using examples in the transportation shared economy, followed by a paragraph about existing limitations and how this study intends to improve them. Chapter 3 further presents fundamental theory and formulations for conjoint analysis, dynamic programming, and reinforcement learning, which are principles for the pricing model formulation and solution framework in this study. Chapter 4 exclusively describes data collection and analysis. It starts with a detailed explanation of the food delivery data collection case study and relevant preliminary data analysis. Similarly, the conjoint analysis survey and sample details are presented afterward and end with elasticity function estimation results. The pricing mathematical formulation model, the dynamic programming equivalent problem and simulation-optimization solution framework are presented in Chapter 5. Chapter 6 includes all results about pricing algorithm performance and relevant discussions to respond to study objectives and questions. Finally, this study ends in Chapter 7 with an overall conclusion. References are attached.

2. Literature Review: Crowdshipping in Research

Although crowdshipping is an emerging research area, several published studies are contributing well to the domain. The author has identified two major categories of studies, according to the employed approach: 1) Survey and data analysis, and 2) Matching and routing optimization. This section reviews the critical studies conducted within each category and presents the underlying methodology, assumption, and findings. Summary of reviews for each section are illustrated in separate Tables (Table 1 and Table 2).

2.1 Survey and Data Analysis

In this category, authors offered an exploration of a wide range of factors that affect the probability of an individual participating in crowdshipping, and measured service success by conducting Stated Preference (SP) or Revealed Preference (RP) surveys and pilot tests. In general, the behavioral studies can be further split depending on whether the work is concentrated on which actor: senders, crowdshipper, or receiver.

Majority of papers studied the crowdshippers' behavior and preferences. Marcucci, Le Pira et al. (2017) conducted an SP survey from college students in Rome and reported that 87% of respondents agree to work as a crowdshipper, while this rate decreases as the package size increases or the compensation fee decreases. Over 90% were willing to accept packages from a crowdshipper, but the rate decreases if there is no way to contact the shipper or to track the delivery task. Serafini, Nigro et al. (2018) used an SP survey for the city of Rome and discrete choice models to identify important factors impacting metro passenger's willingness to be a crowdshipper by picking up parcels from lockers installed around the metro stations and deliver them along their journey. 43.1% of the sample was not willing to modify their original path, while 39% and 15% accepted deviation up to 300 and 600 meters, respectively. Important factors impacting an individual's willingness to participate were the location of locker to pickup parcel, compensation fee, income deposit frequency, and delivery booking type, in the order of mention. Ermagun and Stathopoulos (2018) explored a national dataset obtained from a leading crowdshipping company in the U.S. Preliminary analysis suggested that in 20% of cases, supply is less than the demand, and in 36.4% cases, they are equal. The authors developed a two-part supply model in which requesters are matched with drivers at an auction. authors found that drivers are less interested in deliveries with a restricted schedule. Built environment characteristics were more significant for origin than destination. Population and employment density were negatively correlated with receiving a request. On the other hand, job accessibility and diversity had a positive relationship with the available supply. The probability of receiving a bid was higher in trip origin with a higher percentage of families and auto ownership, while this was lower in origin and destination with low-income residents.

Punel and Stathopoulos (2017) conducted an online SP survey in the U.S. to gain insights on potential crowdshipping senders. The findings indicated that 7% of the sample already used the service, expressing the service is inefficient and complicated to use. Millennials were more eager to use the service as opposed to the older age population who are unfamiliar with the overall concept. Higher-income and employment, as well as lower education also contributed to more crowdshipping usage. Multinomial Logit (MNL) models were estimated and revealed distinct trends for different shipping distances. Local shipments were time-sensitive, while longer shipments focused on the management and control. Longer distance delivery emphasized the driver training and experience in the platform, while the driver rating score more influenced shorter distance delivery decisions. Gatta, Marcucci et al. (2019) conducted a Stated Preference (SP) survey in Rome, Italy, to estimate the potential crowdshipping senders. Shippers were metro passengers who pick up the parcels from lockers located at the stations. Around 50% and 25% of the sample foresaw a successful implementation in urban and suburban areas, respectively. The majority of potential senders were metro passengers or those residing within surrounding areas. Among the included variables, a flexible delivery schedule was the most significant one. The simulated scenario showed a total annual savings of 239 kg particulates and 1,098 tons of carbon dioxide emission. However, economic costs always exceeded revenue for the platform.

Meanwhile, there are a few studies which include more than one entity in their exploration. Le and Ukkusuri (2018) conducted SP and Revealed Preference (RP) surveys in the U.S. and Vietnam to understand the behavior of requesters and potential crowdshippers. The requesters were more likely to choose the service for items with shorter delivery times. 80% of individuals in the dataset were interested in being a crowdshipper, and having additional sources of income was the main motivation. A car represented the main mode of transportation (approximately 70%). The average accepted time deviation and average expected compensation fee for delivery tasks were around 12 minutes and $12, respectively. Authors developed a discrete-continuous model, including selectivity-bias terms on the U.S. dataset (Le and Ukkusuri 2018). They found that individuals who are more social media-driven are more interested in the service. Experience of transporting freight in the past influences the willingness to be a crowshipper. Individuals transporting freight while commuting are less likely to make a long detour, while other trip purposes have flexibility.

The higher the payment, the longer the distance they are willing to travel. Respondents are more likely to travel longer once they transport items for friends, colleagues, relatives, or neighbors. Punel, Ermagun et al. (2018) explored the differences between crowdshipping users and non-users based on an online survey across four U.S. states, and estimated logit models. 7.88% of the sample already used the service. Non-users had stronger beliefs in cost-saving and building a community, while users put more emphasis on the potential of being eco-friendly and efficient. Non-users were more conservative toward sharing information with drivers, which highlights the critical role of safety, privacy, and trust in attracting new users. Users were more among men, lower-income, and full-time employees. Crowdshippers were more interested in medium distance deliveries, while the service is prevalent for LMD. Dense residential and high job accessibility areas were identified as the best and worst areas for the service operation, respectively. Paloheimo, Lettenmeier et al. (2016) conducted a trial crowdshipping for library deliveries in Finland via the PiggyBaggy platform. Crowds were among library customers and locals; the compensation fee per delivery was 2-5 euros. Deliveries under 5 km were on bike, and the total estimated car distance traveled was 19 km (1.6 km per delivery). Scheduling the deliveries was a significant concern among participants. The main motivations for customers and couriers were "making life and transport routines easier" and "trying something new," respectively. Health benefits and exercise were also among the encouragement factors for cyclists. The library played a vital role as a guarantor of the service credibility.

In addition to above discussed methods, Rai, Verlinde et al. (2018) analyzed the externalities of a crowdshipping service based on data available from a platform and survey results of its users. Results from several scenarios showed that the crowdshipping is not sustainable compared to traditional logistic shipments. Decreasing total delivery distance as well as detour distance and increasing number of deliveries per trip significantly decreased externality costs, among which reduced detour distance by 44% brought the most significant benefits.

Table 1 Survey and data analysis review summary

Reference	Case study	Variables		
		Sender	Courier	Receiver
(Gatta, Marcucci et al. 2019)	SP survey in Rome, Italy Public transit-based with automated lockers, e-commerce parcels	-Sociodemographic information -Attitude and motivation -Residency location -Delivery schedule -Delivery time -Delivery fee -Delivery tracking option -Cost/benefit analysis -Emission analysis		
(Marcucci, Le Pira et al. 2017)	SP survey-Rome, Italy College student		-Willingness to work -Package size -Compensation fee	-Delivery tracking option
(Serafini, Nigro et al. 2018)	SP survey-Rome, Italy Public transit-based with automated lockers, e-commerce parcels		-Sociodemographic information -Attitude and motivation -Willingness to work -Willingness to detour -Locker location -Compensation fee -Bank credit frequency -Delivery booking type	
(Punel and Stathopoulos 2017)	SP survey in U.S.	-Sociodemographic information -Attitude and motivation -Delivery distance -Delivery cost -Package size -Experience -Driver expertise -Driver rating -Delivery management -Delivery time		

Reference	Case study	Variables		
		Sender	Courier	Receiver
		-Delivery schedule		
(Paloheimo, Lettenmeier et al. 2016)	Trial operation for library, in Finland via PiggyBaggy platform, Library customers and locals to deliver books and media to customers'		-Attitude and motivation -Compensation fee -Delivery distance -Delivery schedule	-Attitude and motivation -Delivery schedule
(Le and Ukkusuri 2018) (Le and Ukkusuri 2018)	SP and RP surveys in U.S. and Vietnam, Flyer distribution at TRB, Email to students at various colleges, schools, universities, and organizations, Social media advertisement, and Amazon Mechanical's Tur		-Sociodemographic information -Trip time (including detour time) -Profit -Package weight -Number of packages to deliver	-Sociodemographic information -Shipping cost -Delivery time -Delivery schedule -Delivery location -Platform ranking -Platform options -Payment method -Willingness to tip
(Punel, Ermagun et al. 2018)	Online survey for four U.S. states, namely: California, Florida, Georgia, and Illinois	-Sociodemographic information -Attitude and motivation -Delivery quality -Delivery efficiency -Delivery distance -Delivery privacy -Spatial density, diversity, design, and accessibility at ODs		
(Ermagun and Stathopoulos 2018)	National dataset from one of the leading platforms in the U.S. including 16,850 requests from 2070 cities in two years	-Sociodemographic -Package size -Request time of day, day of week -Delivery distance -Delivery cost -Delivery schedule -Spatial density, accessibility, and employment at ODs		

2.2 Matching and Routing Optimization

Matching and routing studies concentrated on the development of algorithms to match the unused transportation capacity optimally with the potential requests and choose efficient routes for transportation, considering constraints in terms of capacity, location, and time.

A number of studies considered public transit mainly or partially as the crowdshipping mode. Ghilas, Demir et al. (2013) introduced a static Pickup and Delivery Problem with Fixed transit Lines (PDP-FL) model for integration of passenger and freight and proposed a mixed-integer program solved by CPLEX. Compared to traditional delivery, reductions up to 27% and 70% in operation cost and CO2 were recorded, respectively. However, the proposed system increased the total trip time for customers. In another study, Ghilas, Demir et al. (2016) designed an adaptive neighborhood metaheuristic algorithm to solve a Pickup and Delivery Problem with Time Windows and Scheduled Lines (PDPTW-SL) to deliver packages using fixed transit lines and pickup and delivery vans. Results proved the performance efficiency of the algorithm considering the spatial pattern of the requests and the configuration of the scheduled lines. The more clustered the demand points, the better the performance of the system. In particular, the integrated transportation system led to 5% and 9% savings in operating costs and driving time, respectively.

Second stream of studies considered bikers and pedestrians as crowdshippers. Kafle, Zou et al. (2017) designed a crowdshipping system where pedestrians and bikers bid over the delivery tasks and proposed a mixed integer non-linear program (MINLP) to formulate the bid selection, pick up point location with truck routes design. The model was applied to random instances. Crowdshipping total saved cost and truck Vehicle-Mile Traveled (VMT) by 9% and 24%, respectively; however, total service time violated by 3%. Crowdshippers delivered 50% of tasks with total earnings of $184. Replacing cyclists by pedestrians reduced truck VMT by only 7%, while the total payment dropped to $29. This is because pedestrians have lower operating costs, speed, and capacity compared to bikers. Sensitivity analysis showed that truck operating costs and crowdshipper's Value of Time (VOT) impact the service efficiency significantly. Akeb, Moncef et al. (2018) proposed a model where neighbors delivered parcels on foot to the final customer. They implemented a mathematical circle packing model to determine the required number of neighbors and their locations, balancing neighbors' monthly gain. The proposed method is flexible enough to consider a wide range of delivery distances and monthly wages.

In this last stream of works which includes most of the publications, motor vehicles are crowdshippers. Pan, Chen et al. (2015) solved a crowdsourced delivery system for returned items using taxis where priority was given to passengers. They tested two different matching strategies: 1) goods can only be transported by one taxi, and 2) goods are allowed to be transferred between taxis. They defined a simple network-based heuristic and applied to the taxi's GPS trace database in China. Transshipment strategy generated 10 km more traveled distance per package compared to the other one. 90% of parcels were delivered within 24 hours using the second strategy, while this was roughly 55% in the first strategy, where 5% of packages had delivery time longer than four days. Archetti, Savelsbergh et al. (2016) formulated and solved a VRP with Occasional Drivers (VRPOD) using a multi-start heuristic. Two compensation schemes were assumed: fixed per task and proportional to detour. The heuristic solved several Solomon instances within seconds. The cost reduction was 25% with low detour flexibility and a high compensation rate, while this equaled 29% in the opposite case. Gdowska, Viana et al. (2018) solved a bi-level matching and routing LMD problem considering both crowdshippers and PF for delivery tasks. The probability of acceptance (rejection) of a delivery task by a crowdshipper followed a uniform random distribution. They modified the mathematical model proposed by Archetti, Savelsbergh et al. (2016) and solved it using a bottom-up heuristic. They examined the methodology on a set of random instances. In 24 of the 25 instances, total delivery cost was reduced by 9%. Arslan, Agatz et al. (2018) solved a dynamic capacitated pick-up and delivery problem with time windows considering both crowdshipping and dedicated delivery drivers. They developed an exact algorithm and a speed up heuristic within the context of an event-based rolling time horizon. Crowdshipping saved costs by 19- 37%. Making more stops per tour by crowdshippers reduced the number of dedicated drivers in the system who lost efficiency in terms of deliveries per mile. Sampaio, Savelsbergh et al. (2018) investigated the benefits of transshipments in crowdshipping using passenger cars. They designed a multi-depot pickup and delivery problem with time windows and proposed an adaptive large neighborhood search algorithm. Results indicated significant benefits where pick-up and delivery locations are far apart, and driver delivery shifts are short. They showed that the transshipment reduced the number of required drivers, while the number of requests per driver increased. Qi, Li et al. (2018) modeled a crowdshipping scenario where packages are transported from a central depot to a set of terminal points by truckers and then distributed among passenger car drivers who deliver them through open-loop routes. They developed a continuous approximation model to find the optimal size of the terminal service area minimizing the total wages paid to crowdshippers and truck drivers. Car drivers' wage response behavior and interplay with passenger rideshare market (synergy and competition) were considered in the model. They applied the model on 15 zip code area located at East Bay San Francisco, Bay Area. The findings indicated that shared mobility is not economically scalable compared to truck-only service except for short and low-density delivery areas. However, it saves asset costs related to fleet size and depot management. Yildiz and Savelsbergh (2019) studied a meal delivery problem using crowdsourced and company-provided drivers, to find optimal service areas for restaurants maximizing the profit. The findings indicated that the system's profitability highly depends on the location of the restaurant relative to its customers. Dedicated drivers could increase profit for rejected requests located closer to the restaurant, but not necessarily for furtherly located orders. Larger service areas increased dedicated drivers and dropped the crowds in the system. Allahviranloo and Baghestani (2019) explored the impact of crowdshipping on individuals' activity patterns and travel behavior by developing a mathematical model for dynamic scheduling of activities minimizing total travel time and deviation from regular activity patterns. The model was examined using the 2001 California household travel survey. Potential senders and couriers were identified and simulated based on their activity patterns. Results showed that a total of 167 (27%) tasks were successfully transferred to the carriers, the majority of which (46%) allocated to the evening period. Requesters saved a total of 5,130 minutes, while the carriers spent 2,316 minutes to deliver the tasks.

In addition of what was already discussed, there were a few papers which considered more than one mode for crowdshipping. Wang, Zhang et al. (2016) formulated a network min-cost flow model for crowdshipping service from pick-up parcel stations (pop-stations) to final customers. The compensation scheme was proportional to the detour distance. Several pruning strategies were evaluated to reduce network size for large scale experiments. The proposed methodology was applied to travelers' trajectories extracted from empirical data and a set of distributed pop-stations in Singapore and Beijing. The results confirmed the efficiency of the algorithm for large scale problems. Simoni, Marcucci et al. (2019) assessed crowdshipping impacts on congestion and emission in Rome, Italy using a dynamic traffic assignment model. They simulated three alternatives: 1) delivery truck, 2) car-based crowdshipping, and 3) public transit-based crowdshipping (Serafini, Nigro et al. 2018). Crowdshipping replaced truck deliveries where detour distance did not exceed a maximum threshold. Results showed that public-transit based crowdshipping benefits emission and congestion, while car-based alternative worsens them by 3-5% and 6-11%, respectively.

Table 2 Matching and routing optimization review summary

Reference	Crowds	Model and Method	Case study	Objective function	Decision variables	Constraints	Pricing	Dynamic	Evaluation measures
(Ghilas, Demir et al. 2013)	Integrated passenger and freight on public transit and taxis	MIP solved by CPLEX w/o user cuts	Hypothetical	Minimizing operation cost (transportation, transshipment and storage costs)	Routing, matching, scheduling	Arrival and departure time windows, vehicle capacity, service times	Fixed per delivery distance unit	x[1]	Driving time and CO2 emission
(Pan, Chen et al. 2015)	Integrated passenger and freight on a fleet of passenger taxis	Simulation-based matching heuristic	852 shops and 2,000 randomly generated packages with more than 7,000 taxi's GPS trace in China.	x	Matching	Passenger priority in routing and matching	x	x	Delivery time and the total transportation distance
(Archetti, Savelsbergh et al. 2016)	Occasional and professional drivers	IP, CPLEX and multi-start heuristic	Solomon instances	Minimizing total operation cost	Routing and matching	Vehicle capacity	Fixed per task and proportional to detour	x	Computation performance, cost, number of occasional drivers and deliveries
(Wang, Zhang et al. 2016)	Occasional crowdshippers to deliver from	Network min-cost flow model with several	Simulated based on trajectories	Minimizing total delivery cost	Matching	Delivery capacity	Fixed-rate proportional to detour	x	Computation performance

[1] Unapplicable

Reference	Crowds	Model and Method	Case study	Objective function	Decision variables	Constraints	Pricing	Dynamic	Evaluation measures
	pop-stations to final customers	pruning strategies to reduce problem size	extracted from empirical data						
(Ghilas, Demir et al. 2016)	Integrated passenger and freight on public transit and taxis	Adaptive neighborhood metaheuristic algorithm	Hypothetical	Minimizing total operation cost	Routing, matching, scheduling	Time windows, vehicle capacity, service times, transit departure time	Fixed per delivery distance unit	x	Computation performance, cost, driving time, number of vehicles and deliveries
(Kafle, Zou et al. 2017)	Biker/peds to receive packages from truckers at relay points and deliver to final customers	MINLP, and tabu search	Random hypothetical	Minimizing operating cost, payment to crowds, and delivery time window violation cost	Bide selecting, matching, routing, and scheduling	Available trucks, time window, capacity, delivery distance, buildings, scheduling	Biding	x	Computation performance, total cost, VMT and service time violation
(Gdowska, Viana et al. 2018)	Crowdshippers (in-store shoppers) and professional fleet for delivery	IP, Heuristic	Random instances	Minimizing the total delivery cost	Matching and routing	Capacity, Crowdshipper acceptance rate	Fixed per task	x	Cost
(Akeb, Moncef et al. 2018)	Neighbors collect and deliver parcels to the consumer in the	Surface packing model	Real data on parcels requiring a second delivery in Paris	Balancing neighbors' monthly gain	Available neighbors, and compensation fee	Population density and delivery distance	Fixed per task,	x	Interaction between monthly gain and service area radius

Reference	Crowds	Model and Method	Case study	Objective function	Decision variables	Constraints	Pricing	Dynamic	Evaluation measures
	neighborhood								
(Arslan, Agatz et al. 2018)	Ad hoc crowdshipper drivers and dedicated delivery vehicles	Exact solution algorithm and a speed up heuristic	Random instances	Minimizing total delivery cost	Matching	Capacity, time window, number of stops per tour, driving time, scheduling	Fixed per delivery distance unit	Event-based rolling time horizon	Cost, matching rate, available drivers, delivery distance
(Sampaio, Savelsbergh et al. 2018)	Crowdshipping with transshipment between drivers	Adaptive large neighborhood search	Random instances	Minimizing cost	Matching and routing	Time windows and synchronization	x	x	Driving distance, available vehicles, computation performance, number of transfers
(Qi, Li et al. 2018)	Passenger ridesharing drivers from a terminal point (freight only)	Continuous approximation optimization	15 zip code areas located in SF	Minimizing the total wages	Size of the terminal service area	Car drivers' wage response behavior and interplay with passenger rideshare market	Base + per mile (Fixed)	x	Delivery cost and distance
(Yildiz and Savelsbergh 2019)	Meal delivery with delivery drivers	IP service area and coverage planning model	Random instances	Maximizing the profit	Service area radius	System equilibrium, service quality, available couriers,	Fixed per delivery and per distance rate	x	Profit, mean delivery time

2.3 Research Gaps and Limits

According to the reviewed crowdshipping practices and research articles, limitations still exist in the field that must be addressed. A concise description of the gaps is presented below:

Crowdshipper employment status: in response to the raising concerns over unfavorable work condition within the gig economy, it is necessary to understand the workers' issues, determine appropriate regulation and identify the right institution to enforce it to reach to a balance between employees' basic labor protection and, feasibility of platforms operation.

Crowdshipper wages: considering the unique features of the crowdshipping platform, there is a need for further work on formulating crowdshippers' wages to include the costs associated with the required physical activity and wear-and-tear on the vehicle in addition to the standard distance-based fees present in ridesharing market. Moreover, exploring the possibility of sharing the unpredicted costs due to parking tickets, extra time delivery, or tolls between courier and firm is another promising area for future research.

System reliability and security: many reviewed studies demonstrated the importance of reliability and credibility to attract enough requesters and shippers to the system. Thus, providing enough information about registered couriers, the option to track the delivery process as well as insurance in the case of damaged or stolen packages, fraud, or delayed delivery are necessary. Referring to users' concerns over revealing their personal information such as home address to couriers, it is necessary to secure the system by presenting crowdshipper rating scores to users and enforcing background checks. Proposing alternate solutions for delivery drop offs such as dropping off at lockers within the vicinity of customers worth to study.

Service operation: crowdshipping supports sustainability goals assuming it operates through already existing trips. However, it is not clear if the current practices satisfy this assumption. Therefore, future studies must explore the strategies to ensure crowdshipping is happening through transportation excess capacity, and appropriate matching and pricing techniques are in place to avoid generating additional VMTs. Implementing the service along with passenger transportation and consolidating more deliveries per route are promising areas to support service efficiency and sustainability. Allowing transshipment between crowdshippers or multimodal crowdshipping is also among potential solutions to efficiently increase delivery distance and extend the service beyond short delivery within congested urban areas.

Market competition: as pointed out before, many crowdshippers participate in several on-demand services, including passenger rideshare market, which is more promising. Thus, it is necessary to study the factors impacting crowdshippers' willingness to work and understand the differences that exist between passenger and parcel markets. Then, incorporate the learned lessons into the platform's matching, routing, and pricing decisions to extract the most favorable condition for the platform's operation.

Infrastructure development: current croswdshippers have repeatedly raised concerns over difficulties finding parking space and long reception waiting time for delivery. This indicates the need to do more research on alternative infrastructural modernization to facilitate last-mile delivery. Applications of lockers or dedicated boxes at the household's place are among possible ideas.

Pricing: the compensation fee dramatically influences the availability of crowdshippers and their flexibility to detour. However, there are restrictions over the payments to maintain a satisfactory profitable service and avoid additional VMT due to newly generated trips. An appropriate compensation scheme leads to high capital efficiency, sustainable operation, a reasonable number of agents willing to deliver, and prevents the system from turning into an on-demand delivery business. Reviewed studies employed fixed payment rates per task or proportional to delivery time and distance. Fixed pricing is simple to implement, but not necessarily fair or tempting enough to satisfy or attract existing and new shippers. Pricing based on detour distance seems promising. However, it needs the crowdshipper's final destination to be known, which raises privacy concerns. Dynamic pricing has not been included in the reviewed crowdshipping papers, although some platforms (e.g., Instacart) implement surge pricing algorithm to match driver supply with demand during busy periods. Dynamic pricing practices in the passenger market showed to be highly effective in alleviating short-term demand fluctuations in a location. Additionally, spatial pricing which sets prices based on trip origin or both origin and destination has been studied in the passenger ride-sharing market, indicating significant profit gains for the platform from using spatial rate for the demand pattern to get more balanced across a network.

Assignment models and algorithms: crowdshipping requires fast decision making as couriers are available for a short time to be asked to fulfill a delivery task. Thus, efficient optimization techniques are needed to give the optimal solution quickly, minimizing total delivery cost. Additionally, the stochasticity and dynamics in demand and supply were not modeled in most of the implemented algorithms, while careful coordination of supply and demand is the key to maximize profit and ensure a target quality of service. Finally, each shipment is associated with a single parcel delivery in most models, although making multiple deliveries per trip increases its efficiency. Major retailers such as Amazon can consolidate three to five deliveries per trip due to their large quantity of orders.

Behavioral analysis: there are five stakeholders involved in the crowdshipping; platform, sender, receiver, shipper, and trucker, among which platform and truckers were rarely studied. While many variables were already considered, the other factors such as shipper's time availability, delivery urgency, package type, and value are not well understood. Further analysis is needed to capture the interaction among pairs of variables such as compensation fee, package type, crowdshipper's willingness to detour, or trip purposes.

Impact assessment: Although several reviewed studies analyzed the change in VMT, environmental footprints, and revenue resulted from crowdshipping, additional performance measures related to noise, accessibility, and public health, are still needed. Moreover, the distribution of impacts in society across different land-use types, as well as the population's level of income, is not well understood yet.

2.4 Summary

This section reviewed several academic research articles on crowdshipping, categorized them in two groups and summarized their main findings. It is found that the success of crowdshipping depends highly on the number of participants in the system where trust plays a critical role in attracting both couriers and senders/receivers into the service. Public responses to SP surveys indicate the appeal of monetary income as the primary motivation for crowdshippers to participate. Gaining health benefits and contributing to the social good were other possible motivations. Conversely, retailers implement crowdshipping as a part of their supply chain if it helps to reduce their LMD costs. Simulation and evaluation of hypothetical experiments and case studies revealed that couriers' income and operation costs, the flexibility to detour, available capacity, and speed all affect service cost efficiency compared to traditional freight delivery. Besides, the delivery characteristics (such as distance and density) also matter. This study also identified contradictory findings as few studies believe in increased profit with crowdshipping over long distance, and others found short distance delivery in dense areas more fruitful. About environmental impact, simulation results based on empirical data verified the fundamentals behind crowdshipping: it mitigates externalities if it happens through already existing trips and/or with cleaner modes; Otherwise, it might increase VMT and emissions through induced trips and longer detours motivated by compensation fees.

Most of the research articles published on crowdshipping either focus on behavioral analysis or matching and routing optimization. Except for a few studies, service externality evaluation (e.g., emissions and congestion) based on empirical data was not the focus of study. This indicates that many identified service challenges in practice, are still unresolved and represent potential research areas future.

3. Background

This section presents the essential background about pricing strategies currently in practice in food delivery market as well as those reviewed in the literature. Principal theories that are fundamental to the subjects and methods used later in this study (conjoint analysis, dynamic pricing, and reinforcement learning) are also described.

3.1 Pricing in Online Food Delivery

Online food delivery users are growing rapidly in many countries including the US. COVID-19 pandemic increased this rate of growth more than before. Since then, the platforms started charging higher prices and fees for delivery which have made them more expensive than before (Mike Pomranz 2021). While food delivery consumers agree to pay additional fees for convenience and time saving, the food delivery pricing system is confusing and unpredictable. At the same time, delivery companies compete to win the profitability and leadership in the market, thus looking for innovative ways to design their pricing scheme and charge consumers.

Although there are many platforms for crowdshipping which differ based on their operation type, delivery type, and mode, platforms for food delivery share a lot in common when it comes to operation and pricing. Excluding the platforms that are private or exclusive to restaurants, almost all third-party online food delivery platforms, such as UberEats, Doordash, and GrubHub, belong to type b: the most common platform type in the crowdshipping market, in which platforms match crowds and senders, facilitate navigation, tracking and setting prices (Pourrahmani and Jaller 2021). These food delivery platforms include the items below in their pricing scheme:

- **Menu item price:** this is the price of food the customer ordered from the App. In some platforms this item is subjected to busy or surge pricing. This price is often more than the actual menu price when you order at the place in the restaurant. Restaurant owners increase this price by about 30% to offset their loss due to delivery, such as commissions they pay to delivery apps. It is surprising to learn that some food delivery platforms list non partners restaurants in their App for which they might even increase the menu price and the added profit will directly go to the delivery platform itself (Noah Lichtenstein 2020). In this case, the driver makes an order to the restaurant in place of the actual requester and deliver the order. If the food quality degrades during the delivery process, this will harm restaurant reputation. Analyzing the interrelation between a restaurant and a delivery platform using a queueing model, Feldman, Frazelle et al. (2018) however showed in theory that restaurants still benefit from the presence of a delivery system within a decentralized system without a partnership contract.

- **Delivery fee:** this refers to the price charged for the food to be transported from food place to the customer. This fee is fixed and varies based on region and restaurants. Delivery fees can be waived by some platforms if the order amount exceeds the small order limit. Several platforms such as Doordash, waive delivery fee for members or those orders exceeding the minimum amount threshold. This fee is also subjected to surge (busy time) pricing that is implemented by some platforms when demand is high.

- **Service fee:** this is the fee charged by delivery platform providing the delivery service. This fee is a percent of the cost of menu item price, added to the final bill. There is no clear or consistent percentage rate of service fee across platforms, it varies from 10-18%, while Seamless does not typically charge a service fee (Noah Lichtenstein 2020).
- **Taxes:** sales taxes apply to the order according to the local laws and fees. Observations showed tax rate fluctuation across different apps is about 1.1% (Noah Lichtenstein 2020).
- **Gratuity or tips:** this is an optional fee paid by customer that adds at the top of driver's income. However, recently, Doordash and Instacart were criticized for using tips to partially cover the shipper's base payment. Postmate drivers also stated there is a lack of clarity about which part of their payment comes from tips. In response to this criticism, Instacart updated its compensation policies so that tips from customers are always added to the company's contribution to driver payment (Bernot 2019, Captain 2019).
- **Other fees:** these are the fees charged by restaurants and platforms to customers which are included in some (but not all) of online food delivery Apps:
    - **Merchant fees:** an additional small fee imposed to each order in addition to actual food menu price by certain restaurants to help them offset the profits

lost because of food delivery. In some delivery platforms, this is included in service fees.

- **Small order fee:** this additional fee applies to small or low-price orders to make it economically efficient for delivery. For some platforms, this fee even applies having a membership subscription.
- **Delivery minimum:** a minimum number of order or price limit set by delivery platforms which must be met for delivery request to be approved.
- **Bag fee:** a small fee charged per each reusable bag included with the order. In some delivery platforms, this is included in service fees.

The variation in pricing scheme causes the delivery price for exactly the same item to vary by 20% or more in different platforms. In addition to price variation across Apps, empirical data collected in 2019 showed that consumers are paying 17-41% more when using online food delivery Apps compared to ordering directly from a restaurant (Kinetic 2019)(Figure 1).

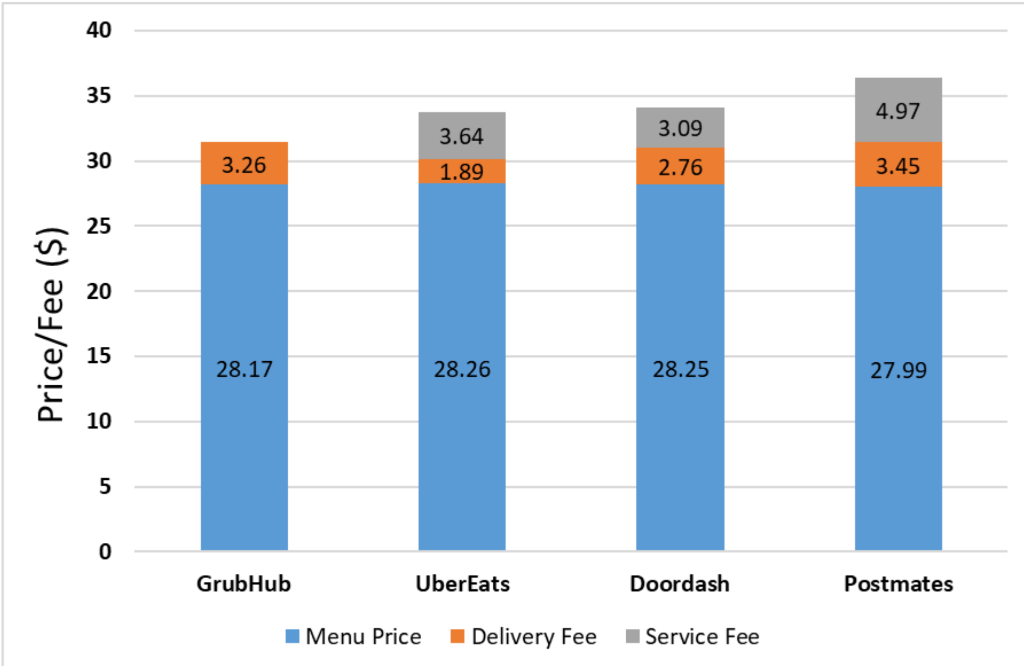


Figure 1 Share of each item in the pricing scheme for each platform (Kinetic 2019)

In all these platforms, the menu price for food was similar and the difference in total cost comes from the additional fees. Depending on which App you order from, consumers might face fees that were 12-30% of the food total cost (Figure 2).

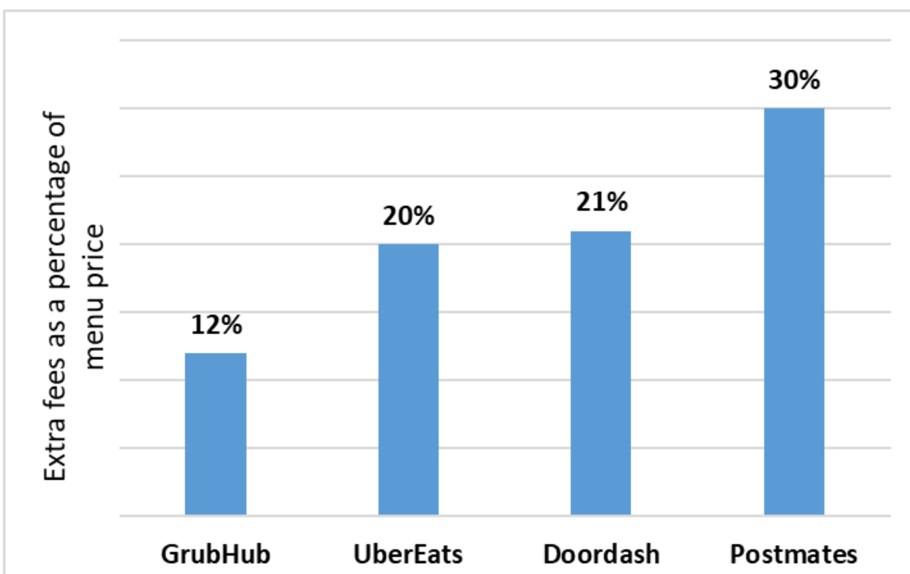


Figure 2 Fees as % of food menu price for each platform (Kinetic 2019)

Pricing items discussed above are subjected to change due to membership and loyalty plans built by platforms to provide more predictable revenue and creating an exclusive network of customers. Such plans typically include a monthly membership fee in exchange for reduced service fees and zero delivery fee for orders value exceeding a certain minimum. Examples are Postmates Unlimited ($9.99/month or $99/year), DashPass from DoorDash ($9.99/month) and Uber program that integrates Eats, rides, bikes and scooter services. Kung and Zhong (2017) studied membership pricing strategy against transaction-based pricing for delivery market. They showed that the two strategies are equivalent in the absence of time discounting and presence of price insensitive customers in terms of order frequency. On the other hand, when the platform is impatient in receiving revenues or consumers' order frequency is affected by the per-transaction fee, membership-based pricing is the most profitable strategy as it enables collecting revenue earlier and maximizing the price-sensitive order frequency by minimizing delivery transaction fees.

3.2 Delivery Pricing in the Research

Delivery fee refers to the price charged for the food to be transported from where it is ordered from (restaurant or any food place) to where it is requested by the consumer. Not enough information is available on the details of the pricing strategy employed by these platforms for delivery. For example, Uber Eats' sometimes low delivery fees is the result of having an established fleet of drivers and logistics expertise derived from the company's core ride-hailing business (Noah Lichtenstein 2020). In general, delivery fees fluctuate based on platform and market depending on location, restaurants, food type or other unknown factors. It might be subjected to surge (busy time) pricing that is implemented by some platforms when demand is high. Surge pricing is dynamic pricing in which the base price increases in real time due to sudden change in proportion of demand and supply. While in passenger ridesharing platforms, surge pricing is explicitly communicated to the riders in the app through a surge factor displayed besides the offered price (e.g., 1.5X, showing that the base price is multiplied by this factor), most food delivery platform users only observe price fluctuations without any clear communication over the reason. This strategy is to stabilize the balance between demand and supply in one or multiple zones. The convention behind surge pricing is to set price in a way to encourage supply to be available where they are needed the most and allocate the limited supply to those demand having higher willingness to pay. Accordingly, the surge pricing increases the base price in zones and times when the demand far exceeds the available supply. It is assumed that this strategy promotes the welfare of the system in general (Guda and Subramanian 2017).

In on-demand transportation services, namely, passenger ride sharing (e.g., Uber and Lyft) and delivery services (e.g., Doordash and Uber Eats), supply refers to the crowds of workers joining the system autonomously to fulfill assigned tasks using their available capacity and receive a compensation fee in return. Demand refers to requesters who submit request based on their mobility or delivery needs and expect their request to be fulfilled given a wait time and cost. The platforms manage the supply and demand by providing capabilities, such as matching, routing and rating, to guarantee efficient and effective service and generate revenue. Platforms typically charge a small percentage of each transaction generated from tasks fulfilled by workers as their commission.

In on-demand transportation services, workers join the service voluntarily without commitment or informing platforms about their work schedule. On the other hand, requesters price sensitivity depends on multitude factors that are susceptible with sudden change in weather condition, special events, or traffic accidents and congestion. These together cause fluctuations in proportion of demand and supply in the system. These factors are difficult to predict or completely unknown. Dynamic pricing strategies can account for the effect of all aforementioned factors without explicitly identifying them. To manage the supply and demand effectively and efficiently, platforms forecast patterns of supply and demand ahead of time, set surge multipliers per different zones accordingly, and communicate this information to drivers about their current and adjacent zones (Chen, Mislove et al. 2015, Rosenblat and Stark 2015). While this is expected to stabilize the system, improve balance between supply and demand and benefit all stakeholders (Cachon, Daniels et al. 2017), the findings from studies using empirical data or analytical models are not always consistent with this.

Pricing studies for on-demand transportation of passengers and goods, surge pricing in specific, are mainly categorized into two domains: empirical or experimental studies and analytical studies. Selected papers and publications at each domain are reviewed briefly below. It is noted that due to smaller number of available research about pricing in delivery services, similar studies from passenger rideshare market are also included as appropriate.

3.2.1 Empirical and Experimental

In (Tong, Dai et al. 2020) the pricing strategy employed by three major food delivery markets in China was identified using empirical data. They found that platforms employing dynamic pricing attract about 100 more orders per hour than static pricing. This gap increases during peak hours. They also indicated that factors such as delivery speed, weather condition, special events and promotions significantly affect the level of demand. Svartbäck and Ekholm (2021) analyzed a sample of restaurant food delivery data and concluded that dynamic pricing of restaurant food delivery can decrease resource waste and improve producer profitability.

While studies about surge pricing in food delivery is rare, there are several published papers about surge pricing in passenger ridesharing market. Using Uber data, Hall, Kendrick et al. (2015) showed the effectiveness of surge pricing in efficient allocation of supply to demand in one single zone in New York. In another study by (Diakopoulos 2015), it is found that surge pricing relocates drivers from adjacent zones rather than motivating new drivers to the system. Later, Chen and Sheldon (2016) found that drivers adjust their working hours to be more active during surge pricing periods. Chen, Mislove et al. (2015) collected Uber pricing and supply/demand data in San Francisco and New York for one week and concluded that this strategy makes drivers become idle in zones with surged price values and cause them to leave these zones. In a study by (Jiang, Kong et al. 2020), drivers relocation decisions were studied from a behavioral perspective. They found that factors such as communicating demand information as well as dynamic subsidies for drivers improve their relocation decisions and lead to system better off. On the other hand (Dholakia (2015) and Rosenblat (2018) raised concerns about the role of trust in the system. While the surge values are communicated to drivers and requesters through surge maps and App offered prices, respectively, significant portion of drivers ignore this information due to lack of trust in the platform. Requesters to leave the system and avoid paying surge price values.

3.2.2 Analytical

Model assumptions and parameter variations distinguish analytical studies from each other. Some papers studied the system with deterministic pricing decisions for fixed spatial zones and considering the variation in supply and demand. Using queue models to consider the interaction between drivers and customers, Taylor (2018) and Bai, So et al. (2019) emphasized the importance of customers' valuations of wait time and price as well as drivers' opportunity costs on optimal price and wage. Choi, Guo et al. (2020) studied the impact of risk on optimal pricing in on-demand platforms. They found that when the customers are risk seeking, the consumer surplus and the platform's expected profit are highest. Nikzad (2017) investigated the impacts of thickness and competition on the equilibria of ride-sharing markets. They found that when the market is sufficiently thick, wage and workers' average welfare decrease with size of the labor pool and vice versa. They showed that effective matching impacts labor like increasing the labor pool in a thin market otherwise reduces their wage and average welfare. Comparing the monopoly and duopoly equilibria, they found that competition benefits drivers, however, the effect of competition on customers' average welfare depends on thickness. When the market is not sufficiently thick, price is higher, and customers' average welfare is lower.

Cachon, Daniels et al. (2017) studied different pricing schemes for Uber/Lyft platforms. Although surge pricing is not optimal, they showed it achieves nearly the optimal profit and drivers and riders are better off with surge pricing with variations in supply and demand ratio. Banerjee, Riquelme et al. (2015) built a queueing-theoretic economic model to study optimal platform pricing and found that static pricing is near optimal when there are high volume of demand and supply in the system. In studies by (Banerjee, Riquelme et al. 2015, Castillo, Knoepfle et al. 2017), it is showed that dynamic surge pricing supports system stability during supply and demand fluctuations. Tang, Bai et al. (2016) employed steady state equilibrium in a queuing model in which requesters and drivers' arrival time depend on offered prices and wages. Using a similar model, Banerjee, Riquelme et al. (2015) found that price is raised with increasing mismatch in supply and demand. They showed that dynamic pricing leads to a higher revenue than static pricing if only demand parameters are unknown to the platform. In another study by (Gurvich, Lariviere et al. 2019), independent working schedule reduces the supply in on-demand transportation services, which increases the offered prices to customers in return. In a recent study by (Garg and Nazerzadeh 2021), the authors studied the impact of a dynamic surge pricing model on drivers earnings. Based on numerical analysis and real data, they found that the additive surge is more incentive compatible compared to multiplicative surge. Prokhorchuk, Dauwels et al. (2019) combined the optimization of routing and pricing for same-day delivery considering uncertainty in travel time. A value function approximator was trained to estimate the opportunity cost of accepting a request by drivers. Their findings emphasize the presence of travel time information on pricing decisions. When there was penalty for missed deliveries, the delivery prices were higher. The difference in prices between the penalty and non-penalty situations is significantly higher when supply level was low.

In a location-based pricing study, Chen, Li et al. (2015) studied the competitive implications between firms where customers can move between locations and firms. They found this pricing strategy is effective to control price variation between zones and maintain reasonable consumers in the system. Guda and Subramanian (2017), and Guda and

Subramanian (2019) analyzed the role of surge pricing and forecast communication in an on-demand platform and showed that surge pricing is useful when there is a mismatch between demand and supply. Based on their finding, price distortion by platform eventually increase profit by incentivizing drivers to leave over supply locations. Afeche, Liu et al. (2018) studied the demand approval and drivers repositioning decisions using a queueing network model. They provided sufficient conditions under which it is optimal to cancel demand at a low-demand location and encourage drivers to relocate to a high-demand location. Besbes, Castro et al. (2021) proposed a two-dimensional framework where the platform sets different prices in different locations considering prices, travel costs, and driver congestion levels. They showed optimal pricing stabilize supply and demand in some locations, while congestion is induced in others. Also, less profitable locations are indirectly priced out to incentivizing the relocation of drivers towards regions that are more beneficial.

According to reviewed studies, the majority of pricing strategies in transportation shared economy belongs to passenger rideshare market proposing highly analytical mathematics and economic models that might not be applied. They rarely consider supply and demand elasticities at the same time with pricing and matching decisions. Furthermore, they lack flexibility of simulation models that enables studying the behavior of different actors in the system and under different settings.

This study formulates and solves a dynamic zone-based pricing for a many-to-many food delivery problem including demand elasticity and supply preferences while making pricing and matching decisions by platform. The simulation model provides enough flexibility to model three actors in the system: platform, delivery drivers and requesters and study the impacts of their decisions in various scenarios. To estimate the requesters elasticity as a function of price and wait time, a conjoint analysis survey is conducted (Subsection 3.3). The dynamic zone-based pricing is formulated as a dynamic programing problem (Subsection 3.4) and solved by building a simulation-optimization model based on reinforcement learning techniques (Subsection 3.5). The next three subsections present background for conjoint analysis, dynamic programing, and reinforcement learning.

3.3 Conjoint Analysis

Food delivery apps become increasingly popular among people as an average person has two food delivery apps and use them three times per month (Lardieri 2019). While more people are using food delivery apps, there is a maximum amount they are willing to pay for the service and a limited time they are willing to wait for the food to be delivered to them. Speed of delivery is the key factor in customer satisfaction (Carsten Hirschberg 2016). A survey conducted by U.S. Foods (Lardieri 2019) found that more than 70% will spend no more than $10 on delivery fees, service fees and a tip. Same survey found that most people don't want to wait longer than 40 minutes on average for their ordered food to be delivered. Accordingly, fees and wait time for food delivery play critical role in requesters' decision to order food and eventually determine the level of service and sustainable use of such services.

Conjoint analysis is a statistical strategy and decomposition method to study the joint effect of attributes in products/services influencing consumers' choice decisions to buy a product/service based on trade-offs. This assumes that the utility of an alternative could be decomposed into partial utilities of its attributes (Rao, Rabinovich et al. 2014). In the

Conjoint Analysis survey, individuals express their preferences among alternatives characterized by a set of attributes.

Conjoint theory was founded in the 1960s by (Luce and Tukey 1964) and since then, it has been developed by researchers working in the field of data analysis primarily in behavioral sciences for marketing research methods (Kruskal 1969, Jain, Mahajan et al. 1979, Acito and Jain 1980). To measure consumers' preferences to choose product alternatives in behavioral research, there are two approaches: revealed preferences (RP) and stated preferences (SP), assuming that consumers' preferences and perception depend on the process of utility maximization. In the former, consumers choice behavior is observed in a real context by researchers. For the stated preference, on the other hand, individuals respond what would they do when they are exposed to a given choice set in a hypothetical context. In both cases, a researcher can study the outcome of a choice set and discover the preferences of consumer by the outcome of a choice experiment.

Conjoint analysis is a stated preference analysis of consumer's preferences and tradeoffs among products/services. Respondents are presented with different hypothetical alternatives in a fractional factorial design, that are mutually exclusive and are asked to score or rank them according to their order of preference. The data from this type of study can provide information on the probability that the consumer will choose or not choose any of those hypothetical alternatives. The data can further be used to analyze factors that contribute to the willingness to pay for a chosen product. Multinomial logit models, such as binary and ordinal logit models, are helpful to analyze the data from a conjoint experiment.

Conjoint analysis is founded on Random Utility Model (RUM)(Lancaster 1966). Utility is an indicator of the value a potential consumer places on an alternative service/product. Under RUM, the utility of each alternative of a product is a linear function of the observed characteristics of the product plus the error term (Verbeek 2008). Thus, individuals' utility for an alternative has two parts: a deterministic and a random part. The former depends on the attributes of a product/service and the latter is random and unpredictable precisely, indicating the effects of unobserved attributes or taste variations. If individuals select the alternative with the highest utility, the probability of choosing an alternative could be examined by calculating the overall utility for individual $\boldsymbol{s}$, $\boldsymbol{u_s}$, as a function of preferences, using Eq. 3.1:

$$\boldsymbol{u_s = \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} x_{ijs} + \varepsilon_{ijs}} \qquad (\mathbf{3.1})$$

where $\boldsymbol{a_{ij}}$ indicates the part-worth utility for $\boldsymbol{j}$th level of attribute $\boldsymbol{i}$. In other word, it measures the influence of the attribute $\boldsymbol{i}$ on the utility when it is at its $\boldsymbol{j}$th level. Here, Levels are the 'values' that each attribute can take. For example, the attribute 'time of the day' can have levels 'AM', 'MD', and 'PM'. $\boldsymbol{x_{ijs}}$ is a binary variable indicating the presence of $\boldsymbol{j}$th level of attribute $\boldsymbol{i}$ for individual $\boldsymbol{s}$.

Different types of conjoint analysis are available, the application of each depends on many factors such as the utility function type, sample size, and most importantly the number of attributes and their associated levels. Rating-based conjoint analysis is the most traditional one in which respondents indicate their preference toward a set of combined attributes by scoring with respect to a scale (e.g., 0-10, 0-100, etc.). In rating-based, analyses are usually based on linear regression. On the other hand, choice-based conjoint analysis is the most popular type in which respondents are exposed to a set of potential products/services characterized by a combination of attributes. They complete sets of choice tasks(less than the full profile) and the analysis is either based on Logistic Regression or Hierarchical Bayes (Louviere 1988).

3.4 Dynamic Programming

3.4.1 Description

Dynamic programing (DP) is a multi-stage decision making process which attempt to solve the problem sequentially in multiple stages. DP has five main elements: *stage*, *state*, *transition, reward,* and *recursive relation*. Each stage either represent a time epoch or decision step. State is a critical component in DP which intends to provide enough information required to take an action at that state. The state variable varies by problem and must be selected carefully to communicate the necessary information for decision making. Size of state variable must be reasonable enough to solve the DP in reasonable time as it is a computationally intensive approach. Actions are selected at each state based on a given policy. Transition maps the current state to the next state because of action. Reward is the value gained by taking the action at that state. The rewards at each stage contribute to the total expected value of the followed policy.

Recursive relation is fundamental to DP approach. In recursive procedure, DP attempts to solve the multistage problem by a N-stage solution created sequentially solving an optimization problem one stage at a time until the overall optimum is found. The optimization problem is solved with respect to minimizing/maximizing the problem objective. This recursive procedure can either be backward or forward. In the former, the procedure starts with the last stage and continues backward, one stage at a time, until the total stages are included. The latter though starts from the initial stage and move forward solving each stage of problem sequentially at a time.

In DP, the N-stage solution is a policy, which determines the optimal action at each stage with respect to problem objective. According to the principle of optimality, any current state is followed by actions constituting an optimal policy (Čepin 2011).

DP computes the optimal policies given a perfect finite Markovian Decision Process (MDP). In a finite MDP, the set of states, actions and rewards are finite. Transition rule which defines the problem dynamic is a model in the form of a function (deterministic) or probability distribution (stochastic) that maps the current state to the next state and reward, given the action to be taken at a decision step (Bellman 1957, Lapan 2018). Perfect Markov property means that the transition rule model completely includes the information necessary to make a decision from the current state and action, not a chain of earlier states and actions(Sutton and Barto 2018).

3.4.2 Formulation

The mathematical formulation of DP is presented here. The objective or value function of a typical multistage DP is defined as $\boldsymbol{V(s_n)}$ at stage ***n.*** Transition function is defined as $\boldsymbol{p(a_n, s_n)}$ which indicates the dependency of next state entirely to the current state and action taken (Eq. 3.2). Here, $\boldsymbol{a_n}$ refers to the action taken from all available actions in set $\boldsymbol{A_n}$ and $\boldsymbol{s_n}$ represents the state of the problem including necessary information for decision making and transition.

$$\boldsymbol{s_{n+1} = p(a_n, s_n)} \qquad (\mathbf{3.2})$$

In this equation $\boldsymbol{p}$ can represent a probability function such as $\boldsymbol{p}(\boldsymbol{s_{n+1}}, \boldsymbol{r}|\boldsymbol{a_n}, \boldsymbol{s_n})$ by which all possible next states and rewards are swept given the current state and action. Then the expected return at every stage and state is calculated as the sum of the value of all next transitions weighted by their probability of occurrence.

At each stage $\boldsymbol{n}$, given the current state $\boldsymbol{s_n}$, the aim is to select an action $\boldsymbol{a_n}$ from a set of possible actions $\boldsymbol{A_n}$ that optimize the immediate reward $\boldsymbol{r}$ and total value over the remaining stages, $\boldsymbol{V}(\boldsymbol{s_n})$, and results in a new state, $\boldsymbol{s_{n+1}}$, using $\boldsymbol{p}(\boldsymbol{a_n}, \boldsymbol{s_n})$, with $N-n$ stages to go. Thus, the optimal value function at stage $\boldsymbol{n}$ is given by Eq. 3.3 which is the principal of optimality. In this equation, $\boldsymbol{\gamma} \in (\boldsymbol{0}, \boldsymbol{1}]$, is a discount factor to reflect the present value of a future reward.

$$\boldsymbol{V}(\boldsymbol{s_n}) = \max_{\boldsymbol{a_{n+1}}}\left(\sum_{s_{n+1},r} \boldsymbol{p}(\boldsymbol{s_{n+1}}, \boldsymbol{r}|\boldsymbol{a_n}, \boldsymbol{s_n})\,[\boldsymbol{r} + \boldsymbol{\gamma}\,\boldsymbol{V}(\boldsymbol{s_{n+1}})]\right) \qquad (\mathbf{3.3})$$

To solve this multistage problem, generalized policy iteration algorithms, known as classic DP algorithms, have been proposed. These algorithms generally are divided into two essential parts: policy evaluation and policy improvement. In the former, the aim is to evaluate the value of the current policy $\boldsymbol{\pi}$ comprised of set of actions taken at consecutive states. This is done using Bellman equation (3.4) iteratively. This means to calculate the value of the next state by backing up the value of the current state from the previous iteration.

$$\boldsymbol{V_k}(\boldsymbol{s_n}) = \sum_{s_{n+1},r} \boldsymbol{p}(\boldsymbol{s_{n+1}}, \boldsymbol{r}|\boldsymbol{\pi}(\boldsymbol{a_n}|\boldsymbol{s_n}), \boldsymbol{s_n})\,[\boldsymbol{r} + \boldsymbol{\gamma}\,\boldsymbol{V_{k-1}}(\boldsymbol{s_{n+1}})] \qquad (\mathbf{3.4})$$

The latter, policy improvement, follows the policy evaluation and improves the current policy by acting greedily to the evaluated value function to find a better policy. These two parts run iteratively until policy is stable and convergence is achieved to optimal value function and optimal policy. The pseudo code of generalized policy iteration algorithm is displayed below in Figure 3.

1 Initialize V*(S) and* π*(S)* for all *s*∈*S;*
2 Policy Evaluation:
3 Repeat
4 $\Delta \leftarrow 0$
5 For each S:
6 $v \leftarrow V(S)$
7 $V(S) = \sum_{s_{n+1},r} p\left(s_{n+1}, r \middle| \pi(a_n|s_n), s_n\right) [r + \gamma\, V(s_{n+1})]$])
8 $\Delta \leftarrow max(\Delta, |v - V(S)|)$
9 until $\Delta < \theta$ (a small positive number)
10 Policy Improvement:
11 policy-stable ← *True*
12 For each S:
13 old-action ← π*(S)*
14 $\pi(S) \leftarrow argmax_{\,a} \left(\sum_{s_{n+1},r} p(s_{n+1}, r|a_n, s_n) [r + \gamma\, V(s_{n+1})]\right)$
15 If old-action at state $S \neq \pi(S)$, then policy-stable ← *False*
16 If policy-stable, then stop and return optimal *V* and π; else go to **2**

Figure 3 Generalized policy iteration algorithm (Sutton and Barto 2018)

To solve DP using policy iteration algorithms, all states must be swept, and a small optimization problem must be solved at every iteration. While there are extension algorithms aiming to improve its efficiency, policy iteration algorithms are still computationally expensive and suffer from several shortages which limit their application on large real-world problems. For the problem to be solvable, action and state spaces must be discrete with reasonable size. Strategies such as state aggregation can mitigate the continuous state spaces issues, however, this came with loss of precision and additional hyperparameters need to be carefully tuned. Besides, DP requires the presence of a perfect probability transition function ***p*** which is not possible in many real problems contexts. Furthermore, estimating value functions by backing up the value of the current state using Bellman equations requires the MDP condition to meet, while in several problems this does not hold or only partially hold.

3.5 Reinforcement Learning

Reinforcement Learning (RL) is a specific branch in machine learning (ML) which deals with optimal multi-stage decision making through automated learning from interactions in a changing environment over time. While DP and Bellman equation are fundamental in RL, it does not have their obligations. RL enables learning from experience without necessarily having a transition model which is called Direct RL. It features advanced methods to generalize learning in large continuous spaces without computational burden. Finally, while many RL algorithms still require MDP condition to hold, there are strategies available to detour.

RL approach stands between supervised learning and unsupervised learning, which are two well studied fields in ML (Lapan 2018). The former refers to building a function that automatically maps some input into some output. The function is built by training its parameters given a set of labeled pairs of inputs and outputs. Regression is an example of supervised learning methods. Predicting car sales prices given the car features such as weight, engine type, model, etc. is an example of regression prediction. On the other hand, unsupervised learning assumes no labeled output exist and learning happens based on discovering hidden patterns in the data structures. Clustering customers based on a set of attributes in the marketing research is an example of unsupervised learning.

To understand the state of RL with respect to the two ML strategies just described, let's consider an example of a small robot navigation problem. In this problem, a robot aims to find the shortest path to a predefined destination. The robot can take actions including moving, such as turn left/right and move forward, and recharge at each decision step. At every step, it can observe the state of the environment to decide about the actions to take next. The observations might include the level of its electric charge and the immediately connected nodes on the route. It is trying to find the shortest path to the destination. The reward can be defined as being proportional to the remaining distance to the destination given to the agent by the environment as feedback to its actions. The reward system is fundamental to RL and differentiates it from unsupervised learning in which no predefined label exist. This system enables the agent to learn automatically by relating the environment's feedback to the actions taken and accumulate the learned experience to improve its next actions. Learning the relation between action and the feedback makes RL more difficult compared to supervised learning in which labeled pairs of input and output are already available.

One challenge in RL is that the source of information depends on agents' actions. Thus, persisting on inefficient actions by the agent returns bad impulse to the learning procedure and might realize wrong decisions. Sustaining the balance between exploration and exploitation is another important factor. While the agent needs to exploit actions that result in better reward, it also needs to actively explore the environment to find undiscovered regions which might possibly realize significantly improved actions and more rewards. Finally, the design of an effective reward system avoid excessive decision steps and accelerate the agent's learning accomplishment (Lapan 2018, Sutton and Barto 2018).

3.5.1 Elements

To formally introduce RL, we need to define its elements. There are two fundamental entities in RL: Agent and Environment. Agent is an entity which is responsible of observing the environment state at each decision step $\boldsymbol{s}$, play roles during each simulation episode $\boldsymbol{a},$ and receive a feedback from the environment $\boldsymbol{r}$ (Figure 4). The environment is the context where the agent is living and playing. These two entities interact with each other through three mechanisms: observation, action, and reward (Figure 5).

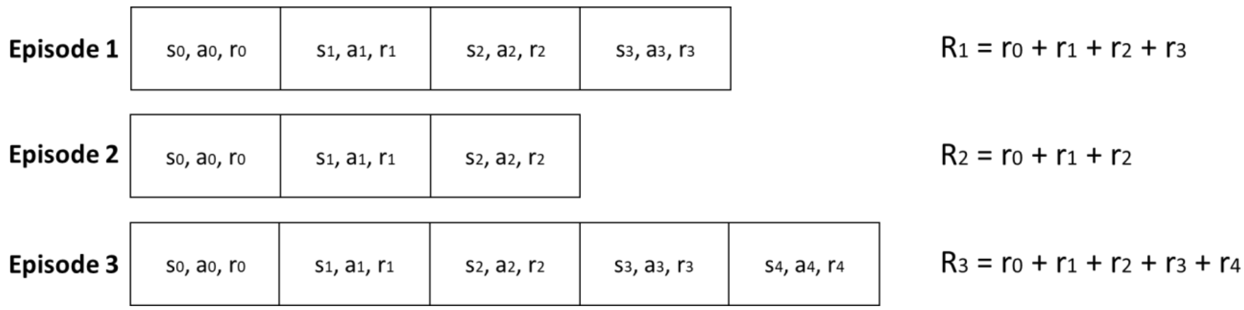


Figure 4 Sample RL episodes

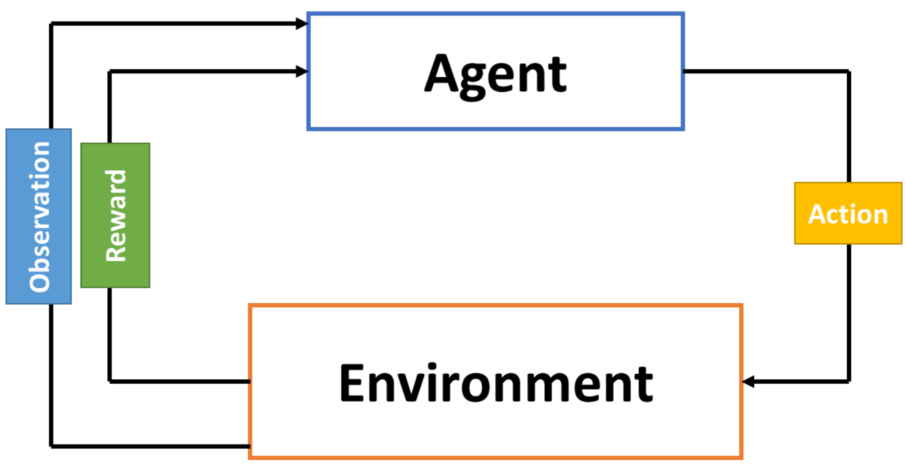


Figure 5 RL diagram

Observations is amount of information an agent receives from its environment. In other words, observation $\boldsymbol{s_t}$ is an $\boldsymbol{N}$ dimensional informational subset of environment $\boldsymbol{E}$ at each decision step $\boldsymbol{t}$, required by the agent to act. Action $\boldsymbol{a_t}$ refers to what agent executes in the environment to progress with each decision step $\boldsymbol{t}$. The consecutive sequence of actions $\{\boldsymbol{a_0}, \boldsymbol{a_1}, \boldsymbol{a_2}, \dots, \boldsymbol{a_T}\}$ at time steps $\boldsymbol{t} \in \boldsymbol{T}$ during an episode is determined by policy, $\boldsymbol{\pi}$. In RL, policy is a set of rules to control the agent's behavior. Policy tells the agent what actions to take at each state. Policy can be deterministic and depend only on the current state $\boldsymbol{\pi}(\boldsymbol{s_t})$ or can be in the form of a probability distribution over actions at each state $\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{s_t})$**.** Both action and observation space can be discrete or continuous, finite or infinite, including any dimension. While more information about environment and actions can improve accuracy, it comes with high computation and maintenance cost. Thus, identifying critical features and an appropriate dimensional presentation are important factors to consider.

In RL, reward is a scalar (negative or positive) that is obtained locally with the objective to reinforce agents' behavior in a positive or negative way. Local reward is obtained after the most recent action the agent has taken in a given state according to reward function $\boldsymbol{\chi}(\boldsymbol{S_t}, \boldsymbol{a^t})$ and does not necessary guarantee it sustains in the next steps. The reward can be obtained periodically (not necessarily after each decision step $\boldsymbol{t}$) depending on underlying problem definition and assumption. The quality of a policy is measured by the cumulative reward an agent gain across a sequence of decision steps in an episode. An agent aims to maximize this cumulative reward in every episode (Eq. 3.5). At each decision step $\boldsymbol{t}$, $\boldsymbol{R_t}$ refers to the rewards gain from $\boldsymbol{t}$ up to terminal step, $\boldsymbol{T}$.

$$\boldsymbol{R_t = r_t + r_{t+1} + r_{t+2} + r_{t+3} + \cdots + r_T} \qquad (\mathbf{3.5})$$

The reward of next decision steps might have different present value. Thus, a discount factor, $\boldsymbol{\gamma}$, might be applied. A scalar less than one (usually 0.9 or 0.99) also provides a limit to the infinite decisions steps horizon we calculate values for (Eq. 3.6).

$$\boldsymbol{R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k}} \qquad (\mathbf{3.6})$$

As discussed before, an agent can follow various policies in an episode, but not all policies are necessarily optimal. Value function is a way to measure the goodness of a policy. It can be ether a function of the states, ***V(S)*** (Eq. 3.7) or a function of both states and actions, ***Q(S,***

A) (Eq. 3.8). The value of a policy at each state s_t is measured as the expected total rewards the agent gain following that policy π from that state.

$$V_\pi(S = S_t) = E_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,\middle|\, S = S_t\right) \qquad (3.7)$$

$$Q_\pi(S = S_t, a = a_t) = E_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,\middle|\, S = S_t, a = a_t\right) \qquad (3.8)$$

An optimal policy, π^*, is the policy which has the better expected total reward compared to any other policy π, for every state: $V_{\pi^*}(s) \geq V_\pi(s), \forall s \in S$. The same thing applies using **Q** function (Eq. 3.9-3.10).

$$V_{\pi^*}(s) = \max_\pi V_\pi(s), \forall s \in S \qquad (3.9)$$

$$Q_{\pi^*}(s, a) = \max_\pi Q_\pi(s, a), \forall s \in S, \forall a \in A \qquad (3.10)$$

Using DP properties, Bellman optimality equations can be derived from Eq. (3.9-3.10) by recursive formulation as Eq. (3.11-3.12).

$$V_*(S = S_t) = \max_a E_{\pi^*}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,\middle|\, S = S_{t,}, a_t = a\right)$$
$$= \max_a \sum_{S'} p(S_{t+1} = S' | S_t, a_t = a)[\chi(S_t, a^t = a) + \gamma V_*(S_{t+1} = S')] \qquad (3.11)$$

$$Q_*(S = S_t, a = a_t) = E(r_t + \gamma \max_{a'} Q_*(S = S_{t+1}, a_{t+1} = a')$$
$$= \sum_{S'} p(S_{t+1} = S' | S_t, a_t = a)[\chi(S_t, a^t = a) + \gamma \max_{a'} Q_*(S = S_{t+1}, a_{t+1}$$
$$= a')] \qquad (3.12)$$

In the above equations, functions P and χ are transition probability and reward functions, respectively. Many of the algorithms that are used in RL are model free and assume these functions are unknown. These methods estimate policy and value functions by executing policies and simulating many episodes. The agent plays and evaluates many possible actions in different states to learn the optimal policy leading to maximum total expected reward. These methods are further discussed next.

3.5.2 Algorithms

In the history of RL, various methods and strategies are developed featuring different characteristics. Value-based methods estimate the value of return for each action and determine the best action to take at every step accordingly. Policy-based methods aim to approximate the policy for the agent to make decision about which actions to carry out at each step. In policy-based methods, policy is usually represented as a probability distribution over actions at each state. Another distinguishing feature among various methods lies in their update strategy. On-policy methods update the value function based on the current most recent policy which is also used to guide agents' behavior. On-policy methods require fresh data to be collected from the environment or simulation in real-time. On the other hand, off-policy imposes the ability to learn from old historical data. Off-policy choses a different policy than the agent's policy to update the value function.

As mentioned earlier in Section 3.5, exploration and exploitation play important roles for action selection in RL methods. The challenge is over the decision to exploit an already found rewarding region or to explore unknown regions to discover better policies. Maintaining an appropriate balance between these two decisions highly improves the agent's learning process. Some methods are employed so far respecting exploration-exploitation balance. The simplest and the most used method is $\varepsilon - greedy$ in which the next action is selected greedily with probability of **1-** $\boldsymbol{\varepsilon}$ ($\boldsymbol{\varepsilon} \in [\mathbf{0}, \mathbf{1}]$) and it will be chosen randomly, otherwise. An effective modification to this strategy is to decrease the value of $\boldsymbol{\varepsilon}$ during training process to reinforce explored learning during initial episodes and prioritize exploitation over exploration during later episodes. Another modification strategy for exploration-exploitation is Boltzmann, or the soft-max selection method which uses the following equation (Eq. 3.13) as an alternative probability of selection to **1-** $\boldsymbol{\varepsilon}$:

$$\boldsymbol{p(a_t|s_t, Q_t) = \frac{exp^{\frac{Q(s_t,a_t)}{\tau}}}{\sum_A exp^{\frac{Q(s_t,a)}{\tau}}}} \qquad (\mathbf{3.13})$$

where $\boldsymbol{\tau}$ is Boltzmann temperature from thermodynamics, which decreases proportionally with episodes progress. Higher values of $\boldsymbol{\tau}$ lead to equiprobable actions and random selection, to emphasize exploration during initial episodes. As $\boldsymbol{\tau}$ decreases, selection becomes greedier, giving higher chance to better actions.

3.5.2.1 Value based methods

Monte Carlo

Monte Carlo (MC) methods learn value functions and optimal policies from experience in the form of sample episodes. They originally follow the overall procedure of generalized policy iteration algorithm explained in Section 3.4.2. As an alternative to use a model, MC performs policy evaluation by interacting with the actual or simulated environment and accumulate the total reward gained over the entire episode. Keeping track of the observed states, the expected return for each state is estimated as the simple average of many returns that start in that state. MC methods require large number of simulation executions to converge to the optimal value function and policy due to high variance (Sutton and Barto 2018).

In addition to the freedom to learn from model or sampled experiences, MC has other advantages to classic DP methods. MC enables efficient learning by focusing on a subset of states which are useful rather than going to the expense of accurately estimating the rest of states which are not important. Moreover, MC is less vulnerable to Markovian property violation. This is because it estimates the value function at each state using the total reward gained over the entire episode, rather than bootstrapping from estimated values of successor states.

Temporal difference

Temporal difference (TD) modifies the idea behind policy evaluation part of MC to form another procedure in the generalized policy iteration framework. Instead of using the total accumulated reward in an episode (same as in MC), TD computes the difference of the new estimate of the value function at any observed state and its old value for the same state, calling it temporal error, to update the value function (Eq. 3.14).

$$\boldsymbol{V}(\boldsymbol{s_t}) = \boldsymbol{V}(\boldsymbol{s_t}) + \propto [\boldsymbol{R} + \boldsymbol{\gamma} \boldsymbol{V}(\boldsymbol{s_{t+1}}) - \boldsymbol{V}(\boldsymbol{s_t})] \qquad (\mathbf{3.14})$$

where $\boldsymbol{s_t}$ and $\boldsymbol{s_{t+1}}$ are the current and next states, respectively. $\boldsymbol{R}$ is the immediate reward gained by transitioning to the new state, and $\propto$ is the learning rate. Since TD updates the value functions at every transition (or in other words, with respect to shorter decision steps), discount factor $\boldsymbol{\gamma}$ can be set to 1. TD methods converge faster than MC due to lower variance, but it increases the bias in the estimate of the value function (Sutton and Barto 2018).

SARSA algorithm (Figure 6), which has its name from State-Action-Reward-State-Action, is an on-policy strategy that has been used widely in RL and has adopted TD method to update the action-state value function, **Q** (Eq. 3.15).

$$\boldsymbol{Q}(\boldsymbol{s_t}, \boldsymbol{a_t}) = \boldsymbol{Q}(\boldsymbol{s_t}, \boldsymbol{a_t}) + \propto [\boldsymbol{R} + \boldsymbol{\gamma} \boldsymbol{Q}(\boldsymbol{s_{t+1}}, \boldsymbol{a_{t+1}}) - \boldsymbol{Q}(\boldsymbol{s_t}, \boldsymbol{a_t})] \qquad (\mathbf{3.15})$$

```
Initialize Q(S,A) for all s∈S; a∈A(s);
Repeat for each episode:
    Initialize S_0
    Choose a_0 from S_0 using Q
    For t = 0, 1, 2, … ,T:
        Take action a_t, observe R and S_t+1
```

Choose a_{t+1} from S_{t+1} using Q
$Q(s_t, a_t) = Q(s_t, a_t) + \propto [R + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$
$S_t \leftarrow S_{t+1},\ a_t \leftarrow a_{t+1}$

Figure 6 SARSA (on-policy TD method)

In 1989, Watkins proposed a slight modification to SARSA by redefining its value function update rule and called this new version as Q learning (Figure 7)(Watkins and Dayan 1992). Unlike SARSA, Q-learning is off-policy since Q function is approximated greedily, independent from the current policy it is executing. In its update rule, Q-learning replaces the next state Q value, which was derived from the current policy execution experience ***[s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}]*** in SARSA, by the maximum future value (Eq. 3.16). This maximum value is usually computed by employing $\varepsilon - greedy$ strategy.

$$\boldsymbol{Q(s_t, a_t) = Q(s_t, a_t) + \propto [R + \gamma \max_{a} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]} \quad \textbf{(3.16)}$$

Initialize $Q(S,A)$ for all $s \in S;\ a \in A(s)$;
Repeat for each episode:
  Initialize S_0
  For t = 0, 1, 2, … ,T:
    Choose a_t from S_t using Q
    Take action a_t, observe R and S_{t+1}
    $Q(s_t, a_t) = Q(s_t, a_t) + \propto [R + \gamma \max_{a} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$
    $S_t \leftarrow S_{t+1}$

Figure 7 Q-learning (off-policy TD method)

Function approximation

The value functions estimated using any of the above discussed methods are stored in a tabular format for every state or every pair of state and action. These methods can only handle problems with reasonable size with low dimension states and actions. Otherwise, it will be memory inefficient to store all cases or it will be computationally expensive to sweep all possible combinations of states and actions to estimate the value function. Additionally, not every pair of states and actions will be realized in the environment, thus sweeping all combinations might be waste of time. On the other hand, if the states and actions are of continuous spaces, it is impossible to use tabular format.

To deal with the above-mentioned cases, parametrized function approximators replace the traditional tabular format storage and representations. In this way, the value function depends on features defined according to states and actions. These features are weighted by a vector of parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)^T$ and ultimately is denoted as $\mathbf{V(s; \theta)}$ or $\mathbf{Q(s,a; \theta)}$, depending on the problem model. The function approximator projects from a vector of parameters, $\boldsymbol{\theta}$ in $\boldsymbol{n}$ dimensions, to the space of the value function. The size of parameter vector is smaller than the entire state or state-action features. This parameter vector must be trained based on enough simulation episodes to generalize information from samples of experiences. Machine learning algorithms such as linear function approximators or Neural Network (NN) might be used in this regard to approximate value functions. Strategies such as tile coding and radial basis function can be employed to further aggregate the state space. However, these strategies often make more problems than they solve as there are more hyperparameters to decide about the level of aggregation, and ranges of parameters to distinguish different states and actions. Compared to the other methods, NN is capable in representing complex functions with less parameters. While NN are widely used as non-linear function approximators nowadays, they are not promised to be converged.

3.5.2.2 Policy based methods

Methods discussed in the previous section were value oriented. They iteratively estimated the value of state or state-action pair and guide the agent behavior by acting greedily in terms of value (Eq. 3.17). Thus, the selected action at each state is the one with the largest Q value.

$$V_{\pi^*}(s_t) = Q_{\pi^*}(s_t, a_t) = \underset{a}{\mathbf{argmax}}\, Q_{\pi}(s_t, a_t),$$
$$\forall s_t \in S, \forall a_t \in A \qquad (3.17)$$

Value based methods determine the policy indirectly using values. On the other hand, policy-based methods parametrize the policy π_θ , usually in terms of the probability distribution of actions, with respect to a vector of parameters $\theta = (\theta_1, \theta_2, \ldots, \theta_n)^T$. Policy evaluation is performed by executing current policy and update the total reward. Gradient descent, algorithms family are used to find the parameters in the direction to increase the probability of actions with better total rewards.

Policy search has better convergence properties compared with value-based approaches. Their major drawback is that their policy evaluation step imposes large variance and fails in learning good policies. Thus, it requires a large number of interactions with the environment which makes it undesirable. Several methods have been proposed in the literature to reduce the variance in policy search strategies (Lapan 2018).

3.6 Summary

This chapter presented background knowledge about the current pricing in online food delivery and reviewed the literature about surge pricing in transportation shared economy. While the empirical data analysis was eye opener about various items in delivery pricing in four large delivery apps, more insights are needed about temporal change in delivery fees and the impact of delivery region on price in various food apps. Research papers about pricing in transportation shared economy were highly analytical or hypothetical. They lack the flexibility needed to evaluate various scenarios and observe the interaction among actors in the system. While dynamic surge pricing supposed to benefit the system in theory, empirical findings indicate that drivers are idled in the surge zones and there is lack of trust in announced surge multipliers from users to follow. The reviewed studies rarely evaluated the service externality on environment and health.

This chapter also described the RL background and clarifies its place among other ML methods, similarities, and differences. Then, it introduced the general RL framework and describes its basic elements and operation mechanisms. Later, various fundamental algorithms to solve some RL problems are introduced and summarized.

4. Data Collection and Analysis

As part of this study, data was collected from food delivery apps to gain more insight into the pricing strategies in various online food delivery platforms. In addition, a survey was conducted on a sample of individuals to understand their food delivery habits and preferences and eventually estimate their elasticity function. These functions are later used in the pricing model.

4.1 Food Delivery Data Collection and Analysis

To better understand online food delivery pricing and how the mechanism differs in different platforms, regions, and restaurants, this study collected 1,080 data records manually (using personal smart phones) from four food delivery apps in the U.S.: UberEats, Doordash, Postmates, and GrubHub. Data was compiled from mid-July to mid-August 2021 from 15 restaurants and cafes available on aforementioned apps located in the northern California regions of Davis, Sacramento (Sac), and San Francisco (SF). Davis is a small city with educated and environmentally aware residents. On the other hand, SF is a large metropolitan area and the birthplace of many on-demand crowdshipping services (e.g. Doordash and Uber). SF is globally known as a center for innovation and internet technology. Finally, Sac is the Capital of the State, stands between Davis and SF in terms of population size. Sac is a fast-growing major city in the State with culturally diverse population. For each region, three fixed locations were chosen as food delivery requesters to capture the effect of delivery distance and time on price and wait time. These locations are selected in different regions within the city. Pricing details and wait time data were collected for all pairs of requesters and restaurants at different times of the day (morning [AM], midday [MD], and evening [PM]) on weekends and weekdays. To explore the effect of food type, we selected five types of popular meals for data collection: Mexican, Indian, American fast casual, American fast food, and café drinks (e.g., coffee and tea). While the initial goal was to consider exactly the same restaurants for each food type, not all restaurants of interest were available in all regions or apps. For example, Panera Bread was not located in the core area of SF, and Starbucks was not operating on GrubHub app. Thus, other similar restaurants were considered as replacements for these cases. Table 3 summarizes the restaurants for which food delivery data was collected. Note that this study did not validate items such as food menu cost or restaurant working hours physically in place and that all reported numbers are based on what data the apps provided.

Table 3 Restaurant information

Food Type	Restaurants											
	UberEats			Doordash			Postmate			GrubHub		
	Davis	Sac	SF	Davis	Sac	SF	Davis	Sac	SF	Davis	Sac	SF
Mexican	Chipotle	Chipotle	Chipotle	Chipotle	Chipotle	Chipotle	Chipotle	Chipotle	Chipotle	Chipotle	Chipotle	Chipotle
Indian	The halal guys	Preethi Indian cuisine	The halal guys	The halal guys	Preethi Indian cuisine	The halal guys	The halal guys	Preethi Indian cuisine	The halal guys	The Halal Guys	Tandoori grill	The halal guys
American (Fast casual)	Panera bread	Sourdough and Co	Boudin café and bakery	Panera bread	Sourdough and Co	Boudin café and bakery	Panera bread	Sourdough and Co	Boudin café and bakery	Panera bread	Sourdough and Co	Boudin café and bakery
American (Fast food)	Subway	Subway	Subway	Subway	Subway	Subway	Subway	Subway	Subway	Subway	Subway	Subway
Café drinks	Starbucks	Starbucks	Starbucks	Peet's coffee	Peet's coffee	Peet's coffee	Philz's coffee	Jamba juice	Philz's coffee	Temple coffee roaster	Temple coffee roaster	Sextant coffee roaster

While the number of locations is limited to three per city, the various selection of food places in each city as well as measurement variation in time and day resulted in total of 1080 data records that is sufficient to conduct statistical analysis.

Before diving into data analysis, there were a few observations during data collection that are worth mentioning:

- In one case, Postmates increased the delivery fee more than five times when there were no couriers around to be matched with a request during an evening weekend in Davis. This observation was removed from the dataset for further analysis.
- Some restaurants might close earlier on some apps while it was still open on others. In Davis, for example, Halal Guys was open on UberEats but was closed on Doordash later in the morning.
- In some cases, GrubHub and Doordash changed the level of pricing details presented to the requesters in the receipt temporarily. While they break the taxes and fees into relevant items in most cases, they show only the total value in a few others.
- Apps differ in a few ways: GrubHub allows requesters to choose from different locations available in a region for the same restaurant. Others automatically choose one for the requesters, depending on their locations or working hours. In general, UberEats and Postmates apps crash more times when requesters frequently change a current order.
- Several restaurants in SF close operations earlier in the evening (about 5:00-6:00 PM) on all apps than in Davis and Sacramento. Moreover, some branches of the same restaurants in a city have variable working hours: some might start the day earlier, whereas others close later in the evening.

4.1.1 Menu Cost

Menu cost refers to the price indicated for the food available on the app, excluding any other fees such as taxes, service, and delivery fees. Data analysis on the collected sample from Davis showed that the menu cost of one specific food is the same across all multiple locations of chain restaurants and cafes (e.g., Subway and Starbucks) offering the same food on the apps. However, this is not the case in Sac and SF, where the same food price varies between restaurants within the same chain. For example, the menu price of a small Italian salad from Sourdough and Co. varies between $7.49 and $8.79, depending on which branch you order from in Sac. Similarly, a foot-long regular oven-roasted turkey Subway sandwich ranges in price from $9.50 to $10.39 on the menu at different Sac branches on Postmates. In SF, the same Subway sandwich menu price can be somewhere between $9.99 and $11.49 in various restaurants operating on Doordash and GrubHub. Whereas requesters can select their restaurant of interest to order from in GrubHub, the other apps choose it automatically regardless of their menu price. Analyzing the menu cost of food versus delivery travel time and distance between requesters and restaurants and time of day on weekdays or weekends did not show a particular trend or relationship. However, a larger sample might be necessary to derive more conclusions.

Although the food menu cost might differ based on the branch for which the request is made, ordering a specific food from the same restaurant or branch is the same in almost all apps in each city. *Table 4* displays more details on a subset of data collected on the food menu for those foods and restaurants available in at least two cities.

Table 4 Food menu price fluctuation for various foods, apps, and cities

Restaurant	Food	Food Menu Price ($)											
		UberEats			Doordash			Postmate			GrubHub		
		Davis	Sac	SF	Davis	Sac	SF	Davis	Sac	SF	Davis	Sac	SF
Chipotle	Burrito bowl	9.05	9.05	10.95	9.05	9.05-9.20	10.95	9.05	9.05	10.95	9.05	9.05	10.95
The Halal Guys	Chicken and beef gyro platter	12.99		14.99	12.99		14.99	12.99		14.99	13.99		14.99
Subway	Oven roasted turkey footlong regular	10.59	10.37-10.39	10.29-10.99	10.59	9.50-10.37	9.99-11.49	10.59	9.5-10.39	10.29-10.99	10.59	9.5-10.39	9.99-11.49

Starbucks	Iced matcha tea latte	4.35	4.55	5.15									
Peets Coffee	Matcha strawberry frappe				6.60	6.50	6.50-6.55						
Philz Coffee	Iced mint mojito							6.40		6.40			
Temple Coffee Roasters	Drip coffee										4.40	4.40	

4.1.2 Delivery fee

Delivery fee refers to the compensation fee paid by the requester for food transportation from the restaurant to the requester location. Either the total or a percentage of the delivery fee is paid to the couriers, depending on platform commission rate and setting. The delivery fee is independent from other fees such as *Drivers benefit* or *regulatory response fees,* which will be discussed later.

According to the additional data analysis, delivery fees might vary depending on delivery distance, time of day, and weekends or weekdays. Figure 8 presents the mean of standard deviations in delivery fees observed when delivering various food types across the day for every app and city. Although UberEats and GrubHub delivery fees are subject to change by time and day, PostMates and DoorDash have more stable and static delivery fees. Davis rarely experiences changes in delivery fees on most apps and in most food types. Café drinks and American fast-food types have more variety than the rest. Note that, except for Davis, café-drinks and American fast-food types represent more variation in locations (i.e., multiple branches for the same food type) than Mexican or Indian food types.

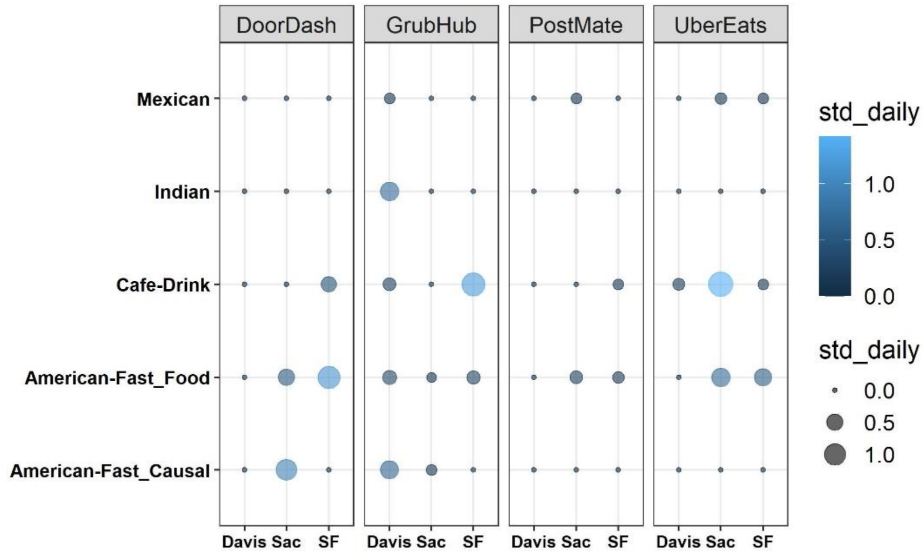


Figure 8 Average delivery fee variation per day for different food types and apps

Analyzing data revealed that from about 60 times the delivery fee was changed from the average price in the sample, 17 cases were switched to another branch for the same food and restaurant. For most of the remaining cases, changes in delivery time (AM, MD, or PM) or day of the week (weekday or weekend) were observed in addition to changes in delivery fees. The change in delivery fees was within the range of $0.5-2.00 from the usual delivery fee[1] for the same origin and destination. The change in the delivery fee and choice of branch, which eventually caused a change in the usual delivery fee for the requester, may have resulted from restaurant working hours restrictions, traffic congestion, variations in restaurant demand, and supply density. To study the reasons and make conclusions more carefully, support data on real-time traffic conditions, data regarding restaurants' temporal workloads and drivers' distribution density are necessary.

[1] Delivery fee observed most of the time for a pair of requester and restaurant locations.

4.1.3 Other fees

In addition to menu cost and delivery fees, additional items are commonly categorized as other fees in the receipt. Table 5 lists these items along with details on their implementation. Not all fees are included for all apps and cities. Regulatory response fees are paid by UberEats and Doordash users to help the platform recover revenue loss due to recent caps on fees the platform could charge restaurants per transaction. Drivers' benefit fees are included in all apps except Doordash, whose drivers get the full delivery fee, tips, and any boosts to their earnings (Helling 2021). Whereas other apps are consistent with their service fees, GrubHub differentiates its pricing with other factors such as delivery distance and destination. In SF, it uses separate service fee rates for different restaurants. Tax rates on orders are about 8-9% of menu cost with variations for some restaurants in UberEats and Doordash. Analysis of our sample showed that the tax rate might apply on a combination of menu cost and delivery fee and any small order fee in GrubHub.

Table 5 Other fees information for all apps and cities

Other fees	UberEats			Doordash			Postmate			GrubHub		
	Davis	Sac	SF	Davis	Sac	SF	Davis	Sac	SF	Davis	Sac	SF
Service fee[1] (% of menu cost)	16% (min: $3, max: $5)	16% (min: $3, max: $5)	16% (min: $3, max: $5)	17%, no more than $15 Chipotle (17%), All others (15%) while not exceeding $15	17%, no more than $15 Chipotle (17%), All others (15%) while not exceeding $15	15%, no more than $15 Chipotle, Halal, and Boudin (15%), subway, and café drinks (13%)	21%, min: $3	21%, min: $3 It can increase up to 40% to meet the $3 minimum (e.g., Jamba juice)	21%, min: $3 It can increase up to 47% to meet the $3 minimum (e.g., Philz coffee)	5-23% It varies based on unknown factors. It generally increases with delivery distance	2-23% It varies based on delivery destination	Chipotle (15%), Boudin (10%), Halal (7%), subway (10%), Sextant coffee (21%)
Taxes GrubHub: (% of menu cost + Small order fee + Delivery fee) All others: (% of menu cost)	8%	Subway (12-13%), Starbucks (0%) and All others (9%)	9%	Halal (9%), Peets coffee (0%) and All others (8%)	Peets coffee (0%) and All others (9%)	Chipotle and Halal (8%), Boudin and Subway (9%) and Peets coffee (0%)	8%	9%	9%	10-13%	10-13%	10-13%

[1] This is a convenience fee that helps the App continue to operate and maintain its platform

Small order fee ($) (condition)				3 (under $12)	3 (under $12)	3 (under $12)				2 (under $10)	2 (under $10)	2 (under $10)
Driver benefit[1] ($)	2	2	2				2.5	2.5	2	2.5, 3.5[2]	2.5, 3.5	2.5
Regulatory response fee[3] ($)			1			1						

To observe the share of each item on total price per order and compare it among various apps in different cities, this study selected two restaurants present on all apps and cities: Subway and Chipotle and the average results are presented in **Error! Reference source not found.**.

According to Figure 9(a), GrubHub is the cheapest option in all cities, whereas DoorDash has the highest total cost in Davis and SF due to its high delivery fees. Excluding the fixed fees such as driver's benefits and regulatory response fees, GrubHub has the lowest fees in almost all categories except taxes, which could not prevent the app from being the cheapest means to order a Subway sandwich online. On the other hand, ordering from Chipotle tells a different story (Figure 9 [b]). Although GrubHub is still the cheapest option in all cities, UberEats has the highest total cost thanks to its higher delivery fees in this scenario. Another observation is that the same food's menu cost is about $2 higher in SF than in two other cities. Moreover, delivery fees are lower in SF for all apps except GrubHub when ordering from Chipotle.

Assuming the item menu cost announced on the online food delivery app is the same as its price when you order in place at the restaurant, requesters must pay the additional delivery and other fees when ordering online. Depending on which city or app one chooses to order

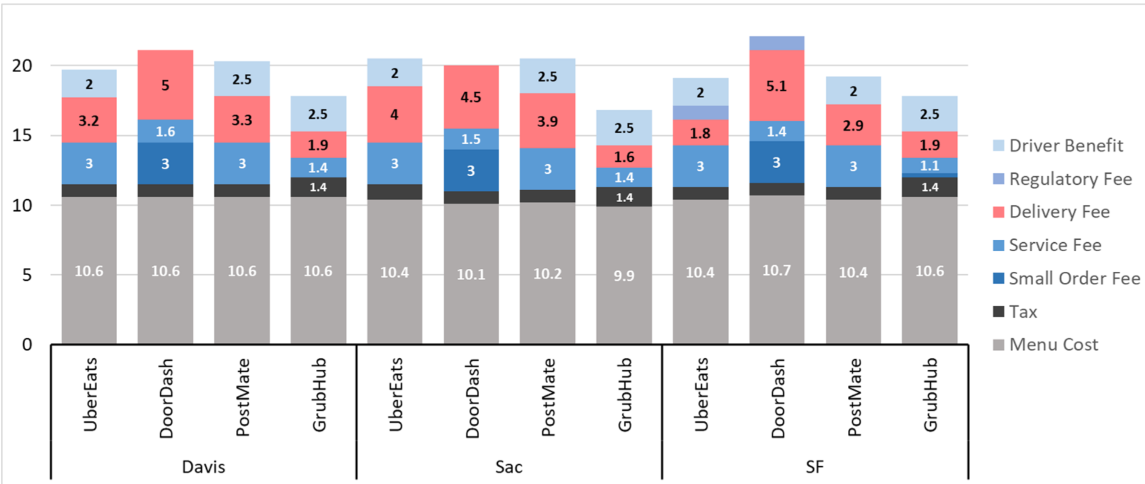
---

[1] The is intended to help cover benefits granted to drivers under Proposition 22, such as healthcare stipend, insurance and guaranteed minimum wages calculated based on local minimum wages

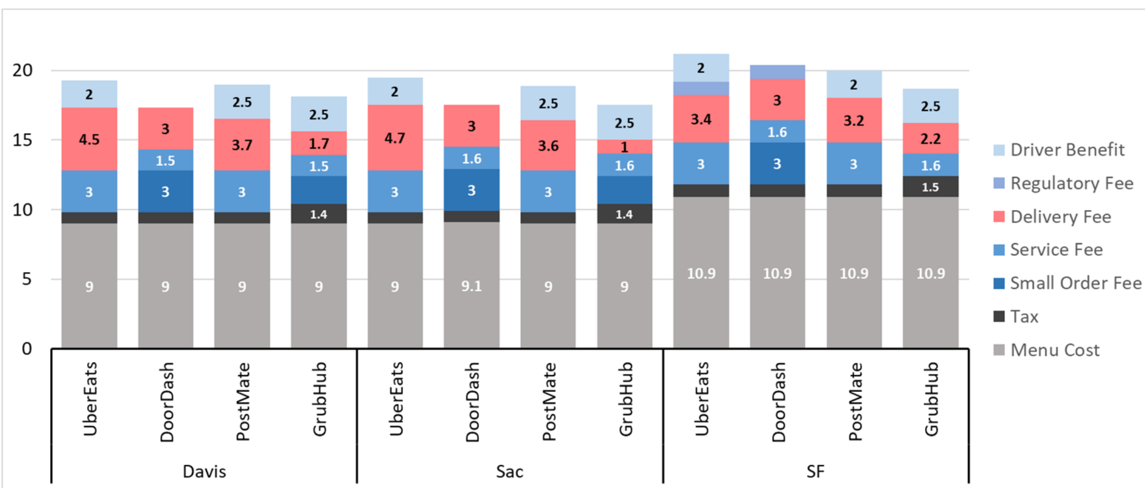
[2] In addition to cover aforementioned benefits, GrubHub drivers receive 30 cents per active mile driven.

[3] These fees go directly to the App platform to recover the lost revenue from the 15% commission cap imposed by at least 68 cities, counties, and states.

from, they, on average, might pay an extra amount of \$7-11 and \$8-10 for a \$10 subway sandwich and \$9-10 for a chipotle burrito bowl, respectively (Figure 9). These extras can equal or even exceed the menu cost at the high end.



(a)



(b)

Figure 9 Average share of fees for a typical order from (a) Subway (regular oven-roasted turkey sandwich, foot long) and (b) Chipotle (chicken burrito bowl)

4.2 Conjoint Analysis Process

The implementation of conjoint analysis involves several phases and a process of subsequent, interdependent decisions: conceptual model, the definition of attributes and levels, communication mechanism, research population and sampling, research design, survey validation, and market simulation estimation. These are further discussed in the following subsections.

4.2.1 Implementation

4.2.1.1 Conceptual model

According to this study's literature and empirical practices, delivery waiting time and fees are two critical attributes in requester satisfaction. This study aims to evaluate the impacts of these two attributes on the requesters' decisions. The study also measures the effect of time of the day on the requester's decisions given a pair of prices and waiting time. Data on sociodemographic characteristics of the requester (e.g., age and income) are also collected. The conceptual model, depicted in Figure 10, consists of independent attributes at left (i.e., delivery wait time and delivery fee) and dependent variable at right (i.e., customer decision). Control variables are inserted in the box connected to the dependent variable, which consists of demographic characteristics and delivery time of day.

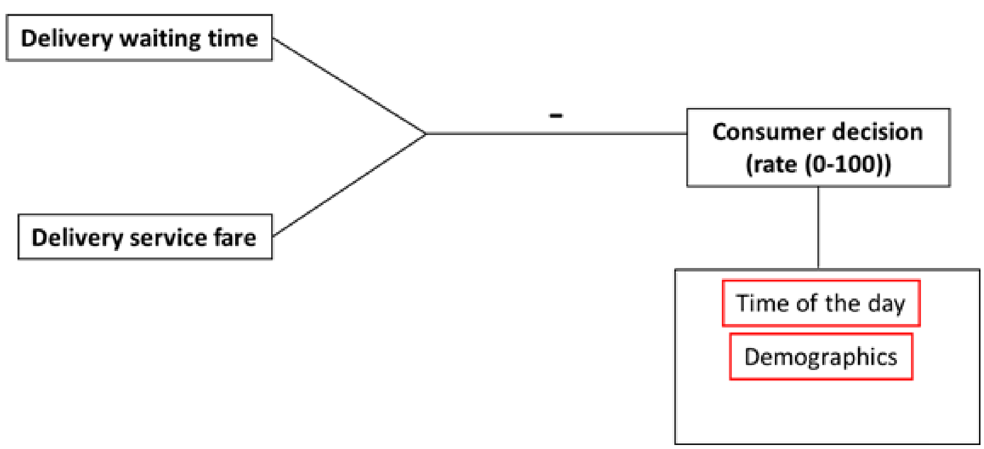


Figure 10 Conceptual model

4.2.1.2 Definition of attributes and levels

Initial attributes and levels are defined according to the previous decision. Next, a focus group was formed with eight participants (five female, mean age = 23.4 years) from students at the University of California, Davis. The focus group was intended to validate the initial list of attributes and levels and further translate or redefine terms into the fluent language of the participant. The final list of attributes and levels is presented in Table 6.

Table 6 Survey attributes and levels

Attribute	Attribute Level
Delivery time	Morning (7-10 AM); Afternoon (11 AM- 4 PM); Evening (5-9 PM)
Delivery price	Less than $5; $5-10; $10-15; More than $15
Delivery wait time	Less than 15 minutes; 15-30 minutes; 30-45 minutes; More than 45 minutes
Food delivery habit (Frequent: more than twice a week?)	Yes; No
Age	16-25; 26-35; 36-45; 46-55; 56-65; 65+
Gender	Female; Male
Education level	High school; Colleges; Bachelors; Masters; Doctorate
Employment level	Unemployed; Paid employed; Self-employed; Temporary laid-off; Retired
Annual income	Less than $10,000; $10,000-29,999; $30,000-49,999; $50,000-69,999; $70,000-89,999; $90,000-149,999; $150,000 or more

4.2.1.3 Communication mechanism

This study uses Amazon's Mechanical Turk (MTurk) (Amazon's Mechanical Turk (MTurk) 2005) to distribute an online survey to crowds of workers to complete the tasks and receive a small compensation fee in return. In large crowdsourcing marketplace, the requester submits a Human Intelligence Task (HIT) to Amazon Mechanical Turk for workers to perform. A HIT represents a single, self-contained task available to workers for a limited period, specified by the requester. A HIT is also associated with a time duration, which is the amount of time to complete a task after accepting it. The requester specifies how many workers can work on a task. Amazon Mechanical Turk guarantees that a worker can work on each task only once. Workers are paid per task by the requester only for satisfactory work.

4.2.1.4 Research population

This study considers individuals as units of analysis and adults (older than 16) living in the case study region (United States) as the target population. Amazon Mechanical Turk does not provide the option to specify this group of workers. To avoid a biased sample, it is best to use probability sampling. However, given this project's limited budget and time availability, a nonprobability sampling method is used to choose the population sample from Amazon Mechanical Turk workers.

Although MTurk can be beneficial for gathering a diverse sample in an abbreviated length of time with low cost, it suffers from a few sources of bias. MTurk workers are younger, more educated, less religious, and more likely to be unemployed than the general population. As MTurk works on some web-based platforms, it requires the availability of technology that logically might not be possible for older adults or low-income populations (Dupuis, Endicott-Popovsky et al. 2013, Difallah, Filatova et al. 2018). Amazon Mechanical Turk provides options to overcome the embedded sampling biases to some degree. One option is a clear specification about target population characteristics of interest in the HIT title and description section. Moreover, the platform has recently provided options to send out the tasks to only workers with qualifications specified by the requester (e.g., certain age or location). Another option for the requester is to add a question before administering the research questionnaire asking the worker to check some specification of interest (e.g., age) before fulfilling the task.

In our case study, food delivery requests are made through apps or web-based services. Thus, the potential limitations of age or technology dependency inside the MTurk sample might not skew the outputs. Also, the survey setup clarified the target population characteristics of interest (e.g., age and location) in the questionnaire title and description section. However, attributes such as employment status and income level might still affect individuals' willingness to pay and rating.

4.2.1.5 Research design

The survey questionnaire had several sections. It starts with a brief introduction and acknowledgment for being a U.S. resident (given the case study and consistency in monetary units) and older than 16. Then, the respondents are presented with two questions about their food delivery habits and order frequency. The third part is the most important and includes conjoint analysis rating-based tables asking individuals to rate each combination of the delivery price and wait time on a scale of 0–100 based on their preference Figure 11. This study assumed delivery price and wait time as the only important attributes in elasticity estimation. Because of the limited number of attributes, each with four levels, rating-based conjoint analysis is appropriate and enables implementing the full factorial design (displaying every level of every attribute in all sets) more efficiently. Three tables representing three times of the day (morning, afternoon, and evening) were included in the survey. Last, several questions were asked about respondent sociodemographic information. The survey was designed using the Qualtrics platform (Qualtrics 2002).

Wait Time

Price

	Less than 15 minutes	Between 15 to 30 minutes	Between 30 to 45 minutes	More than 45 minutes
Less than $5				
Between $5 to $10				
Between $10 to $15				
More than $15				

Figure 11 Conjoint rating-based table

4.2.1.6 Study validation

Validation of the data collected by survey is important to accept the results. In this study, the participant scores given to different cells inside rating tables are compared with each other to assure validity. The reliability of data collected from MTurk has not been significantly different from data collected by other means (Kim and Hodgins, 2017; Sheehan, 2017; Mortensen and Hughes, 2018). Participants who respond using MTurk generally answer reliably and consistently, as evidenced by high test-retest reliability rates (Difallah, Filatova et al. 2018).

4.2.2 Results

The study collected data following the methodology previously discussed using the MTurk online platform. A total of 131 individuals initially participated in the study. Of them, 23 were dropped from further analysis because they were incompatible with survey requirements (e.g., age and residential location) or demonstrated internal validity violation (e.g., inconsistent ratings across wait and price combinations).

The summary of the final sample of 108 individuals is illustrated in Figure 12. It shows that more than 50% of individuals are among younger age groups and females covered 59% of the total sample. The geographical distribution of participants shows that the sample mostly consists of residents living in dense urban areas where food delivery services are more popular. Most participants (78%) indicated that they frequently use online app-based food delivery services. About 74% of participants have university degrees and are employed. More than 50% of the sample have an annual income of more than $70,000. Of these, 11% have an estimated income of more than $150,000 per year.

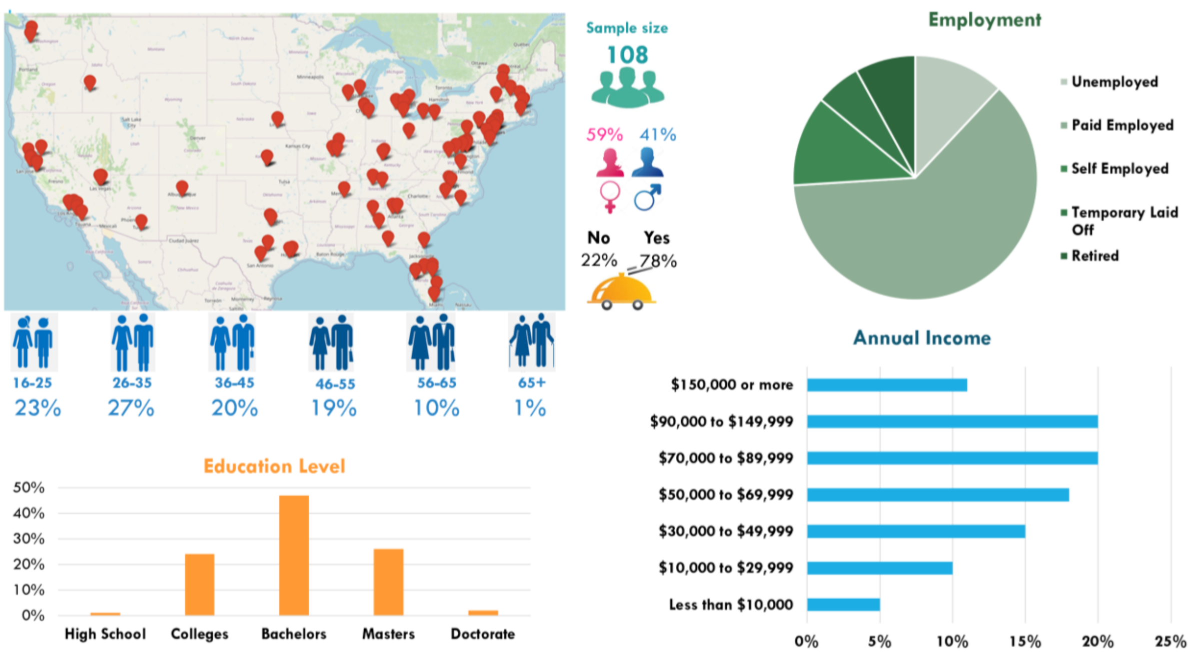


Figure 12 summary of the final sample

Furthermore, Table 7 shows the part-worth utilities (regression model parameters) for all variables and respondents. The model is fitted with F-statistics; F(9, N=5,184), equals 516.1 and p<2.2e-16. The last level of each attribute is determined as the reference category. As shown, apart from Time of Day_MD, all regression coefficients are statistically significant based on t tests. The intercept is positive and large (47.3457), indicating a positive effect on the utility when all other attributes are set to their reference category. Price <$5 and wait time <15 minutes represented the highest part-worth utilities.

Table 7 Part-worth utilities for all variables estimated, including all the respondents

Attributes	**Levels of Attribute**	**Utility Value of Estimate**	**95% Confidence Interval (CI)**	**p-Value**
Intercept	---	47.3457	[46.70, 48]	<0.001 **
Time of Day	AM	-3.1495	[-4.07, -2.23]	<0.001 **
	MD	0.2677	[-0.65, 1.19]	0.632
Waiting Time	<15 min	16.4421	[15.32, 17.57]	<0.001 **
	15-30 min	7.0687	[5.94, 8.20]	<0.001 **
	30-45 min	-5.5579	[-6.68, -4.43]	<0.001 **
Delivery Fare	<$5	29.8125	[28.69, 30.94]	<0.001 **
	$5-10	10.2562	[9.13, 11.38]	<0.001 **
	$10-15	-13.3750	[-14.50 -12.25]	<0.001 **

The relative importance of the attributes is displayed in Figure 13. This is calculated in two steps: First, the importance ratio of each attribute for each individual is calculated as the range of preference for each attribute divided by the total range of preference for all attributes. Then, the preference ratios are averaged across individuals to estimate the relative importance ratio for each attribute in sample. As shown, the price had the highest relative importance to the service's overall appraisal (54.51%), followed by wait time (33.91%). Time of day was revealed to have the lowest relative importance (11.58%).

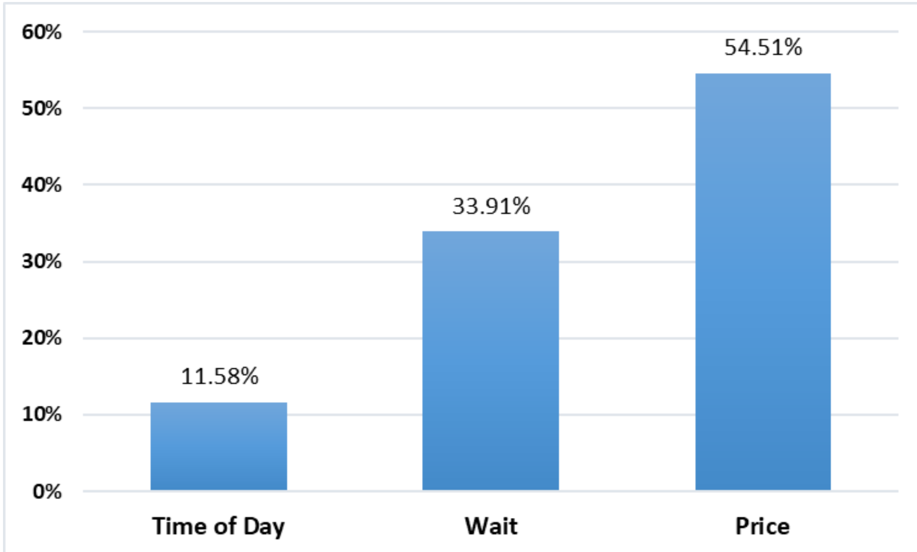


Figure 13 relative importance of attributes

To estimate the elasticity function, the study predicts the individuals' choice of delivery service using their part-worth utility estimated by conjoint analysis and selection of the service with the highest utility. Then, a binomial logit model is estimated using their choice of delivery as the outcome , and the results are displayed in Table 8. The model is fitted such that Akaike information criterion (AIC) equals 1191.2, and all the estimates are significant except the intercept, which is positive, meaning a positive impact on total utility, setting all the rest of the attributes to zero. This means that shifting time from AM to MD and PM increases the utility for food delivery choice slightly. On the other hand, delivery wait time and fare both are negative and significant, wherein delivery fare is more impactful than wait time.

Table 8 Binomial logit model summary

Attributes	Utility Value of Estimate	95% Confidence Interval (CI)	p-Value
Intercept	4.083	[3.53, 4.63]	0.407
Time of Day	0.66727	[0.49, 0.84]	<0.001 **
Waiting Time	-0.84985	[-1.03, -0.67]	<0.001 **
Delivery Fare	-1.24398	[-1.40, -1.08]	<0.001 **

4.3 Summary

In this chapter, empirical analysis results from data collected from four food delivery apps in three Californian cities are discussed. Pricing schemes comprise several items; inclusion or exclusion of each depends on the app and the city where the service is operating. Same as the total cost of food delivery varied by app and city, the food price also differed. GrubHub was the cheapest in most cases, although it lacked the variety in food types and restaurant options of others such as DoorDash. Although delivery fees showed a general increasing trend with delivery distance, the relation was unclear within each app. Delivery fees might vary by time of day, but they might be negligible or moderate depending on the app, food type, and city. This temporal change in delivery fees happens because of either change in restaurant work hours, peak in delivery demand, or traffic congestion. These finding confirm the lack of transparency in pricing discussed earlier in Chapter 1. More data is needed to make better estimates of pricing decisions for each app.

To estimate the requesters' elasticity functions for delivery, conjoint analysis and survey were conducted based on rating table designs for different combinations of the delivery fee and wait time at three times of the day: AM, MD, and PM. Among the measured variables, delivery fees were the most effective parameter in requesters elasticity, followed by wait time and day. Time of the day positively affected requesters' elasticity values (from AM to MD and PM), whereas impact for delivery fee and wait time were negative.

5. Materials and Methods

According to the findings from previous chapters, delivery pricing and wait time play critical roles in service performance and requesters' preferences. Different platforms have different pricing routines and might even operate differently in different cities. Excluding menu cost, items such as service and small order fees are almost fixed and might depend on the interrelations between platforms and restaurants. However, delivery fees vary depending on factors such as delivery travel time or distance, time of the day, or region or spatial zone of food place. Moreover, delivery fees represent one core item of the food delivery pricing scheme and include drivers 'compensations. In this chapter, a dynamic zone-based pricing model is formulated and solved as a multistage decision-making problem using DP and RL techniques discussed before. In this model, surge multipliers are determined per spatial zone and vary across sequence of time stages to adjust according to supply and demand distributions.

At the beginning of each time stage, the platform, as the central control, receives the real-time distribution of drivers and potential requesters. It then determines the optimal zonal surge multipliers for that time stage and announces it to the drivers. Depending on the platform setting, drivers then decide if they would like to participate in the delivery and enter the matching pool with requesters. Finally, the estimated wait time, based on the driver's participation decisions, and offered prices are communicated to requesters when they order through the app. Requesters evaluate the order according to their elasticity and confirm or cancel the request based on the final price and wait time. Using the actual pool of requesters and drivers, the platform performs the same zone matching and calculates the average earning per zone, which is then communicated to drivers who remained in the system without being matched at the current time step. These drivers then decide to relocate or remain in the current zone considering their service time window scheduled. This framework is illustrated in Figure 14.

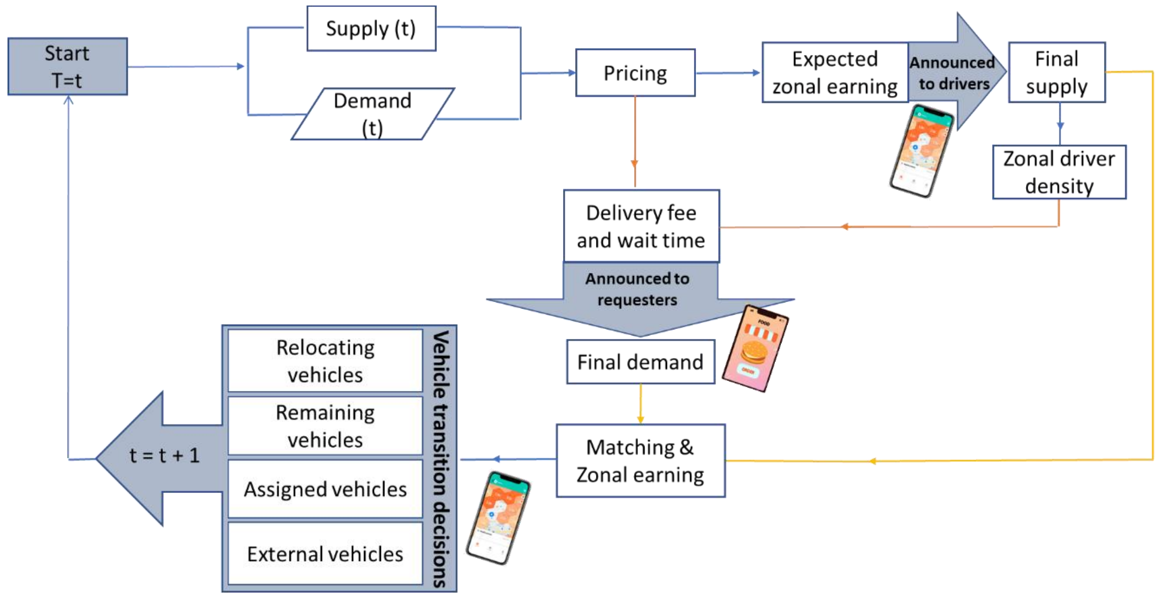


Figure 14 Pricing model implementation overview

This problem is formulated based on the DP framework. Unlike the myopic policy where the problem is solved based on deterministic information available at each time step, disregarding the potential future events (Lugtigheid, Banjevic et al. 2008), DP follows a policy where the future state of the problem depends on decisions made at different states during downstream steps (Powell, Simao et al. 2012).

5.1 Problem Definition

This section describes the mathematical formulation of the dynamic zone-based pricing model for a food delivery service. The platform would like to determine the optimal pricing decisions with respect to its objective for the entire time horizon. The network is partitioned into disjoint zones where supply and demand vary at every time stage. At the beginning of each time stage, the mass of drivers available at each zone is determined by tracking their repositioning decisions and accounting for new drivers' arrivals. The platform will declare drivers' approximate zonal earning at each time stage, and drivers decide whether to participate or not, considering their value of time. The aggregated zonal delivery origin-destination (OD) demand is given for each time interval from which the fraction of potential requests is determined with respect to the elasticity function. Here, request/trip origin is the restaurant/food place where any requester places the order, and the destination is the food delivery location from which the order is requested.

5.1.1 Platform

The platform sets prices for requests and decides the matching between requests and drivers for the food delivery. The incremental pricing rule is adopted, comprising a fixed rate per unit of delivery distance. Depending on the supply and demand relation, dynamic surge multipliers are determined to adjust the price at each zone in the system. Same-area matching is implemented where drivers are matched with requests within the same zone. The platform charges a fixed percentage of the fare as its commission rate. Neither the driver nor the requester can reject the trip after they have been matched by the platform and requesters that are not assigned to any driver leave the platform.

The platform has two objectives: 1) maximizing platform profit by maximizing the total earning from deliveries and 2) maximizing requesters social welfare by maximizing the total deliveries. In the latter case, author makes two additional assumptions in the model to improve the system equity for both requesters and drivers: 1) the domain of surge multipliers is expanded to also include values less than one to lower the original base fare; and 2) the priority is given to drivers with the lowest number of completed trips during the assignment process to increase the working drivers in the system.

5.1.2 Requesters

The network is partitioned into $\boldsymbol{N}$ disjoint zones. Each request is represented by <***i, j, t***>, in which $\boldsymbol{i}$ and $\boldsymbol{j}$ are origin and destination zones, respectively, and $\boldsymbol{t}$ is the request announcement time. This study assumes the total potential demand per zone is given for each $\boldsymbol{t}$. Aggregated requests in zone $\boldsymbol{i}$ at time $\boldsymbol{t}$, $\boldsymbol{D_i^t}$, is estimated as a fraction of total demand $\boldsymbol{d_i^{t,k}}$, given the price $\boldsymbol{p_i^{t,k}}$ and waiting time $\boldsymbol{w_i^{t,k}}$ offered by the platform, proportional to the elasticity function $\boldsymbol{\varphi(p_i^{t,k}, w_i^{t,k})}$. This is presented in Eq. 5.1. Here, $\boldsymbol{K_i^t}$ refers to total demand announced in zone $\boldsymbol{i}$, at time $\boldsymbol{t}$.

$$\boldsymbol{D_i^t = \sum_{k=1}^{K_i^t} \varphi(p_i^{t,k}, w_i^{t,k}) d_i^{t,k}} \qquad \forall \boldsymbol{i} \in \boldsymbol{N}, \forall \boldsymbol{t} \in \boldsymbol{T} \qquad (\mathbf{5.1})$$

$$\boldsymbol{p_i^{t,k} = x_i^t \beta dist^k} \qquad \forall \boldsymbol{i} \in \boldsymbol{N}, \forall \boldsymbol{t} \in \boldsymbol{T}, \boldsymbol{k} \in \boldsymbol{K_i^t} \qquad (\mathbf{5.2})$$

$$\boldsymbol{w_i^{t,k} = \eta_i^t + ttim^k + \psi} \qquad \forall \boldsymbol{i} \in \boldsymbol{N}, \forall \boldsymbol{t} \in \boldsymbol{T} \qquad (\mathbf{5.3})$$

$$\boldsymbol{\eta_i^t = 3.01 f_i^t - 0.31} \qquad \forall \boldsymbol{i} \in \boldsymbol{N}, \forall \boldsymbol{t} \in \boldsymbol{T} \qquad (\mathbf{5.4})$$

In Eq. 5.2, $\boldsymbol{\beta}$ equals the fixed delivery rate per unit of distance and $\boldsymbol{x_i^t}$ is the surge multiplier in zone $\boldsymbol{i}$ at time $\boldsymbol{t}$. Also, $\boldsymbol{dist^k}$ refers to the delivery distance to fulfill request $\boldsymbol{k}$; In Eq. 5.3, waiting time equals the summation of driver's en route time $\boldsymbol{\eta_i^t}$ (time for the driver to arrive at order pickup point located in zone $\boldsymbol{i}$, at time $\boldsymbol{t}$, after being matched) $\boldsymbol{ttim^k}$ (delivery distance to fulfill request $\boldsymbol{k}$) and matching time window $\boldsymbol{\psi}$. Using UberX data, (Castillo, Knoepfle et al. 2017) estimated drivers' en route time as a function of drivers' density in zone

$\boldsymbol{i}$ at time $\boldsymbol{t}$, $\boldsymbol{f_i^t}$, which is presented in Eq. 5.4. This study adopts this function for drivers' zonal en route time because estimating this requires available data on delivery drivers' itinerary and is out of scope.

5.1.3 Drivers

Each driver is defined by <$\boldsymbol{i, t_1, t_2}$>, where $\boldsymbol{i}$ is the zone where they are available and $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ correspond to driver arrival and departure time from the system, respectively. Three states are defined for drivers: 1) open to be matched, 2) assigned (en route to pick up or deliver the order) 3) relocating. At the beginning of each time interval, drivers' status is updated, and only open drivers are considered for matching decisions. The platform declares the surge multiplier per zone, then drivers decide whether to participate or not considering the expected earning and their value of time (Eq. 5.5).

$$\boldsymbol{p_i^t \overline{dist}_i^t \mu_i^t - c_r \overline{ttim^t}_i} > 0 \qquad \forall \boldsymbol{i} \in \boldsymbol{N}, \forall \boldsymbol{t} \in \boldsymbol{T}, \boldsymbol{r} \in \boldsymbol{R} \qquad (\mathbf{5.5})$$

$$\boldsymbol{\mu_i^t} = \mathbf{min}\left(\frac{\boldsymbol{d_i^t}}{\boldsymbol{s_i^t}}, \mathbf{1}\right) \qquad \forall \boldsymbol{i} \in \boldsymbol{N}, \forall \boldsymbol{t} \in \boldsymbol{T} \qquad (\mathbf{5.6})$$

In Eq. 5.5, $\boldsymbol{c_r}$ represents driver's value of time, and $\boldsymbol{\overline{ttim^t}_i}$ is the average delivery time from zone $\boldsymbol{i}$, at time $\boldsymbol{t}$. $\boldsymbol{\mu_i^t}$ denotes the probability that a driver participates in zone $\boldsymbol{i}$ at time $\boldsymbol{t}$. If demand $\boldsymbol{d_i^t}$ exceeds the supply $\boldsymbol{s_i^t}$, the demand is randomly rationed, and all drivers participate as long as $\boldsymbol{p_i^t \overline{dist}_i^t \mu_i^t}$ exceeds the second term. Otherwise, $\boldsymbol{\mu_i^t}$ fraction of drivers participates. Here, $\boldsymbol{p_i^t}$ is the offered price per mile of delivery distance from zone $\boldsymbol{i}$ at time $\boldsymbol{t}$. Drivers' VOT that is randomly distributed between \$6 and \$12 per hour, consistent with the range of mean VOT for different population groups in the Bay Area Travel model.

The drivers that are not matched with any request decide to either remain idle in their current zone or relocate to another, constrained to their session end time $\boldsymbol{t_2}$. This decision is made at each time $\boldsymbol{t}$ by comparing average earned profit at the previous time step in the current zone $\boldsymbol{i}$, $\overline{\boldsymbol{P}}_{\boldsymbol{i}}^{\boldsymbol{t-1}}$, with the average profit gained in other zones considering driving cost, $\boldsymbol{\delta}$ (Eq.5.7):

$$\boldsymbol{j^* = arg \max_{j \in N}\{(\overline{P}_j^{t-1} - \delta_{ij}) - \overline{P}_i^{t-1}\}} \qquad (\mathbf{5.7})$$

$$\boldsymbol{s.t. \quad t + ttim_{ij} < t_2}$$

After the initial time interval, $\boldsymbol{w}$ external drivers enter the system choosing the zone with the highest average profit earned by drivers at the previous time interval (Eq. 5.8). It is assumed that $\boldsymbol{w}$ follows a Poisson probability distribution with the parameter proportional to the number of matched requests $\boldsymbol{n}$ in the target zone at the previous time interval (Eq. 5.9).

$$\boldsymbol{j^* = arg \max_{j \in N}\{(\overline{P}_j^{t-1}\}} \qquad (\mathbf{5.8})$$

$$\boldsymbol{w \approx poisson\left(n.\overline{P}_{j^*}^{t-1}\right)} \qquad (\mathbf{5.9})$$

The platform's objective to maximize the profit is indicated by Eq. 5.10.a, where $\mathbf{\Pi}$ is the total earning from deliveries by matching and pricing decisions considering all zones and time intervals (Eq. 5.11.a).

$$\max_{\boldsymbol{x}} \mathbf{\Pi}(\boldsymbol{x}) \qquad (\mathbf{5.10.}\boldsymbol{a})$$

$$\mathbf{\Pi}(\boldsymbol{x}) = \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \mathbf{min}\,(\boldsymbol{D_i^t, O_i^t})\overline{\boldsymbol{dist}}_{\boldsymbol{i}}^{\boldsymbol{t}}\, \boldsymbol{x_i^t} \qquad (\mathbf{5.11.}\boldsymbol{a})$$

The platform's objective to maximize social welfare is indicated by (Eq. 5.10.b), where $\mathbf{\Pi}$ is the total deliveries in all zones and time intervals (Eq. 5.11.b).

$$\max_{\boldsymbol{x}} \mathbf{\Pi}(\boldsymbol{x}) \qquad (\mathbf{5.10.}\boldsymbol{b})$$

$$\mathbf{\Pi}(\boldsymbol{x}) = \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \mathbf{min}\,(\boldsymbol{D_i^t, O_i^t}) \qquad (\mathbf{5.11.}\boldsymbol{b})$$

In Eq. 5.11.a and 5.11.b, $\boldsymbol{O_i^t}$ presents the total available supply of drivers at time interval $\boldsymbol{t}$ which comprises of remaining drivers, relocated drivers, and external drivers who agreed to participate in zone $\boldsymbol{i}$ according to Eq. 5.5 and Eq. 5.6. Reminder that both $\boldsymbol{D_i^t}$ and $\boldsymbol{O_i^t}$ are in terms of $\boldsymbol{x_i^t}$, according to Eq. (5.1-5.2, 5.5-5.9). The term $\mathbf{min}\ (\boldsymbol{D_i^t}, \boldsymbol{O_i^t})$ defines the number of deliveries in zone $\boldsymbol{i}$ at time interval $\boldsymbol{t}$. Accordingly, when there are more requests than drivers, requests are randomly rationed. Conversely, the drivers are randomly rationed when there are more drivers than requests.[1] Average delivery distance in zone $\boldsymbol{i}$ at time interval $\boldsymbol{t}$ is represented by $\overline{\boldsymbol{dist}}_{\boldsymbol{i}}^{\boldsymbol{t}}$.

5.2 Dynamic Programming

DP solves for optimal policy in a dynamic environment, considering the system's current state, which is essential for decision making. DP executes a decision and observes a reward once it acquires the full state information. Then, the system transitions to a new state. The outcome of DP is the optimal policy that determines what decision to make for each system stage. A DP is formulated to solve the pricing problem defined in Section 5.1. Each time step $\boldsymbol{t}$ represents a decision stage. Surge multipliers per zone $\boldsymbol{x^t}$ are decision variables that are continuously distributed within an interval of $[\boldsymbol{x_{min}}, \boldsymbol{x_{max}}]$. Each state is characterized by the zonal distribution of demand and supply as well as average delivery distance, $\boldsymbol{S_t}\left(\boldsymbol{D^t}, \boldsymbol{O^t}, \overline{\boldsymbol{dist}}^{\boldsymbol{t}}\right)$. The transition rule between consecutive states is defined by new demand arrivals and drivers' decisions. While the former is independent of the previous stage, the latter is directly defined by the state and decisions in the preceding stages. Pricing decisions at each stage affect the matched drivers and earned zonal profits, which in return affects the

[1] In social welfare scenario, priority is with drivers having lower deliveries until time interval $\boldsymbol{t}$

drivers' repositioning decisions and external arrivals for the next stage (Eq.5.5‑5.9). Thus, these must be considered in the transition rule (Eq. 5.12).

$$\boldsymbol{S_{t+1}} = \boldsymbol{S^T}(\boldsymbol{S_t}, \boldsymbol{x^t}; \boldsymbol{D^{t+1}}) \qquad (\mathbf{5.12})$$

Given the $\boldsymbol{S_t}$ and $\boldsymbol{x^t}$, the reward function $\boldsymbol{\chi}(\boldsymbol{S_t}, \boldsymbol{x^t})$ is Eq 5.10.a (maximizing total earning) or Eq. 5.10.b (maximizing social welfare), which performs matching decisions between the final requesters and drivers (Eq. 5.1‑5.6) in all the zones and outputs the resulting profit or deliveries (if it is the social welfare scenario).

Ultimately, Bellman's principle of optimality recursion rule of the above-described DP is represented by Eq. 5.13, which demonstrates that we are interested in finding optimal zonal surge multipliers to maximize the objective for all stages, or the time horizon.

$$\boldsymbol{\Pi}(\boldsymbol{t}, \boldsymbol{S_t}) = \max_{\boldsymbol{x^t}}\{\boldsymbol{\chi}(\boldsymbol{S_t}, \boldsymbol{x^t}) + \boldsymbol{\Pi}(\boldsymbol{t+1}, \boldsymbol{S_{t+1}})\} \qquad (\mathbf{5.13})$$

The solution for this problem comprises optimal decisions at each decision stage. This is usually determined by classic DP recursive computation (e.g., generalized policy iteration algorithm explained in Section 3.4.2). However, the complexity and the scale of the described problem present a challenge here. Considering large dimensional state space and continuous decision variable domain, the cardinality of this problem is not finite. If A and S are decision and state spaces, respectively, the computational complexity per iteration for policy iteration is $\boldsymbol{O}(|\boldsymbol{A}||\boldsymbol{S}|^2 + |\boldsymbol{S}|^3)$ and for value iteration is $\boldsymbol{O}(|\boldsymbol{A}||\boldsymbol{S}|^2)$ (Kaelbling, Littman et al. 1996). Thus, the problem is intractable to be solved by these algorithms. Even if we assume a bound over the state space and discretize the decision variable (e.g., N = 100, $\boldsymbol{D^t_{max}}$ = 20, $\boldsymbol{S^t_{max}}$ = 10, $\boldsymbol{x^t} \in \{\mathbf{0.5}, \mathbf{1}, \mathbf{2}\}$: the state has 20,000 dimensions and the decision space has $\mathbf{3^{100}}$ dimensions at each decision stage $\boldsymbol{t}$), it still suffers from the curse of dimensionality and becomes computationally expensive to solve.

5.3 Reinforcement Learning

To solve this problem, a macro simulation-optimization solution framework is introduced in this section. For the simulation, a RL agent plays in a built macro simulation environment by following the defined DP rules and simulate a given pricing policy including state transition and reward accumulation sequence for the entire time horizon, $\boldsymbol{T}$, satisfying constraints and assumptions defined in Eq. 5.1-5.11. During each simulation run, the states, decisions and rewards are recorded for every decision stage as trajectories and are transferred to the RL function approximator. Here, they are used as training sets to estimate a linearly weighted function $\boldsymbol{V}$, in terms of decision states and decision variables to approximate the total profit or deliveries for all time steps $\boldsymbol{T}$, following the sequence in the training set (Figure 15). More details about the value function approximator is provided later in section 5.3.1.

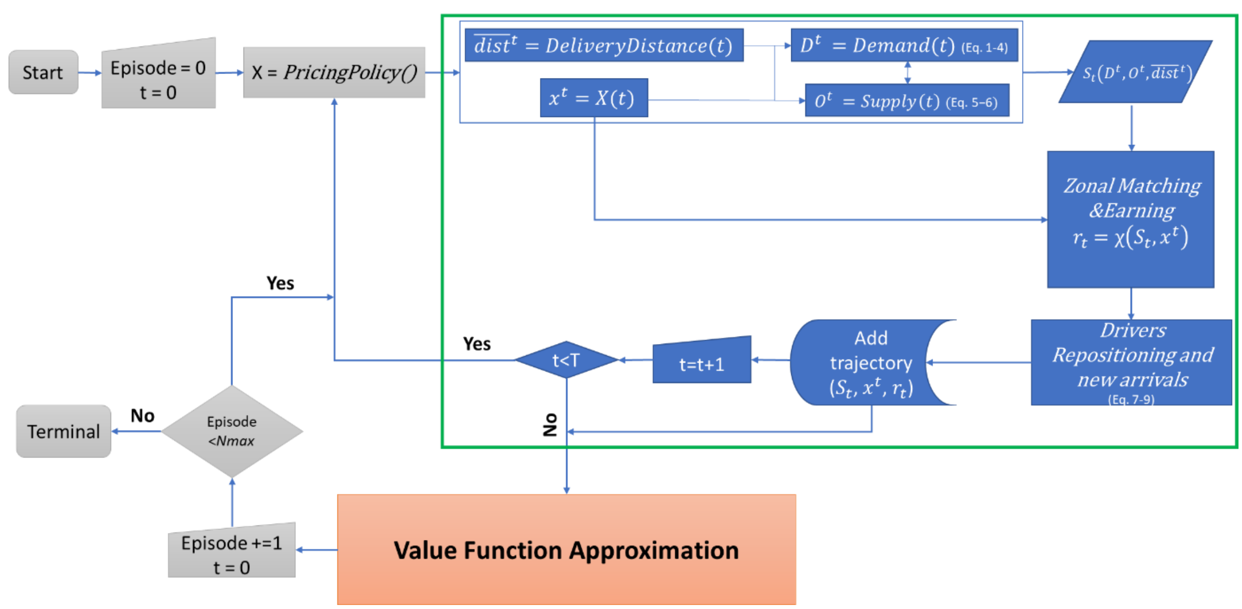


Figure 15 Solution approach step 1

After the value function approximator is trained through enough episodes, it is employed to estimate the optimal pricing policy and value for various distributions of demand and supply in the case study. For each time step, the zonal distribution of demand, supply, and delivery distances are given to a multi-start point optimization algorithm to optimize the value function approximator and find the optimal set of zonal multipliers with respect to the given objective (maximizing total profit or deliveries) (Figure 16).

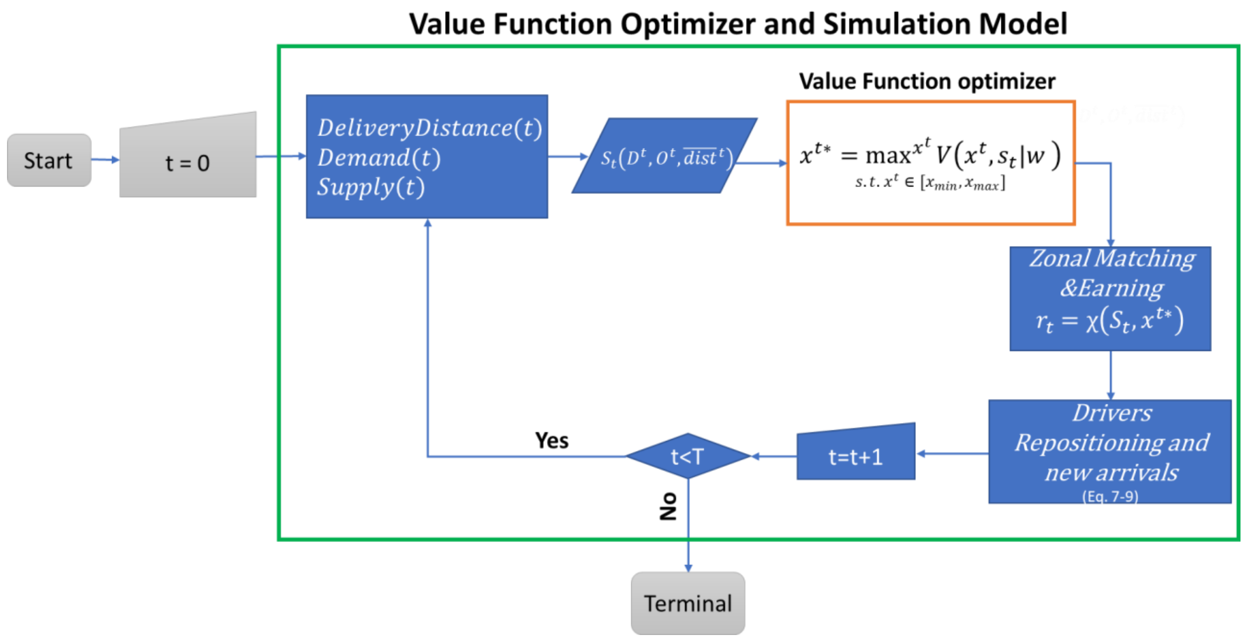


Figure 16 Solution approach step 2

5.3.1 Value Function Approximation and Algorithm

This study performs a function approximation technique to train a parametrized linear function in state features by generating simulated experiences. The objective is that the function takes the state features as inputs at each decision step and returns the optimal decision as output. The process applies RL techniques to train a linear function approximator effective for large continuous state-decision spaces when training data is unavailable.

Learning is done through the real-time agents simulated interaction in the environment to update the function parameters and generalize the learnings to future practices.

Contrary to classic DP methods, the process explicitly specifies an objective function, $\boldsymbol{V}$, to train episodically and minimize the prediction error by finding an optimal weight vector $\boldsymbol{w}^*$ for which $\boldsymbol{V}(\boldsymbol{w}^*) \leq \boldsymbol{V}(\boldsymbol{w})$ for all possible $\boldsymbol{w}$. The approximate function depends on both states, and actions/decisions where it is represented by $\boldsymbol{V}(\boldsymbol{a}, \boldsymbol{s}|\boldsymbol{w})$, including a set of $\boldsymbol{n}$ features defined to represent the state and action decisions (Eq. 5.14).

$$\boldsymbol{V}(\boldsymbol{a}, \boldsymbol{s}|\boldsymbol{w}) = \boldsymbol{w_1 f_1}(\boldsymbol{a}, \boldsymbol{s}) + \boldsymbol{w_2 f_2}(\boldsymbol{a}, \boldsymbol{s}) + \boldsymbol{w_3 f_3}(\boldsymbol{a}, \boldsymbol{s}) + \ldots + \boldsymbol{w_n f_n}(\boldsymbol{a}, \boldsymbol{s}) \qquad (\mathbf{5.14})$$

This problem requires defining an $\boldsymbol{N}$ dimension vector to represent the $\boldsymbol{N}$ zonal state $\boldsymbol{S_t}\left(\boldsymbol{D^t}, \boldsymbol{O^t}, \overline{\boldsymbol{dist}}^{\boldsymbol{t}}\right)$ and decision at each decision step $\boldsymbol{t}$. Accordingly, the feature vector includes one component per zone $\boldsymbol{f_i}(\boldsymbol{a}, \boldsymbol{s})$ $\forall \boldsymbol{i} \in \boldsymbol{N}$ that depending on the problem objective measures the expected profit (for maximizing total profit) or number of deliveries (for maximizing social welfare) with respect to state $\boldsymbol{s_t}$ and decision variable in that specific zone $\boldsymbol{a_t}$ (Eq. 5.15).

$$\boldsymbol{V}(\boldsymbol{a_t}, \boldsymbol{s_t}|\boldsymbol{w}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{w_i f_i}(\boldsymbol{a_t}, \boldsymbol{s_t}) \qquad \forall \boldsymbol{t} \in \boldsymbol{T} \qquad (\mathbf{5.15})$$

For maximizing total profits, this study assumes that the expected profit (comprising the immediate profit at time t and discounted future earnings till the end of the time horizon) at each zone $\boldsymbol{i}$ is proportional to the average delivery distance for matched requesters and drivers, $\mathbf{min}\,(\boldsymbol{D_i^t}, \boldsymbol{O_i^t})\overline{\boldsymbol{dist}}_{\boldsymbol{i}}^{\boldsymbol{t}}$, multiplied by its corresponding surge pricing decision, $\boldsymbol{a_t}$. Hence, our approximated value function is presented as in Eq. 5.16.a:

$$\boldsymbol{f_i}(\boldsymbol{a_t}, \boldsymbol{s_t}) = \mathbf{min}(\boldsymbol{D_i^t}, \boldsymbol{O_i^t})\, \overline{\boldsymbol{dist}}_{\boldsymbol{i}}^{\boldsymbol{t}} \boldsymbol{x^t} \qquad \forall \boldsymbol{i} \in \boldsymbol{N}, \ \forall \boldsymbol{t} \in \boldsymbol{T} \qquad (\mathbf{5.16.a})$$

For the social welfare objective, this study assumes that the expected deliveries (comprised from the immediate number of matches at time $\boldsymbol{t}$ and future matches until the end-of-time horizon) at each zone $\boldsymbol{i}$ is proportional to the matched trips at the current state, $\mathbf{min}\ (\boldsymbol{D}_i^t, \boldsymbol{O}_i^t)$**.** Hence, our approximate value function is presented as in Eq. 5.16.b:

$$\boldsymbol{f}_i(\boldsymbol{a}_t, \boldsymbol{s}_t) = \mathbf{min}(\boldsymbol{D}_i^t, \boldsymbol{O}_i^t) \qquad \forall \boldsymbol{i} \in \boldsymbol{N},\ \ \forall \boldsymbol{t} \in \boldsymbol{T} \qquad (\mathbf{5.16.}\boldsymbol{b})$$

The $\boldsymbol{V}$ function is constructed to estimate the value of being at state $\boldsymbol{s}$, taking decision $\boldsymbol{a}$. The weights must be determined in a way to minimize the difference between the estimated value by function approximator, $\widehat{\boldsymbol{V}}$**,** and $\boldsymbol{U}$ which can be the actual return of the state and decision, an estimated average, an erroneous version of $\boldsymbol{V}$ or one of its equivalent TD versions. Thus, the value function approximation mean squared error is denoted by Eq. 5.17:

$$\boldsymbol{Jw} = \overline{\sum_{\boldsymbol{a} \epsilon \boldsymbol{A}, \boldsymbol{s} \in \boldsymbol{S}} (\boldsymbol{U} - \ \widehat{\boldsymbol{V}}\,(\boldsymbol{a}, \boldsymbol{s}|\boldsymbol{w}\,))^{\mathbf{2}}} \qquad (\mathbf{5.17})$$

The stochastic gradient descent (SGD) algorithm is well suited for solving real-time optimization problems. In gradient-descent methods, the weight comprises a fixed number of real-valued components $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$, and the approximate value function $\boldsymbol{V}$ is a differentiable function concerning all weights. Each episode has a single series of observations from successive states and their corresponding decisions and rewards, occurring at discrete time steps, $< \mathbf{s}_1, \mathbf{a}_1, \mathbf{r}_1, \mathbf{s}_2, \mathbf{a}_2, \mathbf{r}_2, \ldots, \mathbf{s}_T, \mathbf{a}_T, \mathbf{r}_T >$. This trajectory is sampled from agents' interaction with the environment. At the end of each episode, SGD modifies the weights by adjusting the weight vector to reduce the error for that observation (Eq. 5.18 and 5.19).

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{\alpha} \boldsymbol{\nabla} \big[\boldsymbol{U}_t - \ \widehat{\boldsymbol{V}}(\boldsymbol{a}_t, \boldsymbol{s}_t|\boldsymbol{w}_t)\big]^2 \qquad (\mathbf{5.18})$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \boldsymbol{\alpha}[\boldsymbol{U}_t - \ \widehat{\boldsymbol{V}}(\boldsymbol{a}_t, \boldsymbol{s}_t|\boldsymbol{w}_t)]\ \boldsymbol{\nabla}\widehat{\boldsymbol{V}}(\boldsymbol{a}_t, \boldsymbol{s}_t|\boldsymbol{w}_t) \qquad (\mathbf{5.19})$$

where $\boldsymbol{\alpha}$ is the learning rate and $\boldsymbol{\nabla\widehat{V}}$ is the partial derivative with respect to the weight vector component, which is the direction that error falls most rapidly (Eq. 5.20).

$$\boldsymbol{\nabla\widehat{V}}(\boldsymbol{a_t},\boldsymbol{s_t}|\boldsymbol{w_t}) = [\boldsymbol{f_1}(\boldsymbol{a_t},\boldsymbol{s_t}),\boldsymbol{f_2}(\boldsymbol{a_t},\boldsymbol{s_t}),\dots,\boldsymbol{f_{n-1}}(\boldsymbol{a_t},\boldsymbol{s_t}),\boldsymbol{f_n}(\boldsymbol{a_t},\boldsymbol{s_t})] \qquad (\mathbf{5.20})$$

In each episode, the SGD makes little progress in reducing the overall error. According to (Sutton and Barto 2018), there is no way to reduce error for all the states and actions by examining a sample of trajectories. Instead, there is a need to approximate a function to balance the error and generalize the training to other trajectories not observed in the samples.

In this study, the transition rule between states cause the Markovian property to not hold because the pool of available drivers at each state depend on a series of decisions made during previous stages rather than only the current stage. According to this and the finite episodic feature of the pricing problem in this study, MC actual return is considered as the target value $\boldsymbol{U_t}$. At each stage, the true value of taking decision $\boldsymbol{a}$ at state $\boldsymbol{s}$ is the expected value of the outcome following it. In other words, it equals the summation of the immediate reward earned at that specific decision stage and the expected discounted value gained in later stages following the fixed policy until the end-of- time horizon (Eq. 5.21).

$$\boldsymbol{U_t} = \boldsymbol{r_t} + \boldsymbol{E}\left(\sum_{\boldsymbol{t+1}}^{\boldsymbol{T}} \boldsymbol{\gamma^i U_i}\right) \qquad (\mathbf{5.21})$$

In Eq. 5.21, $\boldsymbol{r_t}$ is the immediate reward gained at stage $\boldsymbol{t}$ and $\boldsymbol{\gamma}$ is the discount factor imposed on the value gained at every other stage afterwards. Accordingly, MC target $\boldsymbol{U_t =: G_t}$ is by definition an unbiased estimate of the true value and good alternative for $\boldsymbol{U_t}$. With this choice, the MC version of SGD is guaranteed to converge to a locally optimal approximation of ***V(a_t,***

s_t) (Sutton and Barto 2018). The MC version of the SGD algorithm implemented in this study is presented in Figure 17.

Initialize value-function weights $\mathbf{w_0}$ as appropriate (e.g., $\mathbf{w_0} \leftarrow \mathbf{0}$)
Repeat for each episode:
G ← 0
Generate a trajectory $s_0, a_0, r_0; s_1, a_1, r_1; :::; s_T, a_T, r_T;$ following a pricing policy π
For t = T-1, T-2, … ,0:
    G ← υG + r_t
    $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha[G - \hat{V}(a_t, s_t|\boldsymbol{w}_t)]\, \nabla\hat{V}(a_t, s_t|\boldsymbol{w}_t)$

Figure 17 MC-based SGD pseudocode (Sutton and Barto 2018)

SGD optimization requires that the training data be independent and identically distributed (I.I.D)(Lapan 2018). Here, the state transitions within training trajectories collected by interacting with the environment belong to the same episode. Maintaining memory replay is one potential solution in the literature to train on more-or-less independent data. Memory replay refers to a large queue of experienced trajectories in full episodes to sample training data out of it rather than using the most recent trajectory. The queue structure of memory replay enables the recent data added to its end as it pushes the oldest one out of it. Memory replay strategy enables using previous experiences more efficiently by learning from them multiple times in later episodes.

In this study, the learning process is on-policy and requires training data to be sampled according to the currently updated policy. However, memory replay samples from old data that is not identical to the current distribution. While reducing the size of memory replay might be a possible solution for simple environments, this study implements the parallel training environment idea. In this strategy, the algorithm communicates with $\boldsymbol{m}$ independent simulation environments starting from $\boldsymbol{m}$ different random pricing policies which are improving concurrently. At each iteration, one trajectory is randomly sampled from the $\boldsymbol{m}$

environments for training the function. It is worth mentioning that randomly sampling from $\boldsymbol{m}$ independent environments also decreases the high variance from MC-based algorithms.

The study implemented the MC-based SGD algorithm combined with multiple concurrent environments (Figure 18). In this strategy, each environment initiates by a random pricing policy.

Initialize value-function weights $\mathbf{w_0}$ as appropriate (e.g., $\mathbf{w_0} \leftarrow \mathbf{0}$)
For each *m* environment**:**
    Generate a random pricing policy π_m
    Simulate a trajectory <$s_0, a_0, r_0; s_1, a_1, r_1; \; : : : \; ; s_T, a_T, r_T$> following the given pricing policy π_m
    Add the trajectory and the discounted total reward R_m to batch *B*
Repeat for each episode:
  Randomly select an environment from ***B*** and unpack its trajectory <$s_0, a_0, r_0; \; : : : \; ; s_T, a_T, r_T$>
  $G \leftarrow 0$
  For t = *T-1, T-2, … ,0*:
    $G \leftarrow \upsilon G + r_t$
    $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha[G - \hat{V}(a_t, s_t|\boldsymbol{w}_t)] \, \nabla\hat{V}(a_t, s_t|\boldsymbol{w}_t)$
  For each *m* environment**:**
    π'_m, $R_m' \leftarrow$ PolicyImprovement(π_m, R_m)
    Add the trajectory and the discounted total reward R'_m to batch *B*
  Go to **6**

Figure 18 MC-based SGD algorithm pseudocode

Randomly interacting with the environment is not guaranteed to find optimal policies. Random selection is suitable at the beginning of the training when the agent needs to explore more. Still, exploration must be efficient to avoid exploring the policies already visited or states having low chance to occur. One alternative to random policy selection is to use the function approximator being trained to guide the pricing policy selection. However, the function approximator might not perfectly represent the environment, particularly at the beginning of the training.

This study designs an evolutionary strategy (ES) policy improvement algorithm to guide the policy selection and improvement tasks in concurrent environments (Figure 19). This algorithm comprises of two parts. First it starts from a given random pricing policy (Figure 18, line 3) for each environment; the policy is executed in the simulation environment (Figure 18, line 4); then, the current policy $\boldsymbol{\pi}$ and its expected return, $\boldsymbol{R}$, are stored in the batch (Figure 18, line 5). In subsequent episodes, the pricing policies evolution for each environment using the ES policy improvement function (Figure 18, line 13). The function attempts to locally modify the current policy randomly in the direction of the best policy found so far (Figure 19, line 5-6). The newer policy is compared with current policy (Figure 19, line 7-8): if it improves the expected return, it both replaces the current and the best policy (Figure 19, line 9-11); otherwise, it will be selected based on stochastic Boltzmann probability function (Figure 19, line 13‐17). This algorithm gradually improves the selected pricing policy for all concurrent environments independently; simultaneously, it enables the algorithm to throw away bad policies and train on better ones. Given its stochasticity, it maintains the balance between exploration and exploitation, which works based on the Boltzmann temperature parameter, starting from high values and decreasing gradually. This leads to more exploration at the initial training and more exploitation at the end. The policy and trajectories generated at each episode replace those in the batch, then a single/mini batch trajectory is randomly sampled to update the weights.

```
1   Initialize Boltzmann Constant, T, direction, W(ω_1, ω_2), Hrate, Drate
2   Initialize old policy and best policy, to empty m-size lists
3   Initialize old value, and best value to m zero variables
4   PolicyImprovement(π, R) {
5       π′ ← π + direction
6       Simulate a trajectory <s_0, a_0, r_0; s_1, a_1, r_1; : : : ; s_T, a_T, r_T> following the given pricing policy π’
7       Δ ← R′ − R
8       If Δ>0:
9           [policy,value] ← π’, R’
10           If R’ > best value:
11               [best policy, best value] ← π’, R’
```

```
    Else
        Probability ← Exp(Δ/Boltzman Constant*T)
        If random(0,1) < Probability :
            [policy, value] ← π', R'
        Else
            [policy,value] ← π, R
    direction ← ω1.direction + ω2.random(0,1).best policy
    T ← Hrate*T
    W ← Drate*W
Return(policy, value)
```

Figure 19 ES policy improvement algorithm pseudocode

5.3.2 Computational complexity

Time and space complexity are important factors to evaluate an algorithm performance and scalability. The methodology in this study comprised of two parts: simulation and training. In the former, the pricing is simulated in the environment considering decisions by platform, drivers, and requesters in $\boldsymbol{T}$ consecutive time steps (Section 5.1). In the latter, the data collected from the simulation is used to train a linear value function according to the MC-based SGD algorithm (Figure 18). These two steps are executed following one another iteratively till convergence achieved. The time and space complexity of these two procedures are $\mathbf{O}(BTMN^2)$ and $\mathbf{O}(TN^2)$, respectively. Here, $\boldsymbol{B}$ is the batch size, $\boldsymbol{T}$ is the number of time steps per episode, $\boldsymbol{N}$ is the number of zones, and $\boldsymbol{M}$ is the maximum number of drivers per zone. The quadratic term contributes the most to the complexity which occurs at the steps such as drivers relocation decisions between any pair of zones. While time complexity is true in the worst case, strategies such as sorting the candidate zones based on their expected gain in profit and compatibility with drivers' service time reduces the execution time in most of the cases. This is comparable to DP approach discussed before which has exponential complexity solving the problem in discrete space.

Quadratic complexity in term of input size does not hinder the real-time application of the methodology. The simulation and training are implemented using either historical available data or simulation environment. Then, the trained function is employed to make pricing decision in real time without a need for batch training and convergence. Any single update of function weights using real-time data costs $\mathbf{O}(N)$ and $\mathbf{O}(BTN)$, in time and space complexity, respectively.

5.3.3 Implementation

The RL-based simulation model and training described in the previous section are implemented following OpenAI Gym's Env class API (Brockman, Cheung et al. 2016) and executed using Python 3.7.4 scripting language. The Gym environment provides a standardized interface for the RL process. Our simulation environment, called "Shared Mobility Environment," uses several internal and external classes and functions to model crowds of drivers and delivery tasks.

Figure 20 shows how the environment works and interacts with other components. Functions named Initialize, Reset, and Step are the basic operands in most Gym environments. *Initialize* is used at the very beginning of the simulation to initialize the environment. *Reset* is used to reset the environment at the initial stage or when an episode is finished. *Step* is used to execute a given action in the environment and feedback its reward and next state observation to the agent. The action at each stage is selected using the policy selected randomly from concurrent environments. The concurrent environment policies are updated iteratively by the policy improvement ES function. When the required number of experienced episodes is collected (i.e., batch size), training the value function is implemented as the next step following the MC-based SGD algorithm.

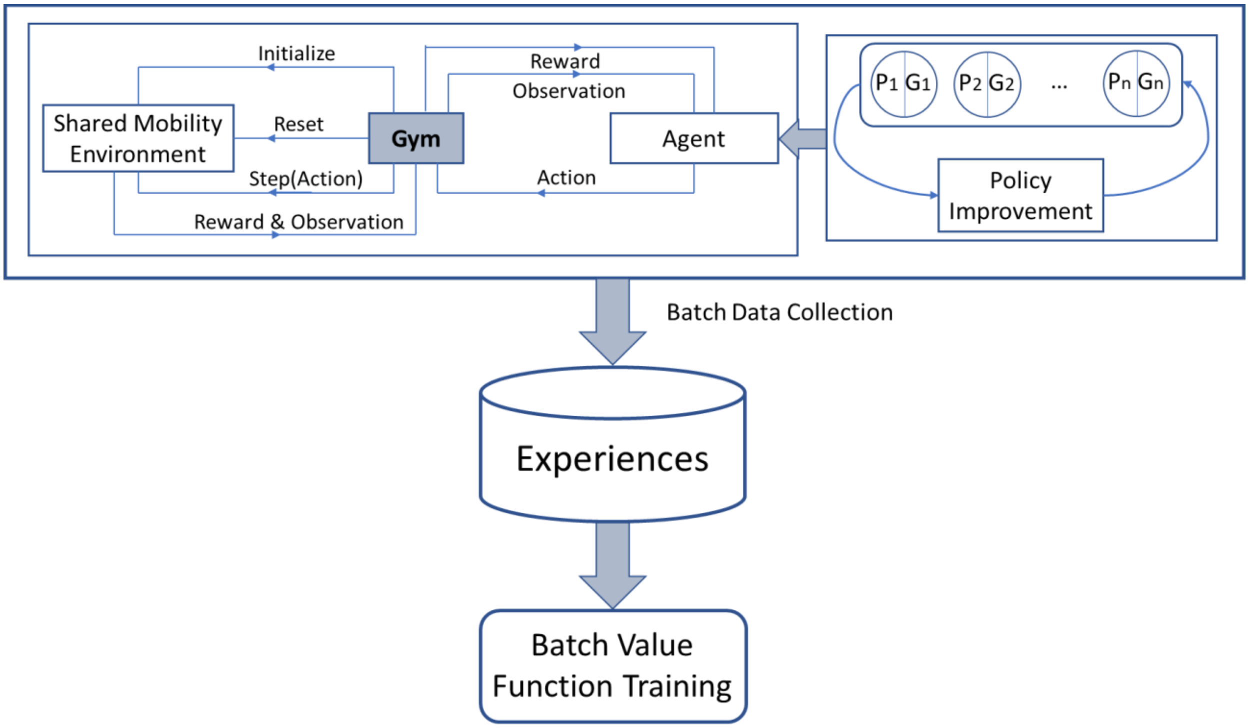


Figure 20 RL implementation in Python

5.4 Summary

In this chapter, the pricing mathematical model and simulation model framework is presented, and the mathematical formulation behind its RL function approximation modeling is described step by step. Based on the characteristics of our problem, the MC-based function approximator has been presented as the solution algorithm, and several improvement strategies such as ES policy improvement, memory replay, and concurrent environments have been discussed.

6. Results and Analysis

6.1 Case Study Description

This study considers San Francisco as a case study to evaluate the methodology. San Francisco, one of the largest metropolitan areas in the country, is undoubtedly the birthplace of many shared economy companies (e.g., DoorDash and Uber) that has transformed its industry and, in some cases, changed the world. Sharing economy companies have presented challenges to the city, particularly regarding how it manages transportation. On the other hand, it benefits thousands of residents and visitors. Regulators seek to find solutions to balance these positive and negative impacts on the city.

One vital resource to provide inputs for the model of this study is the San Francisco Bay Area Metropolitan Transportation Commission activity-based model (MTC-ABM). The activities or day patterns that drive individuals' need to make travel-related choices in time (hourly, 24 hours) and space (1454 TAZs) are based on MTC's 2000 Bay Area Travel Behavior Survey (Jaller, Pourrahmani et al. 2019).

Food delivery requests are derived selecting the home-based eat-out trips from the MTC travel model. Trips are selected from those featuring "eat-out" activity as the primary tour purpose, and the duration of the "eat out" activity does not exceed 60 minutes. Each trip is associated with a set of features such as a pair of origin (i.e., home zone) and destination zones (i.e., food place zone), departure hour of the day, and trip transport mode. For each potential request, a minute is a random number generated from a uniform distribution. The figure below shows the trip mode and distance distribution (Figure 21). The number of trips is the highest for midday (MD), followed by the morning (AM) and afternoon (PM). A high proportion of trips are made in active modes (walk/bike) and car. The walk/bike trip distance is less than 1.5 miles, and car trips are within 6 miles of travel distance. The fraction of active modes decreases from AM to PM, while that of trips made by car increases from AM to PM.

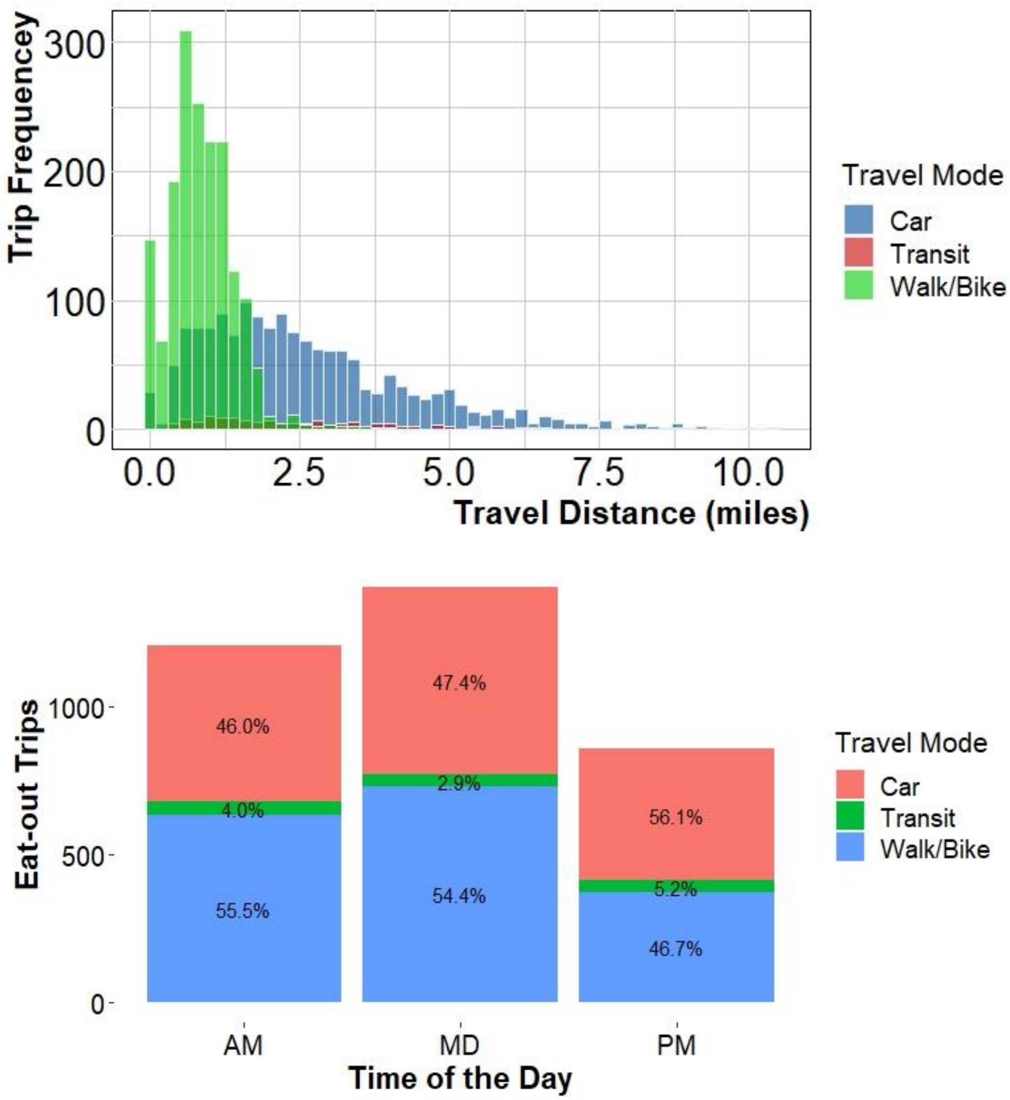
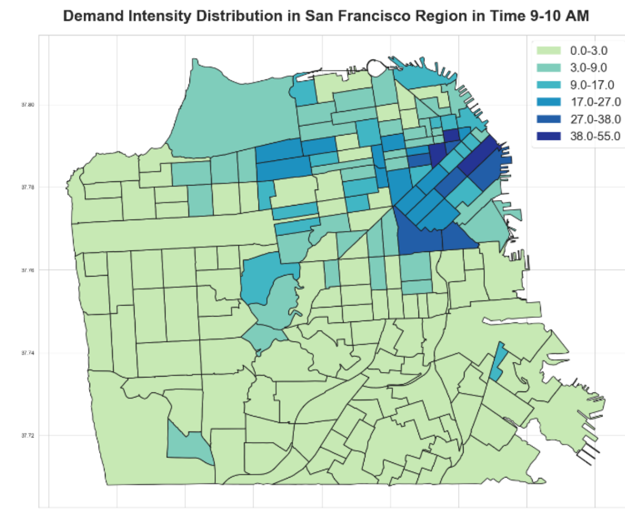
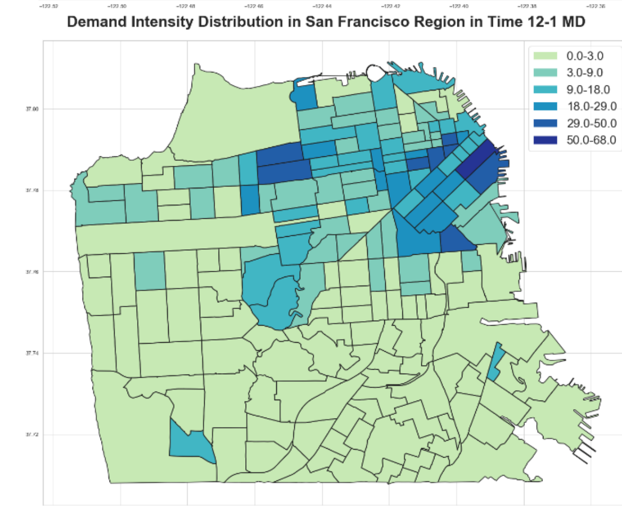


Figure 21 Eat out trip distribution description

Figure 22 shows the distribution of the eat-out destinations for a sample of 1 hour period. The trip destinations vary by time of day, expanding from the northern part to the center and middle of the city. Zones covering downtown areas and city centers have relatively higher demand than the rest of the zones.





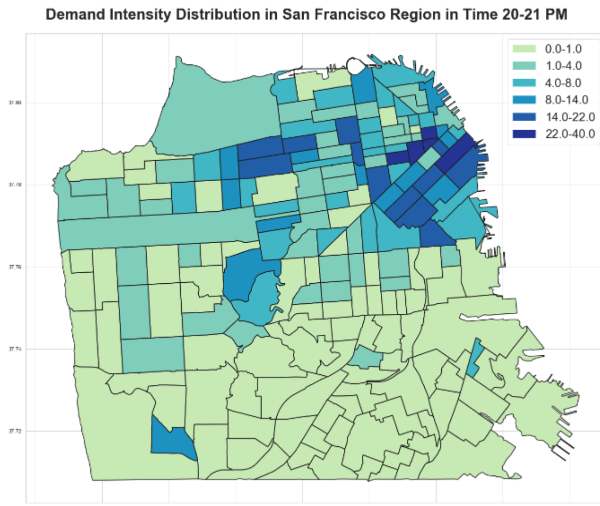


Figure 22 Eat-out trip destination distribution

6.2 Algorithm Parameter Settings

The algorithm has different sets of parameters that must be initialized to calibrate the model and method to this case study. The proposed MC-based SGD algorithm is characterized by several variables, among which learning rate, $\boldsymbol{\alpha}$, is very important. Specifically, the learning rate (lr) is a configurable parameter in learning algorithms with a small positive value, often in the range of 0 to 1. While it can remain fixed to a value, it is recommended to be decreased gradually in RL for better performance (Sutton and Barto 2018). Four scenarios were considered: in the first two scenarios, $\boldsymbol{\alpha}$ is fixed at 0.3 and 0.1. In the remaining two scenarios, it decreases across episodes, either continuously (by a factor of 0.99) or discretely (three intervals: 1 (episodes 1-1300), 0.3 (episodes 1300-2000), and 0.1 (episodes 2000-2500)). For the latter case, the cut-off episodes were chosen by repetitive examination of several alternatives and by observing algorithm performance. Each scenario is executed for the food delivery sample case study in San Francisco. The simulation time is one hour (9-10 AM), which comprises four steps (15 minutes each). The initial vehicle fleet size, 500, is randomly distributed in the zones. The parameters (i.e., time interval, wait time, speed, operation cost) are normalized between 0 and 1. The discount rate γ=0.9 is chosen. Each scenario has been repeated for 10 simulation seeds, and the prediction cost, mean square error (Eq. 5.17), and confidence interval per iteration are computed. The results are averaged over the last 500 iterations and depicted in Figure 23.

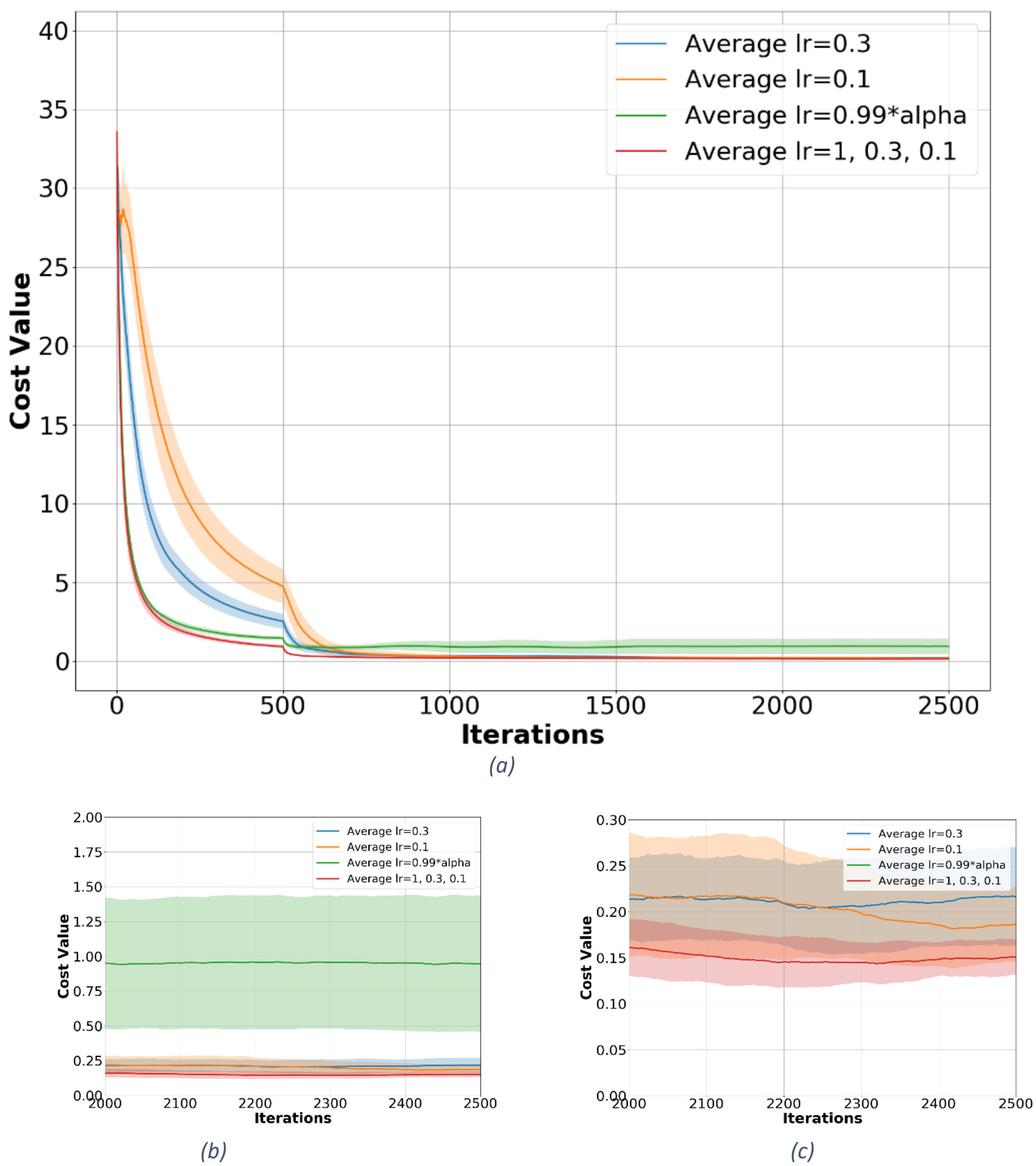


Figure 23 Algorithm convergence behavior

Up to about iteration 600, scenarios corresponding to =0.3 and 0.1 feature relatively larger confidence intervals than other scenarios. Starting from iteration 700, the prediction cost diverges for 0.99*$\boldsymbol{\alpha}$ and its confidence interval grows. Among the three remaining scenarios, the discretely decreasing $\boldsymbol{\alpha}$ scenario performs better with the least average cost and the most acute confidence interval and stable convergence behavior across all the episodes.

Two other important parameters are the number of episodes and the number of environments, $\boldsymbol{m}$**.** For the former, the experiments showed that the total number of episodes in the range of 1,500‐2,500 is appropriate for convergence. Later, a similar analysis was conducted to set the environment size parameters: $\boldsymbol{m}$=5, 10, , and 15 were examined. The convergence behavior and the policy value did not show a meaningful trend. Thus, $\boldsymbol{m}$=10, was selected for future analysis.

To demonstrate the effect of the ES policy improvement algorithm and the multiple concurrent environments, three sets of 10 simulation seeds were executed with the same parameters and case study: 1) policy improvement and concurrent environments; 2) policy improvement and memory replay, and 3) random policy generator. Policy improvement parameters are set empirically as $\boldsymbol{T}$ = 100, ***Boltzman_constant*** = 1, ***Hrate*** = 0.99, ***Drate*** = 0.95, and $\boldsymbol{W}$ = [1, 2]. Figure 24 shows the current and average discounted cumulative value of the policy examined per episode.

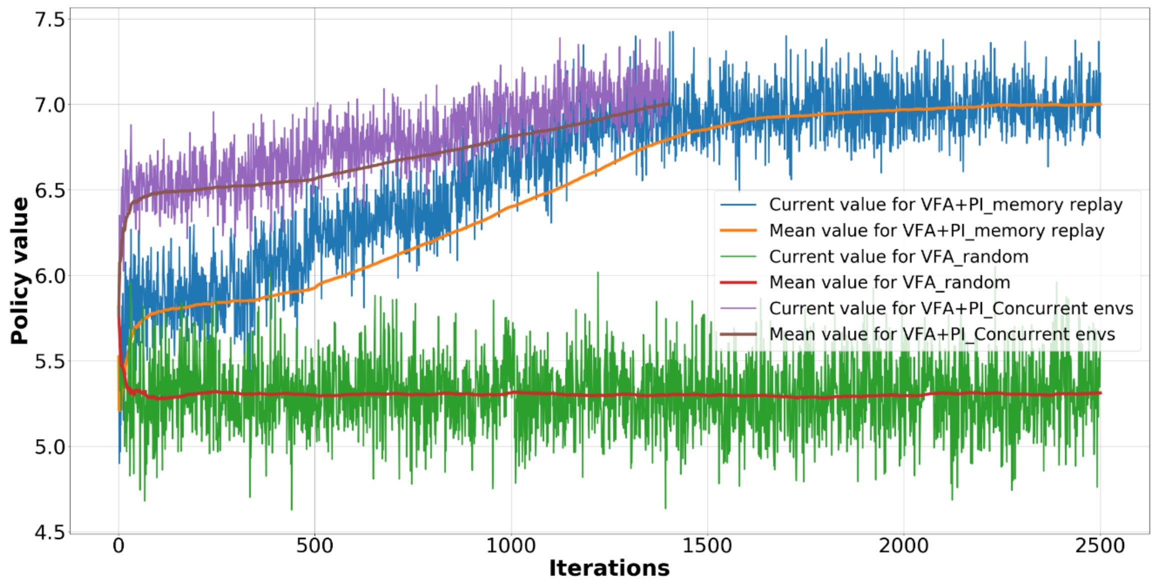


Figure 24 Policy value progress using the proposed policy improvement algorithm (VFA+PI), memory replay, concurrent environments, and random policy (VFA)

Although all scenarios show enough variations in the examined policies for training, the policy improvement heuristic has dramatically increased the policy values and provided a better training set than random policy generators. Also, implementing concurrent environments has improved the quality of the policy training set and converged to the ***value* =** 7 faster than using memory replay for more than 1,000 episodes.

To examine the dynamic zone-based pricing strategy performance, two other pricing strategies, static and myopic pricing, are also implemented. In the static pricing, the delivery rate per mile is fixed at the base rate, without any variation across zones and time. The myopic strategy, however, varies the price only in temporal dimension. It determines one surge multiplier for all zones, potentially varying from time to time. As might already be clear from its name, the difference lies in the fact that in the myopic strategies, the future discounted reward is disregarded and only the immediate reward is considered in value estimation per time stage.

The policy improvement algorithm introduced previously is used to solve the problem for myopic strategy in the form of generalized policy iteration algorithm. In this way, the algorithm starts with one random policy and improves it iteratively until the difference between two consecutive policies is small enough. The policy values are total earning and deliveries for profit and social welfare scenarios, respectively. Here, no SGD and value function learning is conducted.

6.3 Food Delivery Scenarios

To perform the analysis for food delivery, 9-10 AM eat out trips (1,202 trips) were considered as case study. In next sections, the results from different scenarios are evaluated considering various features and measures. The results are organized into four groups: platform, drivers, requesters, and system. This enables to evaluate the model and results from different perspectives.

All results and data points presented in this chapter are the average of 35 simulation runs. There are abbreviations sometimes used to display the results in tables and graphs. Table 9 provides full definitions for each of these abbreviations and words for more information.

Table 9 Abbreviations and definitions

Abbreviation	**Definition**
SW	Dynamic zone-based pricing strategy with social-welfare maximization objective
My-SW	Myopic pricing strategy with social-welfare maximization objective
Pr	Dynamic zone-based pricing strategy with profit maximization objective
My-Pr	Myopic pricing strategy with profit maximization objective
R	Random driver distribution
W[1]	Weighted driver distribution

[1] Initial pool of drivers is distributed with respect to a probability function weighted by presence of demand in the region and time

W-NC[1]	Weighted driver distribution with no choice to reject participation
GC	General cost
TT	Travel time
TD	Travel distance
PC	Parking cost
PT	Parking time
OC	Operation cost for cars
VOT	Value of time
FDF	Food delivery fare

6.3.1 Platform

This section compares several pricing strategies and algorithms: static pricing, myopic pricing, and dynamic zone-based pricing schemes. The total platform deliveries and profit for various fleet sizes and distributions are displayed in Table 10 and Similar trends are observed for the total profit in Table 11. **An increase in fleet size and transitioning from a random distribution increase the total earning in all scenarios**. **Pr surpassed the other two scenarios, which is clearer between Pr and static**. T-tests confirmed the significance of the difference in these results for all the scenarios.

Table 11.

In Table 10, the number of deliveries for all strategies increases with fleet size. Also, **transitioning to weighted distribution from random increases the deliveries significantly**. Implementing no choice strategy for drivers has also slightly increased the number of deliveries for most of the scenarios. **Comparing the three pricing schemes, we observe that the number of matches is higher for SW than for myopic and static schemes in almost all distributions**. This change for some scenarios is large, although it

[1] Similar to the W distribution, only that in this scenario the drivers have no choice to reject participation in the matching pool. In other words, Eq. 5.5 and 5.6 are relaxed from the mathematical model.

might be negligible in quantity compared to other schemes for some scenarios. As the numbers seem close for some scenarios, standard t-tests were conducted to verify the significance of difference in average deliveries between SW and two other pricing schemes. The results confirmed the difference's significance for most scenarios at a 95% confidence interval level (Appendix A).

Table 10 Total deliveries for different fleet size, drivers' distribution, and pricing schemes

Fleet size	R			W			W-NC		
	SW	My-SW	Static	SW	My-SW	Static	SW	My-SW	Static
50	66	49	46	94	87	89	109	87	89
100	119	111	110	165	160	163	167	160	161
400	335	284	324	541	523	531	561	523	545
800	490	432	483	819	746	813	839	749	823
1200	609	567	593	957	891	941	997	899	996
1400	668	622	637	990	971	952	1,055	982	1,042
1800	745	709	709	1,034	1,006	1,005	1,117	1,034	1,110

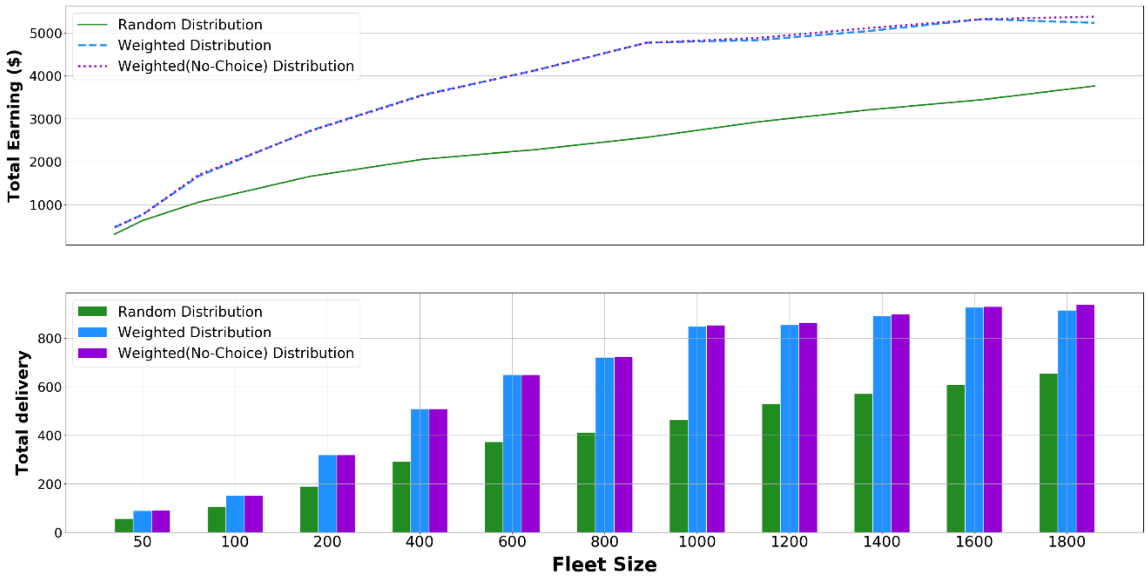
Similar trends are observed for the total profit in Table 11. **An increase in fleet size and transitioning from a random distribution increase the total earning in all scenarios. Pr surpassed the other two scenarios, which is clearer between Pr and static**. T-tests confirmed the significance of the difference in these results for all the scenarios.

Table 11 Total profit for different fleet size, drivers' distribution, and pricing schemes

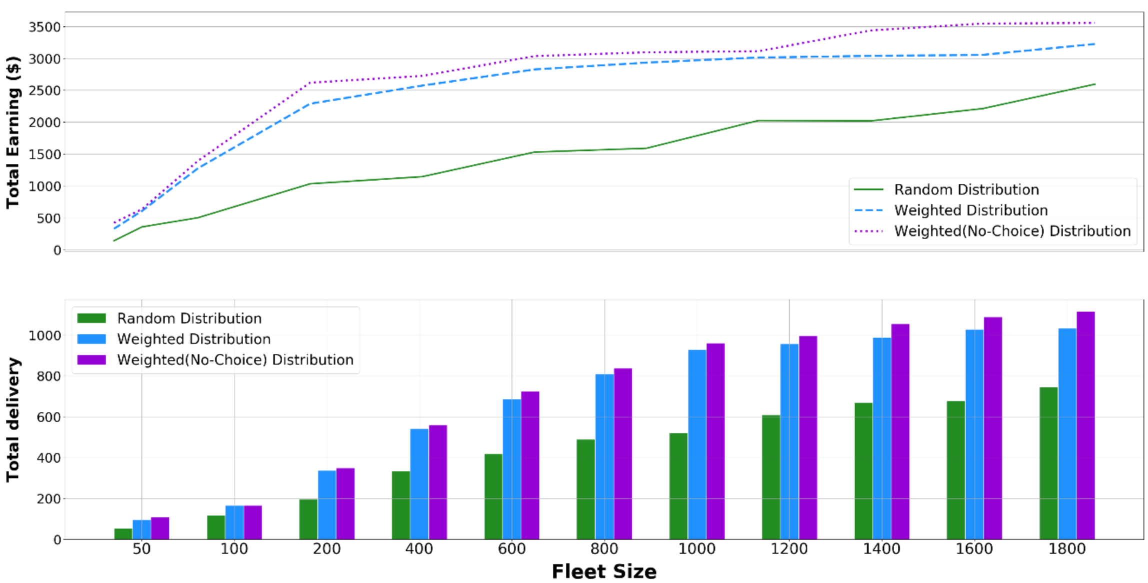
Fleet size	R			W			W-NC		
	Pr	My-Pr	Static	Pr	My-Pr	Static	Pr	My-Pr	Static
50	311	301	104	466	419	131	477	419	131
100	629	619	199	767	734	249	766	734	247
400	1,659	1,608	653	2,724	2,696	859	2,713	2,696	890
800	2,277	2,224	947	4,120	3,873	1,528	4,122	3,909	1,596
1200	2,927	2,879	1,152	4,829	4,712	1,791	4,879	4,741	1,907
1400	3,208	2,973	1,233	5,044	4,793	1,855	5,112	4,844	2,048
1800	3,760	3,544	1,358	5,229	4,973	2,056	5,374	5,095	2,277

Figure 25 presents the total earning/profit and deliveries for Pr and SW across a range of fleet sizes (50‑1800). The total earning for Pr ranges from $500 to more than $5,000, whereas the maximum is about $3,500 for SW. On the other hand, the total deliveries range from less than 100 to more than 1,000 for SW as fleet size increases. In Pr, the largest total delivery is about 900, achieved by a fleet size of 1,800. **In both Pr and SW, the random distribution of drivers is inefficient in terms of total earning and deliveries achieved**.

Another difference between these two scenarios is evident in the impact of no-choice strategy for drivers on deliveries and earnings. In Pr, this impact is hardly noticeable as it almost overlaps with the weighted distribution scenario. On the other hand, no choice implementation for SW has increased both the total earnings and deliveries for all fleet sizes, compared with the weighted distribution scenario.



(a)



(b)

Figure 25 Total earning and deliveries for (a) Pr and b) SW scenarios

To evaluate the stability of price per zone in time, the standard deviation of prices in consecutive time steps is measured for each zone, and the average of this zonal price deviation is presented for different scenarios across fleet sizes in Figure 26. Accordingly, **prices are more stable for the Pr scenario as the average deviation mainly stands**

between 0 and 0.2, while this value grows to 0.6-0.9 in SW. An increase in fleet size slightly mitigates the deviation in both scenarios.

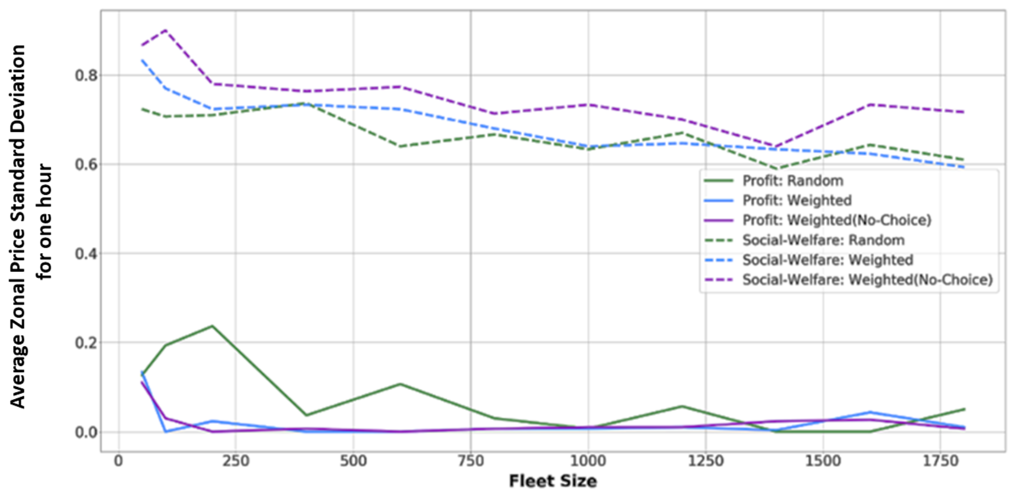


Figure 26 Average temporal price deviation per zone

Error! Reference source not found. Figure 27 displays the surge multipliers versus delivery distance for sample scenarios. Here, according to the results, two scenarios were selected for Pr and SW, representing worst (R, 50) and best (W-NC, 1800) cases. It is observed that surge multipliers are highest and almost constant in Pr, while there are many variations in SW. This confirms the finding from Figure 26. Also, **SW has increased the delivery distance compared to Pr** in both worst and best cases.

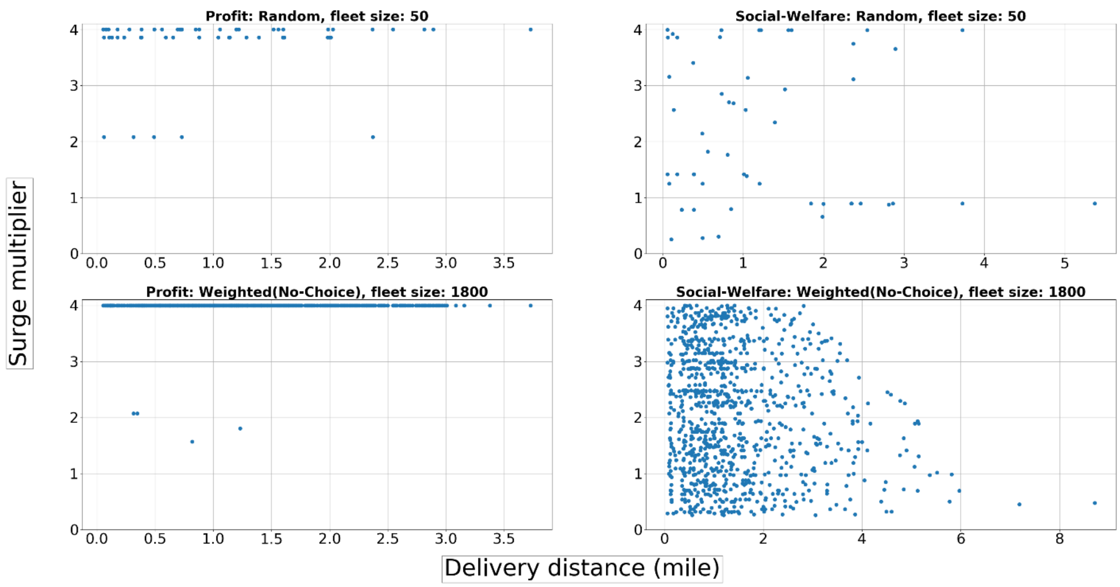
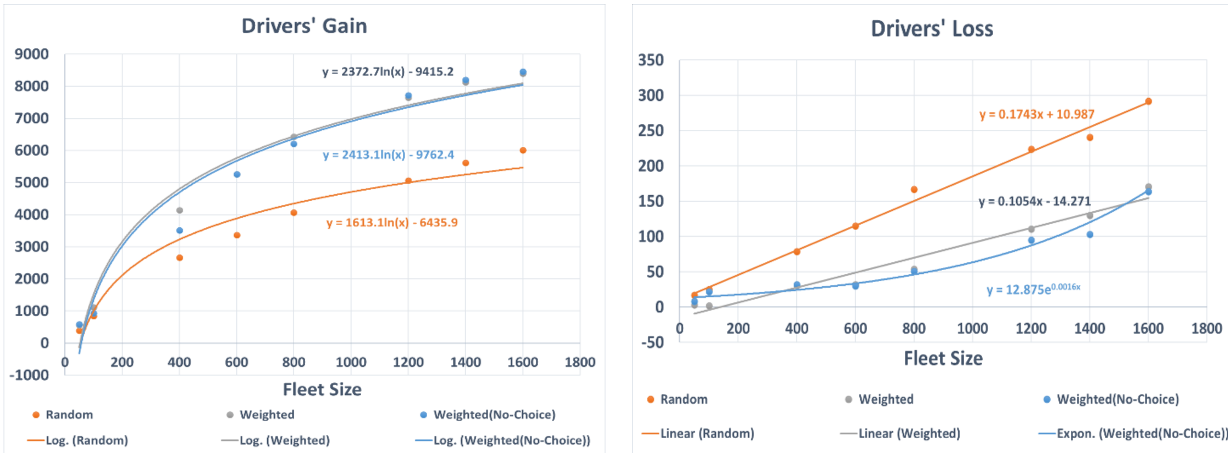


Figure 27 Surge multiplier vs. delivery distance for matched requesters and drivers

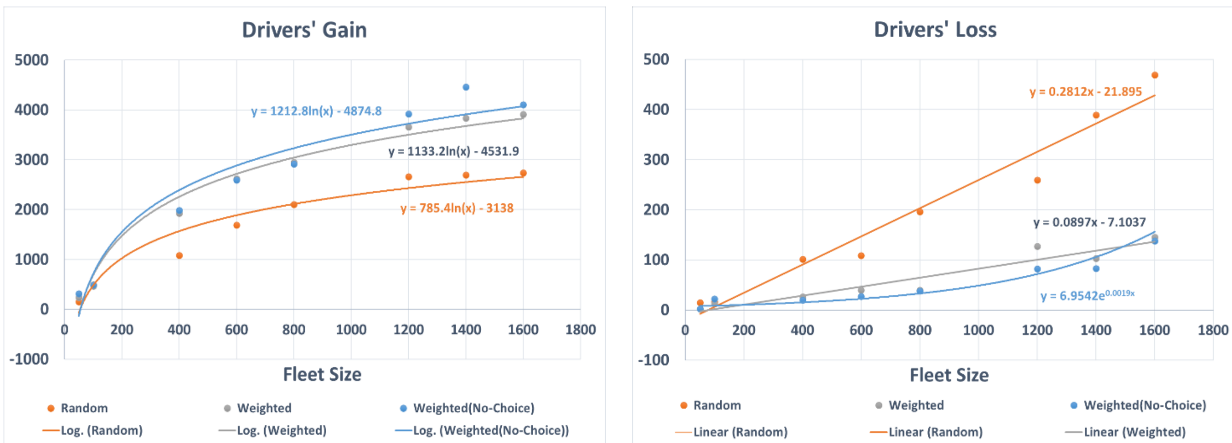
6.3.2 Drivers

In this section, the model is evaluated from the drivers' perspective. To measure the model performance effectively for drivers only, we set the demand side as fixed by turning off the elasticity function in the model. In other words, equations 5.1-5.4 are relaxed from the mathematical model, and total potential demand equals the total actual requesters. It must be noted that the level of earning/profit is higher here compared to the section 6.3.1.

Figure 28 presents the total drivers' profit gain and loss across various scenarios and settings. Here, profit refers to the drivers earning from deliveries minus the cost from relocations when they are not matched with any requester and decide to relocate. The drivers' gain shows the total drivers' profit only for those drivers with positive profit, in other words, for those who are earning rather than losing money by participating in the system. On the other hand, drivers' loss shows the profit for drivers with negative profit for the same scenarios and settings.



(a)



(b)

Figure 28 Drivers' profit gain and loss in (a) Pr and (b) SW scenarios

The rate of increase in gain surpasses the loss between equivalent scenarios. An increase in fleet size and transitioning from random to weighted distributions improves drivers' earnings and decreases their loss. Gain follows a logarithmic trend in all scenarios, whereas loss for those has a linear trend for random and weighted distributions and exponential trend for the weighted (no choice) case. This indicates that **if the fleet size increases dramatically in the long run, loss surpasses gain.** Note that weighted (no choice) cases in both Pr and SW have the highest and lowest gain and loss, respectively. Similar to the observation we had for the platform, weighted (no choice) overlaps the weighted distribution in gain for Pr, while this increased the gain over for SW. **In general, the Pr framework benefits drivers more than SW, as it poses higher gains and lower losses for drivers**.

Figure 29 displays the relocation distances driven by drivers when they were not assigned to a delivery task in their current zone. Here, the no-match relocation indicates the relocation driven to another zone which did not eventually result in a delivery task in the destination zone. It is observed that **random distribution has the highest relocation among scenarios**. **Weighted distribution of drivers and no choice strategy implementation have decreased the relocation distances for drivers by guiding drivers to more fruitful zones and increasing the matching rates**. SW has lower relocation and no-match relocation distances than in the Pr scenarios. We also observed that SW has more deliveries than Pr; however, it is worse than Pr in terms of drivers' gain and loss. This indicates that **higher offered prices in Pr benefit drivers more and compensate for larger relocations and smaller number of matches when considering all participating drivers.**

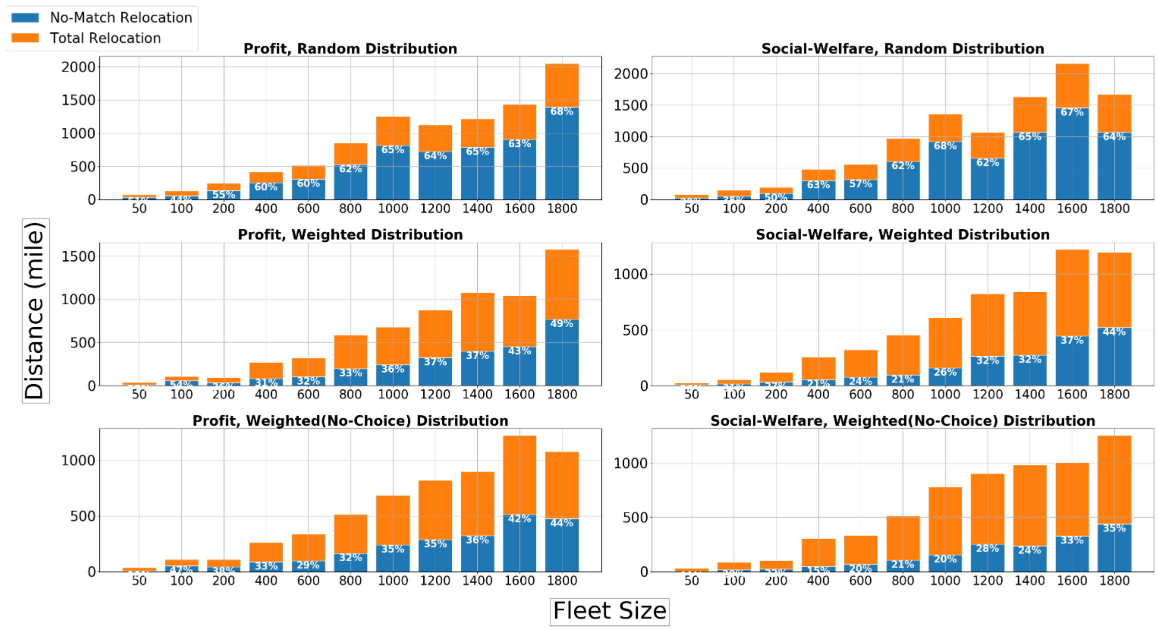


Figure 29 Drivers relocation distances

6.3.3 Requesters

This section attempts to evaluate the system from requesters' perspectives. Figure 30 presents the share of requesters and deliveries from total demand for various scenarios and settings. The rate of change in the number of deliveries grows fast up to fleet size 1,000 and then becomes relatively stable in scenarios with the weighted distribution. **The SW grows the portions of requesters and deliveries more compared with Pr; however, it is more successful in increasing the requesters rather than the deliveries. Increasing the fleet size, driver weighted distribution and no choice implementation are effective in converging the portion of requesters and deliveries.**

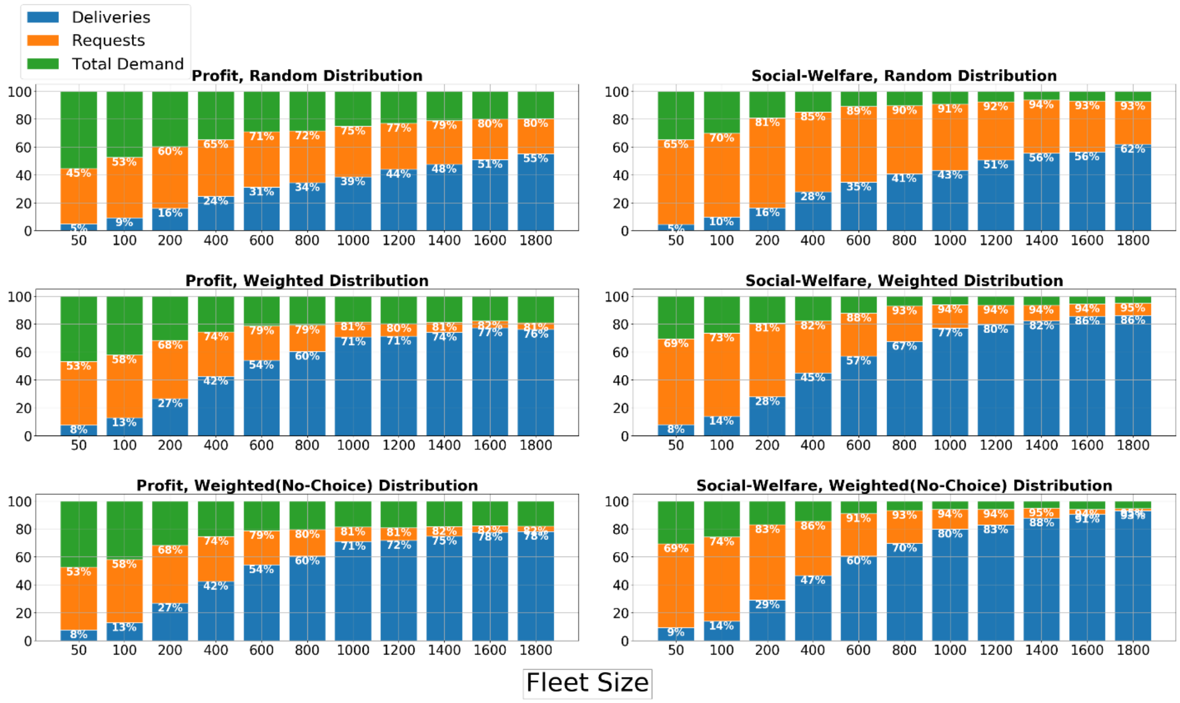


Figure 30 Portions of requests and deliveries

Although the pricing model is a macro simulation model and its outputs are aggregated at the zonal level, it is still possible to gather results at the individual level but with wider confidence intervals. Our results showed that the average delivery fare per individual requester is about \$5-6 and \$3-3.50 for Pr and SW scenarios, respectively. Similarly, wait times are 6-7 minutes for Pr users and 7-8 minutes for SW users. These values vary according to fleet size and drivers 'distributions in the system. Z-tests are conducted to understand the significance of the difference between mean wait time and delivery fare values in equivalent Pr and SW scenarios. Z-tests were conducted in two levels: one among different fleet sizes for the same scenario and settings to understand the effect of fleet size on individual wait times and fare and another between Pr and SW results for the same fleet size and distributions to understand the significance of delivery fare and wait time variations. Results revealed that fleet size does not affect individual delivery fares in Pr, but it only affects their wait time for W and W-NC distributions when a fleet size increase to about 800 or more. A similar analysis for SW shows that increase in fleet size affects individuals' delivery fare. Next, the second category compared Pr with SW. Both delivery fee (R: starting from fleet size 100; W: starting from fleet size 200; W-NC starting from fleet size 600) and wait time (R: starting from fleet size 600; W: starting from fleet size 400; W-NC starting from fleet size 200) are significantly different between these two scenarios within equivalent settings.

A series of general cost analyses were conducted for various scenarios to evaluate requesters' experience switching to online food delivery from original eat-out trips (Figure 31). Here, general cost (GC) is calculated for each requester mode.

$$\mathbf{GC}_{\text{car-trip}} = (\mathbf{TT}_{\text{in-bound}} + \mathbf{TT}_{\text{out-bound}} + \mathbf{PT}_{\text{food-place}})*\mathbf{VOT} + (\mathbf{TD}_{\text{in-bound}} + \mathbf{TD}_{\text{out-bound}})*\mathbf{OC} + \mathbf{PC}_{\text{food-place}} \tag{6.1}$$

$$\mathbf{GC}_{\text{bike-trip, walk-trip}} = (\mathbf{TT}_{\text{in-bound}} + \mathbf{TT}_{\text{out-bound}})*\mathbf{VOT} \tag{6.2}$$

$$\mathbf{GC}_{\text{food-delivery}} = \mathbf{FDF} \tag{6.3}$$

In equations 6.1-6.2, restaurants' service fees and waiting times are not included. It is also assumed that taxes and tips are equal in both online and in-place food order requests.

The change in GC resulting from switching to online food delivery for requesters is shown in Figure 31. Here, R and W-NC distributions are considered, as they previously were found to be the worst and best settings according to total deliveries and earnings per fleet size. The left and right figures display the total gains (those with positive GC) and losses (those with negative GC) in GC for all matched requests in the system**. The loss in GC surpasses the gain for all scenarios**. **SW has higher gains and lower loss amounts than Pr due to its lower offered delivery fees and more matched requests**. Comparing Figure 31 with what we have for drivers' gain and loss in Figure 28, it is noted that **SW and Pr have the opposite**

effect on requesters and drivers' gain and loss. Whereas Pr is more suitable for drivers in that it enables larger gains and smaller losses, it is expensive for requesters, whose losses and gains are the worst.

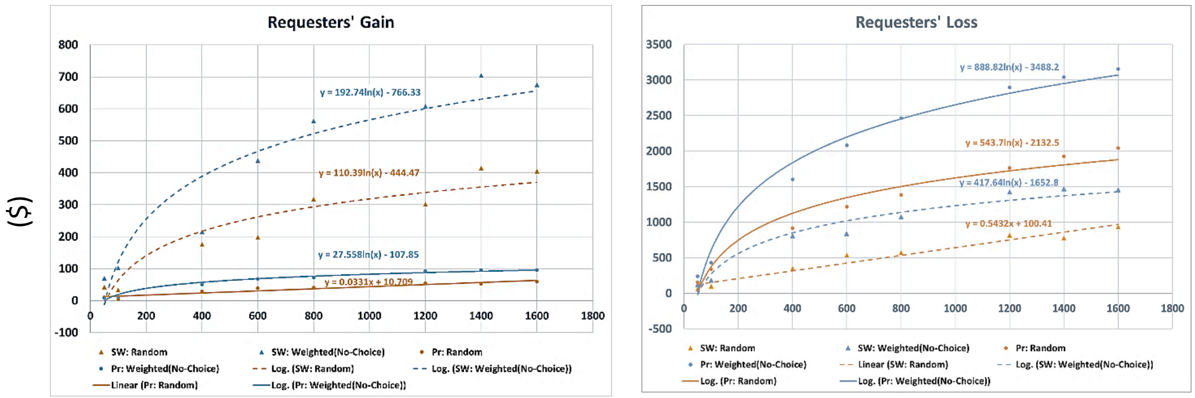


Figure 31 Requesters' change in general cost after switching from eat-out trip to online food delivery

To get more insights on requesters' experience, total and per capita values of time saved and change in GC are summarized in Table 12. To estimate an interval for each measure, two scenarios, *Random distribution: fleet 50* and *Weighted (no choice) distribution: fleet 1800* are considered. More requesters are bikers for both Pr and SW, followed by cars and walkers (we disregarded the public transit users for this analysis). The per capita of saved time and GC are highest for walkers**. Each walker, on average, was able to save from 30 minutes up to about an hour by switching to online food delivery. Also, they saved $3-4 in GC in the SW context. For the cars and bikers, the estimated intervals for GC are always negative**: cars lost 30 cents to $4, and bikers lost $1-3 on average, varied by scenarios. **Based on utility theory (if the impact of all other factors are fixed), drivers' and bikers' value of comfort for food delivery must at least equal these estimated changes in GC to choose this option over an eat-out trip.**

Table 12 Delivery requesters experience

Scenario	Mode	Number of deliveries		Time Saved				General Cost Saved ($)			
				Total (hr)		Mean (min)		Total		Mean	
		L[1]	U[2]	L	U	L	U	L	U	L	U
Pr	Car	16	295	3	54	11	11.5	-(61)	-(1,222)	-(4)	-(4)
	Bike	38	621	5	108	7.5	10.5	-(82)	-(1,846)	-(2)	-(3)
	Walk	3	9	1.5	6	28	41	-(3)	-(22)	-(1)	-(2.5)
SW	Car	24	374	5	79	12	13	-(6)	-(151)	-(0.3)	-(0.5)
	Bike	39	623	5.5	108	8.5	10.5	-(24)	-(541)	-(1)	-(1)
	Walk	4	12	3	12	46	57.5	11	45	3	4

Below, the change in GC against requesters' VOT is depicted for those two scenarios (Figure 32). Increasing the fleet size and transitioning to WNC distribution increases the deliveries for both Pr and SW. **In Pr, no negative GC is observed when requesters' VOT exceeds $30-$40 per hour, increasing fleet size from R: 50 to WNC: 1800. This cut-off range of VOT decreases to $20-30 in SW**.

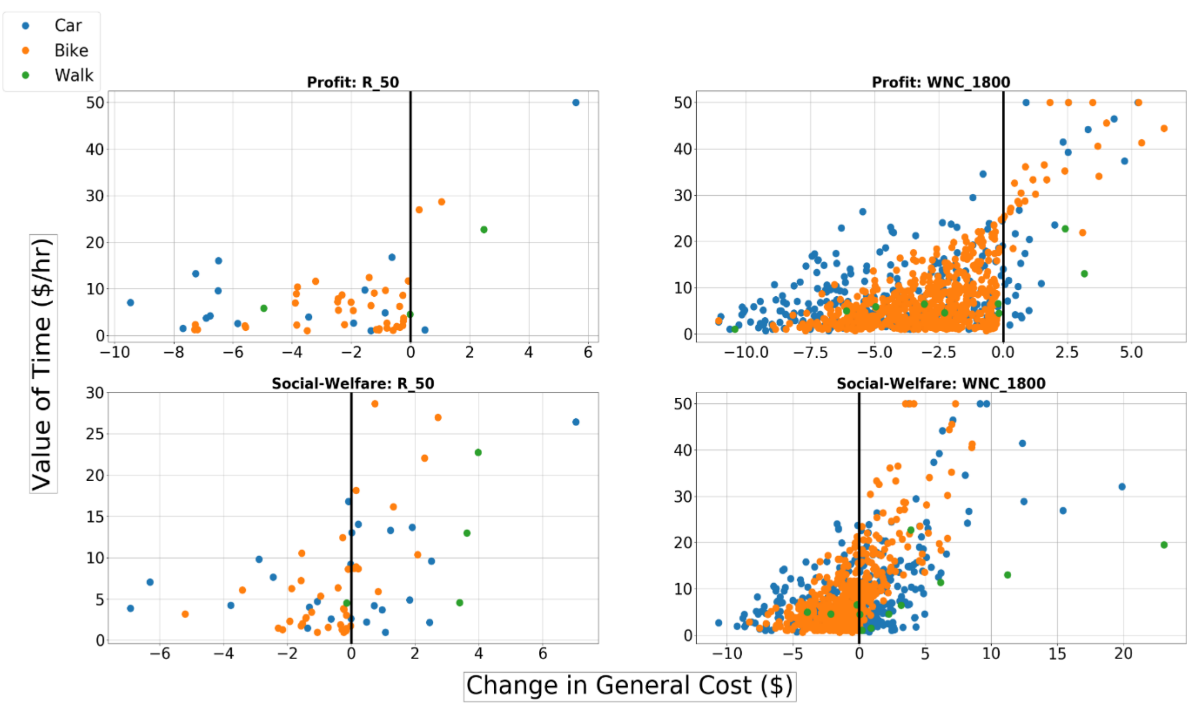
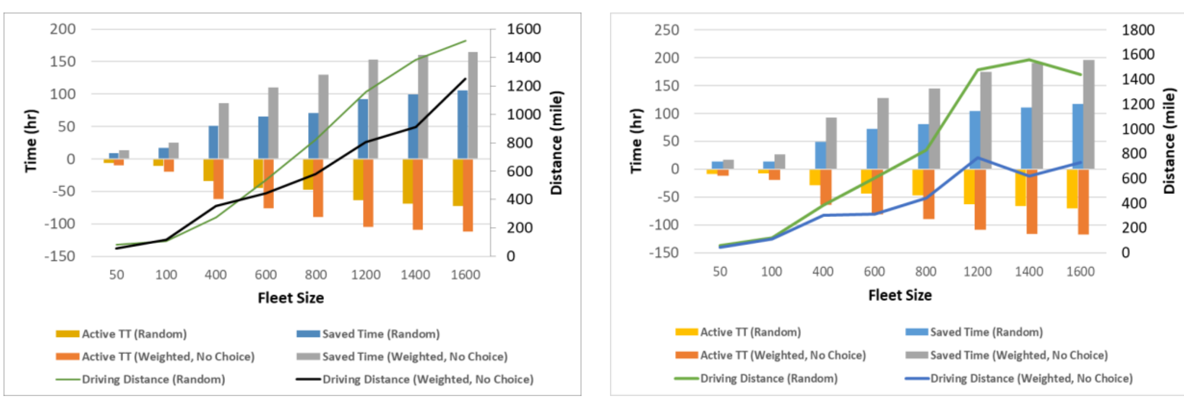


Figure 32 Change in general cost versus value of time for requesters

[1] Lower bound
[2] Upper bound

6.3.4 System and Environment

In this section, the service impact is evaluated from system and environment points of view. According to Figure 33, **the driving distance increases, and the active transport time decreases for all scenarios.** The increase in driving distance is dramatic for profit-maximizing scenarios**. This change indicates that the total delivery and relocation distances driven by delivey drivers can be more than that driven by requesters' personal cars during eat-out trips**. Replacing bike and walk eat-out trips, which demonstrate the significant portion of requesters, has decreased active transport time by delivery cars. This impacts individuals health in the long term. On the other hand, online food delivery generated additional productive time for individuals by saving their travel times to food places. This saved time might enable individuals to incorporate physical activity in ways they prefer, such as doing exercise at home and gyms or walking/bike for leisure.



(a) (b)

Figure 33 Change in total driving distance, active travel time, and total saved time by switching to online food delivery in a) Profit and b) Social-Welfare

Figure 34 shows the financial status of three major actors in the service. In all scenarios, drivers are the winners despite their loss due to relocations. Platforms come in second place. Here, platforms' commission is estimated as 20% of each delivery fare, and no further cost is assumed. Repeating the findings observed in the previous section, requesters are losers whose monetary loss surpassed the potential benefits from saving time in all scenarios at the aggregated level. This loss expands in Pr where the platform offers higher delivery fares. As the drivers and platform gains grow, requesters' loss exacerbates.

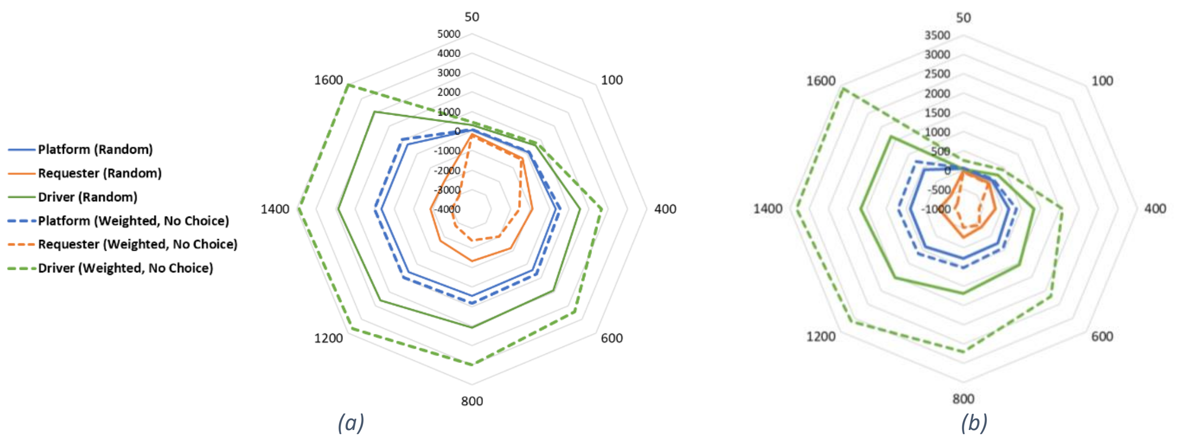


Figure 34 Total drivers' profit, platform's commission, and requesters' change in general cost across fleet sizes for a) Pr and b) SW

6.4 Summary

This chapter presents and discusses the results and analysis for a sample case study. In summary, a dynamic zone-based pricing scheme is more effective in increasing the platform's total earnings and deliveries than static and myopic strategies. The increase in fleet size and distribution of the drivers based on demand density increases the matching between drivers and requesters, thus increasing the deliveries and earnings dramatically in return. Profit maximizer scenario (Pr) achieved larger earnings for every fleet size than the social welfare scenario (SW) while having fewer deliveries. This is because Pr decides higher surge multipliers than SW, which might use lower surge multipliers to increase the potential requesters and matches. Thus, Pr can maximize the total earnings with fewer deliveries.

Implementing a no-choice strategy for the SW forces drivers to remain in the matching pool even though the surge multipliers are lower than expected earnings. This increases the number of deliveries by matching more drivers with requesters encouraged by the lower-level delivery fees offered by the platform.

On the other hand, no-choice strategy did not make a distinctive impact in the Pr case. This is because Pr offers higher level prices; thus, drivers are more interested in participating in the matching pool and being matched though they can choose to reject. Zonal price deviations indicate more stability pricing decisions for Pr than SW, where there are many variations. Delivery distance has been increased up to twice the one for Pr by SW in some cases. Dynamic zone-based pricing works based on per mile fees. Thus, lower surge values offered in SW make delivery possible for requesters with longer delivery distance within reasonable price satisfying their elasticity constraints.

Drivers' experiences demonstrate that transitioning to weighted distributions and no-choice strategies can mitigate the loss and improve the benefits. Relocation distances for drivers were highest in random distributions where supply does not match the demand needs efficiently. In general, the portion of no-match relocation distances was lower for SW, where the matching rate is slightly higher. However, the higher offered prices for drivers in Pr still can compensate the relocation costs, referring to higher benefits and lower costs in Pr versus SW. Another point is the effect of the no-choice strategy on drivers: the results showed that forcing drivers to participate in the matching pool increases deliveries and earnings in total and benefits drivers by increasing their gain and decreasing their loss.

Regarding demand, SW has increased the potential pool of requesters based on its lower offered prices. However, not all attracted requests have received a delivery. Increasing fleet size, the weighted distributions, and the no-choice strategy improved the number of deliveries. Analyzing users' experiences showed that general cost is negative for most requesters as their delivery cost exceeds their value of time saved by avoiding eat-out trips. Users' VOT determines their preference to replace eat-out physical trips with online food delivery.

From a system point of view, transitioning to food delivery has increased the driving distance in the system, resulting not just from delivery distances but also the relocations. Substituting the active mode trips has contributed to an increase in driving mileage and decreased the physical activity time for these users. On the other hand, time saved for all users might enable them to get their physical activity needs in other non-transportation forms. Analyzing the proft of each actor in the system demonstrates that drivers have the highest profit in all scenarios, followed by a platform that earns 20% per delivery without bearing any loss. Requesters are losing in general cost, which is the reverse of drivers' and platforms' earnings.

7. Conclusion

This study explored crowdshipping applications in the context of online food delivery services. Through a literature review of crowdshipping research, it identified pricing as an important understudied factor in the service operation. Subsequently, the pricing schemes of four food delivery platforms were explored by analyzing sets of empirical data on online food deliveries. Several items in pricing were detected and formalized for various online food delivery apps. According to a sample of delivery requests, the fees paid by users in addition to the food price account for 30-50% of the total food delivery price.

Delivery fees, referring to the money charged for transporting the food from the place where it was made to the requester's location, were identified as an important item in the pricing schemes of apps that can affect the total cost. While delivery fees are assumed to be paid to the drivers as compensation for their fulfillment of the delivery tasks, there is no transparency regarding how these fees are determined by the respective platforms. The fees may, for example, depend on factors other than the delivery distance or the time of the day. In contrast, in the passenger rideshare market the pricing is proportional to the transportation distance or time, with variations in time and region accounting for real-time changes in supply and demand. Setting the price in proportion to the delivery distance/time is psychologically appealing to users and mitigates pricing confusion. Thus, introducing a variation in delivery fees based on changes in demand across time and space potentially enhances the service performance in terms of the number of deliveries and profitability. However, while surge pricing based on travel distance or time has been well studied in the passenger rideshare market literature, it is still under-explored in delivery pricing studies.

This study formulated and solved a dynamic zone-based pricing model for a many-to-many food delivery service using crowds of car drivers. Delivery fees were assumed to be proportional to delivery distance. To account for the variation in pricing across region and time, it was modeled as a dynamic programing multistage decision-making problem wherein the zonal surge multipliers are updated at each time interval. The state of the problem transitions between time intervals following drivers' assignment and relocation decisions as well as new drivers' arrival, depending on the driving operation cost, expected zonal earnings, and drivers' service time window in the system. Drivers' preferences and requesters' elasticities were incorporated in the mathematical model. Optimal zonal surge multipliers were determined for each time interval considering the total expected returns for the entire time horizon. The problem was solved separately for two objective functions: 1) total profit maximization and 2) total delivery maximization.

Considering the problem size and complexity, a simulation-optimization solution framework was presented. First, a reinforcement learning agent plays in a macro simulation environment developed in Python following the dynamic programing model rules. The agent takes pricing actions and collects batches of sample experience data from environmental feedback to train a linear action value function approximator using the Monte Carlo and Stochastic Gradient Descent methods. To maintain a balance between the exploration and exploitation of the pricing policy, an evolutionary strategy is designed and employed in a concurrent environment. Next, the optimal action value function is optimized to find the optimal pricing scheme in scenarios designed by varying the fleet size as well as drivers' distribution and preferences.

The proposed model and method were applied for a sample of eat-out trips in San Francisco as the case study. The performance of the proposed dynamic zone-based pricing was compared to that of static and myopic pricing strategies. The results indicated that the pricing model in this study increased the total profit and deliveries for almost all scenarios compared to the static and myopic strategies. Increased fleet size and the efficient distribution of drivers increased the deliveries and profits for both platforms and drivers. However, increasing the fleet size, i.e., the market thickness, increased the idle drivers in the system by decreasing the probability of their being matched with requesters. Drivers' participation decisions affected the system only when the pricing level was low or the fleet size was significantly lower than the demand in the system. Forcing drivers to accept the matches offered by the platform increased the supply and total deliveries in the social welfare scenario, where low surge multipliers were unable to tempt drivers to stay. High relocation distances increased the driving mileage in the system. Efficient driver distribution and relocation guidance were identified as some of the remedies to mitigate excess driving distance in the system. The analysis of requesters' experiences revealed that they mostly lost in terms of general cost when shifting from personal eat-out trips to online food delivery. The gain in general cost usually happened to those individuals who had a high VOT or long travel time from their home to the restaurant; for such individuals, saving time represented the main advantage of online food delivery compared to personal eat-out trips.

In general, the findings indicated that the dynamic zone-based pricing increases the earning for both platform and drivers compared to static pricing, however it might limit the requests with longer delivery distance or lower willingness to pay. Absence of an accurate short-term demand/price level prediction system causes the drivers to bear the operation cost of repositioning which might not lead to a delivery assignment. This in return increases the driving mileage and its' traffic externalities in the system. Thus, there is a need for efficient driver distribution and repositioning decisions to reduce the relocation distances for motorized vehicles. Replacing delivery vehicles with clean and environmentally friendly transport modes was identified as an alternative solution. For food delivery to be competitive with other options, such as physical trips to restaurants, strategies to reduce the wait times and delivery fees or providing food price discounts might be the most effective. Membership subscription programs could also help platforms implement these strategies for loyal users while still making a profit through regular membership fees. Finally, the proposed simulation-optimization model provides flexibility and benefits the researchers, platform providers, and policy makers to examine various pricing schemes and evaluate the results from different perspectives.

This study can be improved in several ways. First, the empirical data collection can be expanded to capture a larger sample size as well as possible additional sources of data on restaurants' temporal workloads and traffic conditions. The conjoint analysis can be replicated using other types of conjoint surveys, such as choice-based models, and variables such as income level can be included in the elasticity function estimation. Meanwhile, equipping the pricing model with effective demand prediction models, introducing penalties for unmatched requests, and adding uncertainty to drivers' repositioning decisions can all enhance the model's applicability. Examining matching strategies other than same-area matching and allowing for multiple deliveries per vehicle are also suggested additions to further develop the model in future studies. Finally, implementing deep reinforcement learning techniques or other nonlinear function approximators for pricing shows good potential for improving the simulation accuracy and speed.

7.1 Summary

This section summaries the findings from this dissertation in several categories.

7.1.1 Platform provider

1. Dynamic pricing model increased the total profit and deliveries compared to that of static and myopic strategies in most of the scenarios (Appendix A). The magnitude of the increase in deliveries and profit are presented in Tables 1 and 2, respectively. The rates vary depending on fleet size (50-1800).

Table 13 Increase in deliveries by dynamic pricing versus alternative pricing strategies in SW scenarios

Alternative strategy	Driver distribution		
	Random distribution	Weighted distribution	Weighted distribution with driver no choice
Static	1-48%	1-6%	0-22%
Myopic	5-35%	2-10%	4-25%

Table 14 Increase in profit by dynamic pricing versus alternative pricing strategies in Pr scenarios

Alternative strategy	Driver distribution		
	Random distribution	Weighted distribution	Weighted distribution with driver no choice
Static	2-3 times	2-4 times	2-4 times
Myopic	2-8%	1-11%	1-14%

2. In dynamic pricing strategy, deliveries (in SW), and profit (in Pr) increase by 0.3% with a 1% increase in fleet size. Also, efficient distribution of drivers (from random distribution to weighted distribution) increases the deliveries (in SW) and profits (in Pr) by 40-70% and 20-80%, respectively.

3. Eliminating the option for drivers to decide to participate in the matching pool significantly increases the total deliveries in the SW by about 2-16% (t test results are attached).
4. Surge multipliers per zone are more stable across time for the Pr scenario as the average deviation from the mean stands between 0 and 0.2, while this is within the range of 0.6–0.9 in SW.
5. SW scenarios cover deliveries up to 6 miles and more, while deliveries rarely exceed 3 miles in Pr scenarios due to higher rage of prices.
6. Compared to PR, SW increases the requesters and deliveries by 21% and 14%, respectively; however, since the rate of increase in demand is higher than deliveries, the ratio of delivery to demand is still higher for Pr.

7.1.2 Drivers

1. An increase in fleet size and transitioning from random to weighted distributions improves drivers' earnings and decreases their loss.
2. Increasing the fleet size, i.e., the market thickness, increases the idle drivers in the system by decreasing the probability of their being matched with requesters. This indicates that if the fleet size increases dramatically in the long run, loss in general cost surpasses gain for drivers.
3. Weighted distribution of drivers and implementing no choice strategy decrease the relocation distances for drivers by guiding them to zones with higher matching rates.

7.1.3 Requesters

1. According to a sample of delivery request data collected from Davis, Sacramento and San Francisco, the extra fees paid by requesters in addition to the food price account for 30-50% of the total food delivery cost.
2. Whereas Pr is more suitable for drivers in that it enables larger gains and smaller losses, it is expensive for requesters, whose losses surpass gains.
3. The per capita of saved time and generalized cost are highest for those who walk to restaurants for their meal in the case study. They, on average, was able to save from 30 minutes up to about an hour by switching to online food delivery. Also, they saved $3–4 in general cost in the SW scenario which includes lower surge multipliers.
4. For the car drivers and bikers in the case study, the estimated change in general cost is always negative: drivers lose 30 cents to $4, and bikers lose $1-3 on average.
5. In Pr, no negative general cost is observed when requesters' VOT exceeds $30‑40 per hour. This decreases to $20‑30 in SW in the case study.

7.1.4 Environment and health

1. Total delivery and relocation distances driven by delivery drivers can be more than that driven by requesters' personal cars during eat-out trips in the case study.
2. Replacing bike and walk eat-out trips, which demonstrate the significant portion of requesters in the case study, decreases active transport time in the system. This impacts individuals' health in the long term.

Appendix A T-test results

Table 15 T test results for number of deliveries resulted from different pricing strategies with random distribution

Fleet size	Algorithm	Mean	Std	t-stat	p-value
50	SW	65.89426	6.97615	13.96563	7.54E-16
	Static	46.34117	0.758947		
	SW	65.89426	6.97615	12.09472	5.51E-14
	My	48.80817	0.634683		
100	SW	118.7391	7.133645	7.745639	4.79E-09
	Static	110.1441	0.579655		
	SW	118.7391	7.133645	6.957459	5.03E-08
	My	111.1332	0.203117		
400	SW	334.8235	15.45603	4.666028	4.57E-05
	Static	324.1778	0.903327		
	SW	334.8235	15.45603	20.1068	1.67E-20
	My	284	0.647643		
800	SW	489.7969	11.89771	3.427369	0.001557
	Static	482.7408	2.526658		
	SW	489.7969	11.89771	25.90954	2.45E-24
	My	432	1.44714		
1200	SW	608.6226	5.557777	15.24435	3.38E-23
	Static	592.5963	5.128743		
	SW	608.6226	5.557777	40.68041	2.17E-39
	My	567	2.483706		
1400	SW	668.4154	6.847546	19.4896	1.22E-27
	Static	636.8517	5.433231		
	SW	668.4154	6.847546	35.84474	9.44E-29
	My	622.1332	0.441285		
1800	SW	745.03	14.16569	15.64751	2.28E-17
	Static	708.7743	1.436663		
	SW	745.03	14.16569	15.65258	3.15E-17
	My	709	1.036847		

Table 16 T test results for number of deliveries resulted from different pricing strategies with weighted distribution

Fleet Size	Algorithm	Mean	Std	t-stat	p-value
50	SW	94.31713	4.109609	8.276138	3.27E-11
	Static	89.17623	2.31862		
	SW	94.31713	4.109609	12.43202	6.02E-16
	My	86.74331	1.403094		
100	SW	165.4677	5.715476	3.514536	0.001095
	Static	162.7743	1.436663		
	SW	165.4677	5.715476	6.768213	8.70E-08
	My	160.1332	0.198737		
400	SW	540.5629	6.683313	7.129384	1.34E-09
	Static	530.6913	5.221494		
	SW	540.5629	6.683313	15.54152	2.75E-17
	My	523	0.835293		
800	SW	818.9941	8.831761	3.92	0.000407
	Static	813.422	2.101428		
	SW	818.9941	8.831761	48.93	1.52E-32
	My	746	0.680127		
1200	SW	957.0775	6.548961	14.39469	2.00E-20
	Static	941.1817	4.081176		
	SW	957.0775	6.548961	62.74976	2.13E-45
	My	891	2.914695		
1400	SW	989.6146	7.318166	29.98775	3.19E-30
	Static	952.3998	2.297825		
	SW	989.6146	7.318166	15.74848	4.25E-18
	My	971	1.927386		
1800	SW	1033.762	15.10703	10.78947	7.01E-13
	Static	1004.548	3.227382		
	SW	1033.762	15.10703	10.2989	5.15E-12
	My	1006	0.873861		

Table 17 T test results for number of deliveries resulted from different pricing strategies with weighted-No choice distribution

Fleet Size	Algorithm	Mean	Std	t-stat	p-value
50	SW	108.89	2.494438	40.54686	2.67E-49
	Static	89.17623	2.31862		
	SW	108.89	2.494438	45.33678	1.91E-52
	My	86.74331	1.807383		
100	SW	167.07	3.399346	12.27476	3.02E-17
	Static	160.5951	1.594992		
	SW	167.07	3.399346	15.24808	8.25E-17
	My	160.1332	0.138464		
400	SW	561.35	4.642796	20.77516	2.03E-26
	Static	544.9457	2.168871		
	SW	561.35	4.642796	48.50689	1.56E-37
	My	523	1.268478		
800	SW	838.78	1.885618	5.187243	9.01E-06
	Static	823	15.19963		
	SW	838.78	1.885618	73.8425	7.55E-44
	My	749	7.875325		
1200	SW	996.89	10.7082523	0.973490545	0.337086046
	Static	995.7603872	1.095445077		
	SW	996.89	10.7082523	45.90339898	3.19E-32
	My	899	0.437762648		
1400	SW	1055.02	9.201449	12.31703	3.81E-15
	Static	1041.624	1.922499		
	SW	1055.02	9.201449	58.91688	9.71E-37
	My	982	0.885189		
1800	SW	1116.65	7.483315	3.592431	0.001003
	Static	1109.778	0.903327		
	SW	1116.65	7.483315	54.77534	3.98E-35
	My	1034	0.703801		

Table 18 T test results for earnings resulted from different pricing strategies with random distribution

Fleet Size	Algorithm	Mean	Std	t-stat	p-value
50	Pr	311.0521	12.35008	98.04248	5.68E-45
	Static	104.4435	1.918078		
	Pr	311.0521	12.35008	5.567427	3.07E-06
	My	301.05	0.863058		
100	Pr	628.5033	48.67123	63.4225	5.35E-37
	Static	198.9678	1.976133		
	Pr	628.5033	48.67123	2.173234	0.036814
	My	619.12	1.089068		
400	Pr	1658.692	21.08378	317.2889	6.34E-65
	Static	652.9727	3.350349		
	Pr	1658.692	21.08378	15.82029	1.44E-17
	My	1608.46	2.966073		
800	Pr	2276.687	7.037198	1000.065	7.12E-126
	Static	947.3814	5.189642		
	Pr	2276.687	7.037198	42.77769	2.04E-42
	My	2223.6	3.518976		
1200	Pr	2926.642	22.51545	354.8896	1.92E-96
	Static	1152.004	14.53837		
	Pr	2926.642	22.51545	10.54107	2.83E-12
	My	2879.3	1.160022		
1400	Pr	3208.411	24.93598	396.9135	1.86E-86
	Static	1233.293	11.70314		
	Pr	3208.411	24.93598	52.47812	3.38E-34
	My	2972.66	0.852251		
1800	Pr	3760.498	5.971806	1681.804	1.36E-158
	Static	1357.741	5.662206		
	Pr	3760.498	5.971806	201.7195	2.60E-61
	My	3543.94	1.969388		

Table 19 T test results for earnings resulted from different pricing strategies with weighted distribution

Fleet Size	Algorithm	Mean	Std	t-stat	p-value
50	Pr	466.0975	8.728761	203.2166	8.30E-82
	Static	131.1633	6.027423		
	Pr	466.0975	8.728761	27.77847	6.34E-36
	My	419.05	5.675832		
100	Pr	766.543	7.313751	455.5789	3.04E-79
	Static	249.3001	2.251615		
	Pr	766.543	7.313751	30.5646	2.53E-26
	My	733.91	0.135374		
400	Pr	2723.526	15.7238	641.5226	4.84E-124
	Static	859.054	12.97096		
	Pr	2723.526	15.7238	8.938172	3.86E-12
	My	2695.71	7.707444		
800	Pr	4119.634	19.5306	616.2672	3.91E-111
	Static	1527.666	13.02436		
	Pr	4119.634	19.5306	68.82159	2.93E-39
	My	3873.01	3.05741		
1200	Pr	4828.973	12.35375	1238.136	5.27E-133
	Static	1790.892	7.316854		
	Pr	4828.973	12.35375	54.79535	1.06E-40
	My	4711.83	3.695038		
1400	Pr	5043.654	0.960914	5099.845	8.96E-113
	Static	1854.753	3.921225		
	Pr	5043.654	0.960914	625.657	8.86E-90
	My	4793.44	2.214689		
1800	Pr	5229.27	11.46156	956.6133	8.00E-142
	Static	2055.864	12.64061		
	Pr	5229.27	11.46156	96.3601	4.01E-59
	My	4973.24	8.688418		

Table 20 T test results for earnings resulted from different pricing strategies with weighted-No choice distribution

Fleet Size	Algorithm	Mean	Std	t-stat	p-value
50	Pr	477.1424	11.65222	174.6296	2.02E-81
	Static	131.1633	6.027423		
	Pr	477.1424	11.65222	35.13179	1.41E-28
	My	419.05	0.772615		
100	Pr	766.0307	11.9571	290.4382	2.56E-71
	Static	246.9514	3.796881		
	Pr	766.0307	11.9571	19.91322	3.71E-22
	My	733.91	3.072888		
400	Pr	2713.385	16.42027	705.5262	9.03E-121
	Static	890.4463	7.553633		
	Pr	2713.385	16.42027	7.692981	6.04E-09
	My	2695.71	0.00599		
800	Pr	4122.104	14.94131	823.2119	5.76E-77
	Static	1593.539	2.385206		
	Pr	4122.104	14.94131	68.31493	4.13E-38
	My	3908.82	0.755589		
1200	Pr	4879.444	12.47817	1458.886	9.16E-92
	Static	1907.331	2.733379		
	Pr	4879.444	12.47817	67.59558	1.47E-38
	My	4740.62	1.198254		
1400	Pr	5112.286	14.51307	1108.448	8.60E-86
	Static	2047.761	3.52328		
	Pr	5112.286	14.51307	96.71927	4.10E-46
	My	4844.22	3.006909		
1800	Pr	5374.119	4.143501	4121.125	3.59E-103
	Static	2276.851	0.674008		
	Pr	5374.119	4.143501	373.6875	1.32E-64
	My	5095.05	0.505014		

Table 21 T test results for number of deliveries resulted from weighted and weighted-no choice distribution

Fleet Size	Algorithm	Mean	Std	t-stat	p-value
50	SW-W	94.31713	4.109609	17.93	< .00001
	SW-WNC	108.89	2.494438		
100	SW-W	165.4677	5.715476	1.42	.164713
	SW-WNC	167.07	3.399346		
400	SW-W	540.5629	6.683313	15.11	< .00001
	SW-WNC	561.35	4.642796		
800	SW-W	818.9941	8.831761	12.96	< .00001
	SW-WNC	838.78	1.885618		
1200	SW-W	957.0775	6.548961	18.76	< .00001
	SW-WNC	996.89	10.7082523		
1400	SW-W	989.6146	7.318166	32.91	< .00001
	SW-WNC	1055.02	9.201449		
1800	SW-W	1033.762	15.10703	29.08	< .00001
	SW-WNC	1116.65	7.483315		

References

Acito, F. and A. K. Jain (1980). "Evaluation of conjoint analysis results: a comparison of methods." Journal of Marketing Research **17**(1): 106-112.

Afeche, P., Z. Liu and C. Maglaras (2018). "Ride-hailing networks with strategic drivers: The impact of platform control capabilities on performance." Rotman School of Management Working Paper(3120544): 18-19.

Akeb, H., B. Moncef and B. Durand (2018). "Building a collaborative solution in dense urban city settings to enhance parcel delivery: An effective crowd model in Paris." Transportation Research Part E: Logistics and Transportation Review **119**: 223-233.

Allahviranloo, M. and A. Baghestani (2019). "A dynamic crowdshipping model and daily travel behavior." Transportation Research Part E: Logistics and Transportation Review **128**: 175-190.

Amazon's Mechanical Turk (MTurk) (2005).

Archetti, C., M. Savelsbergh and M. G. Speranza (2016). "The vehicle routing problem with occasional drivers." European Journal of Operational Research **254**(2): 472-480.

Arslan, A. M., N. Agatz, L. Kroon and R. Zuidwijk (2018). "Crowdsourced delivery—A dynamic pickup and delivery problem with ad hoc drivers." Transportation Science **53**(1): 222-235.

Bai, J., K. C. So, C. S. Tang, X. Chen and H. Wang (2019). "Coordinating supply and demand on an on-demand service platform with impatient customers." Manufacturing & Service Operations Management **21**(3): 556-570.

Banerjee, S., C. Riquelme and R. Johari (2015). "Pricing in ride-share platforms: A queueing-theoretic approach." Available at SSRN 2568258.

Bellman, R. (1957). "A Markovian decision process." Journal of mathematics and mechanics **6**(5): 679-684.

Bernot, K. (2019). "Your tips aren't going to Instacart and DoorDash drivers the way you'd think." from https://thetakeout.com/tips-instacart-doordash-drivers-base-pay-1832330324.

Besbes, O., F. Castro and I. Lobel (2021). "Surge pricing and its spatial supply response." Management Science **67**(3): 1350-1367.

Brockman, G., V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba (2016). "Openai gym." arXiv preprint arXiv:1606.01540.

Cachon, G. P., K. M. Daniels and R. Lobel (2017). "The role of surge pricing on a service platform with self-scheduling capacity." Manufacturing & Service Operations Management **19**(3): 368-384.

Caitlyn Hitt. (2020). "The Tasty History of Takeout and Delivery."

Captain, S. (2019). "DoorDash reveals how much it relies on customer tips to pay its workers." from https://www.fastcompany.com/90305854/exclusive-doordash-reveals-how-much-it-relies-on-customer-tips-to-pay-its-workers.

Carsten Hirschberg, A. R., Thomas Schumacher, Martin Wrulich (2016). The changing market for food delivery mckinsey.
Carter, G. (2019). "How To Make Money On The Side With Amazon Flex." from https://www.listenmoneymatters.com/amazon-flex/.
Castillo, J. C., D. Knoepfle and G. Weyl (2017). Surge pricing solves the wild goose chase. Proceedings of the 2017 ACM Conference on Economics and Computation.
Cavadas, J., G. H. de Almeida Correia and J. Gouveia (2015). "A MIP model for locating slow-charging stations for electric vehicles in urban areas accounting for driver tours." Transportation Research Part E: Logistics and Transportation Review **75**: 188-201.
Čepin, M. (2011). Dynamic Programming. Assessment of Power System Reliability, Springer**:** 253-255.
Chen, L., A. Mislove and C. Wilson (2015). Peeking beneath the hood of uber. Proceedings of the 2015 internet measurement conference.
Chen, M. K. and M. Sheldon (2016). Dynamic Pricing in a Labor Market: Surge Pricing and Flexible Work on the Uber Platform. Ec.
Chen, Y., X. Li and M. Sun (2015). "Competitive mobile targeting." Available at SSRN 2575192.
Choi, T.-M., S. Guo, N. Liu and X. Shi (2020). "Optimal pricing in on-demand-service-platform-operations with hired agents and risk-sensitive customers in the blockchain era." European Journal of Operational Research **284**(3): 1031-1042.
Devari, A., A. G. Nikolaev and Q. He (2017). "Crowdsourcing the last mile delivery of online orders by exploiting the social networks of retail store customers." Transportation Research Part E: Logistics and Transportation Review **105**: 105-122.
Dholakia, U. (2015). "Why do consumers hate Uber's surge pricing." Psychology Today URL https://www. psychologytoday. com/blog/the-science-behind-behavior/201512/why-do-consumers-hate-uber-s-surge-pricing.
Diakopoulos, N. (2015). "How Uber surge pricing really works." The Washington Post **17**.
Difallah, D., E. Filatova and P. Ipeirotis (2018). Demographics and dynamics of mechanical turk workers. Proceedings of the eleventh ACM international conference on web search and data mining.
Dupuis, M., B. Endicott-Popovsky and R. Crossler (2013). An analysis of the use of amazon's mechanical turk for survey research in the cloud. ICCSM2013-Proceedings of the International Conference on Cloud Security Management: ICCSM.
Ermagun, A. and A. Stathopoulos (2018). "To bid or not to bid: An empirical study of the supply determinants of crowd-shipping." Transportation Research Part A: Policy and Practice **116**: 468-483.
Feldman, P., A. E. Frazelle and R. Swinney (2018). "Service delivery platforms: Pricing and revenue implications." Available at SSRN **3258739**.
Garg, N. and H. Nazerzadeh (2021). "Driver surge pricing." Management Science.

Gatta, V., E. Marcucci, M. Nigro, S. Patella and S. Serafini (2019). "Public transport-based crowdshipping for sustainable city logistics: Assessing economic and environmental impacts." Sustainability **11**(1): 145.

Gdowska, K., A. Viana and J. P. Pedroso (2018). "Stochastic last-mile delivery with crowdshipping." Transportation Research Procedia **30**: 90-100.

Generation IM, F. Preston, N. Kukrika, M. Jaller and H. S. Matthews. (2020). "The Carbon Footprint of Retail: Ecommerce vs Bricks & Mortar." from https://www.generationim.com/research-centre/insights/ecommerce-vs-bricks-mortar/.

Ghilas, V., E. Demir and T. Van Woensel (2013). Integrating passenger and freight transportation: Model formulation and insights. Proceedings of the 2013 Beta Working Papers (WP), Technische Universiteit Eindhoven. **441**.

Ghilas, V., E. Demir and T. Van Woensel (2016). "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines." Computers & Operations Research **72**: 12-30.

Giuliano, G. (2013). Tackling Environmental Issues in Major Cities. Urban Freight in Livable Cities. C. Wolmar. Goteborg, The Volvo Research and Educational Foundations, VREF**:** 44-53.

Guda, H. and U. Subramanian (2017). "Strategic surge pricing and forecast communication on on-demand service platforms." SSRN Electronic Journal.

Guda, H. and U. Subramanian (2019). "Your uber is arriving: Managing on-demand workers through surge pricing, forecast communication, and worker incentives." Management Science **65**(5): 1995-2014.

Gurvich, I., M. Lariviere and A. Moreno (2019). Operations in the on-demand economy: Staffing services with self-scheduling capacity. Sharing economy, Springer**:** 249-278.

Hall, J., C. Kendrick and C. Nosko (2015). "The effects of Uber’s surge pricing: A case study." The University of Chicago Booth School of Business.

Helling, B. (2021). "Is DoorDash Worth It For Drivers?".

Holguín-Veras, J. and F. Aros-Vera (2014). "Self-Supported Freight Demand Management: Pricing and Incentives." EURO Journal on Transportation and Logistics **3**(1): 1-24.

Holguín-Veras, J., S. Hodge, J. Wojtowicz, C. Singh, C. Wang, M. Jaller, F. Aros-Vera, K. Ozbay, A. Weeks and M. Replogle (2018). "The New York City off-hour delivery program: A business and community-friendly sustainability program." Interfaces **48**(1): 70-86.

Holguín-Veras, J., M. Jaller and T. Wachtendorf (2013). "Improving Postdisaster Humanitarian Logistics: Three Key Lessons from Catastrophic Events." TR News(287).

Holguín-Veras, J., N. Xu, M. Jaller and J. Mitchell (2015). "A Dynamic Spatial Price Equilibrium Model of Integrated Urban Production-Transportation Operations Considering Freight Delivery Tours." Transportation Science **50**(2): 489-519.

Ibbetson, R. (2019). How more than 1.5MILLION packages are delivered each DAY in New York City adding to gridlock on roads as the internet ordering economy booms, Mailonline.

Jain, A. K., V. Mahajan and N. K. Malhotra (1979). "Multiattribute preference models for consumer research: A synthesis." ACR North American Advances.

Jaller, M. and J. Holguín-Veras (2013). "Comparative Analyses of Stated Behavioral Responses to Off-hour Delivery Policies." Transportation Research Record (TRR) **2379,**: 18-28.

Jaller, M., C. Otero-Palencia and E. Pourrahmani (2020). Automation, Electrification and Shared Mobility in Freight. Pacific Southwest Region UTC. Davis, CA, Institute of Transportation Studies.

Jaller, M. and A. Pahwa (2020). "Evaluating the environmental impacts of online shopping: A behavioral and transportation approach." Transportation Research Part D: Transport and Environment **80**: 102223.

Jaller, M., W. Penagos, M. del Castillo, C. Calderón, D. Suero and P. Pedroza (2016). "Application of the Capability Maturity Model to Assess the Risk Management and Disaster Response Capabilities of a Region in a Developing Country." (In review).

Jaller, M., L. Pineda and H. Ambrose (2019). Evaluating the Use of Zero-Emission Vehicles in Last Mile Deliveries. Presented at the 98th Annual Meeting of the Transportation Research Board.

Jaller, M., E. Pourrahmani, C. Rodier and J. Bischoff (2019). Simulation-Optimization Framework to Evaluate a Sustainable First Mile Transit Access Program Using Shared Mobility. Transportation Research Board, Washington, D.C.

Jiang, Z.-Z., G. Kong and Y. Zhang (2020). "Making the most of your regret: A behavioral investigation of workers' relocation in on-demand platforms." Jiang, ZZ, G Kong, Y Zhang.

João Pedreda. (2020). "Restaurant Delivery Pricing Strategies are Evolving to Survive the Pressure from Delivery Platforms."

Joerss, M., J. r. Schröder, F. Neuhaus, C. Klink and F. Mann (2016) "Parcel delivery: The future of last mile." Travel, Transport & Logistics 32.

Kaelbling, L. P., M. L. Littman and A. W. Moore (1996). "Reinforcement learning: A survey." Journal of artificial intelligence research **4**: 237-285.

Kafle, N., B. Zou and J. Lin (2017). "Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery." Transportation research part B: methodological **99**: 62-82.

Kim, HS., and D.C. Hodgins (2017). "Reliablity and Validity of Data Obtained from Alocohol, Cannabis, and Gambling Populations on Amazon's Mechanical Turk." Psychology of Additive Behaviours 31(1):85-94.

Kinetic. (2019). "Consumer Fee Comparison: Food Delivery Companies."

Kritzinger, S., K. F. Doerner, R. F. Hartl, G. Kiechle, H. Stadler and S. S. Manohar (2012). Using Traffic Information for Time-Dependent Vehicle Routing. The Seventh International Conference on City Logistics, Mallorca, Spain, Elsevier.

Kruskal, J. B. (1969). Toward a practical method which helps uncover the structure of a set of multivariate observations by finding the linear transformation which optimizes a new "index of condensation". Statistical computation, Elsevier.

Kung, L.-C. and G.-Y. Zhong (2017). "The optimal pricing strategy for two-sided platform delivery in the sharing economy." Transportation Research Part E: Logistics and Transportation Review **101**: 1-12.
Lancaster, K. J. (1966). "A New Approach to Consumer Theory." Journal of Political Economics **74**(132-157).
Lapan, M. (2018). Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more, Packt Publishing Ltd.
Lardieri, A. (2019). Survey: People Willing to Pay More For Food Delivery, usnews.
Le, T. V. and S. V. Ukkusuri (2018). "Crowd-shipping services for last mile delivery: analysis from survey data in two countries." arXiv preprint arXiv:1810.02856.
Le, T. V. and S. V. Ukkusuri (2018). "Selectivity correction in discrete-continuous models for the willingness to work as crowd-shippers and travel time tolerance." arXiv preprint arXiv:1810.00985.
Louviere, J. J. (1988). "Conjoint Analysis Modelling of Stated Preferences - A Review of Theory, Methods, Recent Developments and External Validity." Journal of Transport Economics and Policy: 93-119.
Luce, R. D. and J. W. Tukey (1964). "Simultaneous conjoint measurement: A new type of fundamental measurement." Journal of mathematical psychology **1**(1): 1-27.
Lugtigheid, D., D. Banjevic and A. K. Jardine (2008). "System repairs: When to perform and what to do?" Reliability Engineering & System Safety **93**(4): 604-615.
Marcucci, E., M. Le Pira, C. S. Carrocci, V. Gatta and E. Pieralice (2017). Connected shared mobility for passengers and freight: Investigating the potential of crowdshipping in urban areas. 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), IEEE.
Mehmann, J., V. Frehe and F. Teuteberg (2015). Crowd logistics− a literature review and maturity model. Innovations and Strategies for Logistics and Supply Chains: Technologies, Business Models and Risk Management. Proceedings of the Hamburg International Conference of Logistics (HICL), Vol. 20, Berlin: epubli GmbH.
Mike Pomranz. (2021). "It's Not Just You–Food Delivery Is More Expensive Than It Used to Be."
Morana, J., J. Gonzalez-Feliu and F. Semet (2014). Urban consolidation and logistics pooling. Sustainable urban logistics: Concepts, methods and information systems, Springer**:** 187-210.
Morgan Stanley. (2020). "Can Food Delivery Apps Deliver Profits for Investors?".
Mortensen K., and Hughes T. L. (2018). "Comparing Amazon's Mechanical Turk Platform to Convesntional Data Collection Methods in the Health and Medical Research Literature." Journal of General Internal Medicine. 33(4): 533-538.
New York City Department of Transportation (2019). Improving the Efficiency of Truck Deliveries in NYC.

Nikzad, A. (2017). "Thickness and competition in ride-sharing markets." Available at SSRN 3065672.
Noah Lichtenstein. (2020). "The hidden cost of food delivery."
Paloheimo, H., M. Lettenmeier and H. Waris (2016). "Transport reduction by crowdsourced deliveries–a library case in Finland." Journal of Cleaner Production **132**: 240-251.
Pan, S., C. Chen and R. Y. Zhong (2015). "A crowdsourcing solution to collect e-commerce reverse flows in metropolitan areas." IFAC-PapersOnLine **48**(3): 1984-1989.
Pourrahmani, E. and M. Jaller (2021). "Crowdshipping in Last Mile Deliveries: Operational Challenges and Research Opportunities." Socio-Economic Planning Sciences: 101063.
Powell, W. B., H. P. Simao and B. Bouzaiene-Ayari (2012). "Approximate dynamic programming in transportation and logistics: a unified framework." EURO Journal on Transportation and Logistics **1**(3): 237-284.
Prokhorchuk, A., J. Dauwels and P. Jaillet (2019). "Stochastic dynamic pricing for same-day delivery routing." arXiv preprint arXiv:1912.02946.
Punel, A., A. Ermagun and A. Stathopoulos (2018). "Studying determinants of crowd-shipping use." Travel Behaviour and Society **12**: 30-40.
Punel, A. and A. Stathopoulos (2017). "Modeling the acceptability of crowdsourced goods deliveries: Role of context and experience effects." Transportation Research Part E: Logistics and Transportation Review **105**: 18-38.
Qi, W., L. Li, S. Liu and Z.-J. M. Shen (2018). "Shared mobility for Last-Mile delivery: Design, operational prescriptions, and environmental impact." Manufacturing & Service Operations Management **20**(4): 737-751.
Qualtrics (2002).
Rai, H. B., S. Verlinde and C. Macharis (2018). "Shipping outside the box. Environmental impact and stakeholder analysis of a crowd logistics platform in Belgium." Journal of cleaner production **202**: 806-816.
Rai, H. B., S. Verlinde, J. Merckx and C. Macharis (2017). "Crowd logistics: an opportunity for more sustainable urban freight transport?" European Transport Research Review **9**(3): 39.
Rao, S., E. Rabinovich and D. Raju (2014). "The role of physical distribution services as determinants of product returns in Internet retailing." Journal of Operations Management **32**(6): 295-312.
Roberti, R. and M. Wen (2016). "The electric traveling salesman problem with time windows." Transportation Research Part E: Logistics and Transportation Review **89**: 32-52.
Rodrigue, J. and L. Dablanc (2011). City logistics.
Rosenblat, A. (2018). Introduction: Using an App to Go to Work-Uber as a Symbol of the New Economy. Uberland, University of California Press: 1-20.

Rosenblat, A. and L. Stark (2015). "Uber’s drivers: Information asymmetries and control in dynamic work, workshop paper prepared for the Winter School “Labour in the on-demand economy” at the Centre for European Policy Studies (CEPS)." Brussels, Belgium: 23-25.
Rougès, J.-F. and B. Montreuil (2014). Crowdsourcing delivery: New interconnected business models to reinvent delivery. 1st international physical internet conference.
Ruan, M., J. Lin and K. Kawamura (2012). "Modeling urban commercial vehicle daily tour chaining." Transportation Research Part E: Logistics and Transportation Review **48**(6): 1169-1184.
Rześny-Cieplińska, J. and A. Szmelter-Jarosz (2019). "Assessment of the Crowd Logistics Solutions–The Stakeholders’ Analysis Approach." Sustainability **11**(19): 5361.
Sampaio, A., M. Savelsbergh, L. P. Veelenturf and T. Van Woensel (2018). The Benefits of Transfers in Crowdsourced Pickup-and-Delivery Systems, working paper Eindhoven University of Technology. http://www. optimization ⋯.
Serafini, S., M. Nigro, V. Gatta and E. Marcucci (2018). "Sustainable crowdshipping using public transport: A case study evaluation in Rome." Transportation research procedia **30**: 101-110.
Sheehan KB (2017). "Crowdsourcing research: Data Collection with Amazon's Mechanical Turk." Communication Monogram 0(0):1-17.
Simoni, M. D., E. Marcucci, V. Gatta and C. G. Claudel (2019). "Potential last-mile impacts of crowdshipping services: a simulation-based evaluation." Transportation: 1-22.
Sutton, R. S. and A. G. Barto (2018). Reinforcement learning: An introduction, MIT press.
Svartbäck, D. and A. G. Ekholm (2021). "Dynamic Pricing of Restaurant Food Delivery: An Explorative Market Efficiency Study." Available at SSRN 3809633.
Tang, C., J. Bai, K. So, X. Chen and H. Wang (2016). "Coordinating Supply and Demand on an On-Demand Platform: Price." Wage, and Payout Ratio.
Taylor, T. A. (2018). "On-demand service platforms." Manufacturing & Service Operations Management **20**(4): 704-720.
Thompson, R. G. (2015). Vehicle orientated initiatives for improving the environmental performance of urban freight systems. Green logistics and transportation, Springer**:** 119-129.
Tiwari, S., H. M. Wee, Y. Zhou and L. Tjoeng (2021). "Freight consolidation and containerization strategy under business as usual scenario & carbon tax regulation." Journal of Cleaner Production **279**: 123270.
Tong, T., H. Dai, Q. Xiao and N. Yan (2020). "Will dynamic pricing outperform? Theoretical analysis and empirical evidence from O2O on-demand food service market." International Journal of Production Economics **219**: 375-385.
Ursani, Z., D. Essam, D. Cornforth and R. Stocker (2011). "Localized genetic algorithm for vehicle routing problem with time windows." Applied Soft Computing **11**(8): 5375-5390.
Verbeek, M. (2008). A guide to modern econometrics, John Wiley & Sons.

Wang, Y., D. Zhang, Q. Liu, F. Shen and L. H. Lee (2016). "Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions." Transportation Research Part E: Logistics and Transportation Review **93**: 279-293.

Watkins, C. J. C. H. and P. Dayan (1992). "Q-learning." Machine Learning **8**(3): 279-292.

Xiao, Y. and A. Konak (2016). "The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion." Transportation Research Part E: Logistics and Transportation Review **88**: 146-166.

Yildiz, B. and M. Savelsbergh (2019). "Service and capacity planning in crowd-sourced delivery." Transportation Research Part C: Emerging Technologies **100**: 177-199.