Matching mechanisms for two-sided shared mobility systems

Jie Gao

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Information and Systems Engineering) at

Concordia University

Montréal, Québec, Canada

December 2021

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Jie Gao**

Entitled: **Matching mechanisms for two-sided shared mobility systems**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

____ Chair
Dr. Bruno Lee

____ External Examiner
Dr. Alfredo Núñez Vicencio

____ External to Program
Dr. Mingyuan Chen

____ Examiner
Dr. Jia. Y. Yu

____ Examiner
Dr. Jamal Bentahar

____ Supervisor
Dr. Chun Wang

Approved by ____
Dr. Mohammad Mannan, Graduate Program Director

December 08, 2021

____
Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Matching mechanisms for two-sided shared mobility systems

Jie Gao, Ph.D.

Concordia University, 2021

Shared mobility systems have gained significant attention in the last few decades due, in large part, to the rise of the service-based sharing economy. In this thesis, we study the matching mechanism design of two-sided shared mobility systems which include two distinct groups of users. Typical examples of such systems include ride-hailing platforms like Uber, ride-pooling platforms like Lyft Line, and community ride-sharing platforms like Zimride. These two-sided shared mobility systems can be modeled as two-sided markets, which need to be designed to efficiently allocate resources from the supply side of the market to the demand side of the market. Given its two-sided nature, the resource allocation problem in a two-sided market is essentially a matching problem.

The matching problems in two-sided markets present themselves in decentralized and dynamic environments. In a decentralized environment, participants from both sides possess asymmetric information and strategic behaviors. They may behave strategically to advance their own benefits rather than the system-level performance. Participants may also have their private matching preferences, which they may be reluctant to share due to privacy and ethical concerns. In addition, the dynamic nature of the shared mobility systems brings in contingencies to the matching problems in the forms of, for example, the uncertainty of customer demand and resource availability.

In this thesis, we propose matching mechanisms for shared mobility systems. Particularly, we address the challenges derived from the decentralized and dynamic environment of the two-sided shared mobility systems. The thesis is a compilation of four published or submitted journal papers. In these papers, we propose four matching mechanisms tackling various aspects of the matching mechanism design. We first present a price-based iterative double auction for dealing with asymmetric information between the two sides of the market and the strategic behaviors of self-interested agents. For settings where prices are predetermined by the market or cannot be changed frequently due to regulatory reasons, we propose a voting-based matching mechanism design. The mechanism is a distributed implementation of the simulated annealing meta-heuristic, which does not rely on a pricing scheme and preserves user privacy. In addition to decentralized matching mechanisms, we also propose dynamic matching mechanisms. Specifically, we propose a dispatch framework that integrates batched matching with data-driven proactive guidance for a Uber-like ride-hailing system to deal with the uncertainty of riders' demand. By considering both drivers' ride acceptance uncertainty and strategic behaviors, we finally propose a pricing mechanism that computes personalized payments for drivers to improve drivers' average acceptance rate in a ride-hailing system.

Acknowledgments

I take this opportunity to express my deepest thanks to all the people who had and have been helping me during the completion of my Ph.D. program. First and foremost, I would like to thank my advisor, Prof. Chun Wang, for his constant support and encouragement in all aspects of my research and academic career. Prof. Chun Wang has been not only a great advisor but also a cordial friend and a person mentor. He is always enthusiastic in sharing his experience and knowledge with me. He has taught me how to do deep research, to find new research directions by discovering variations and generalizations, to deliver presentations, and to be confident. Thank you so much for being so dedicated and involved in my work, while at the same time giving me a lot of freedom in pursuing my broad interests. Thank you for all the help, for having a supervision style based on trust, and for putting your faith in my proposals. It has been a great joy for the past five years being your student.

I also especially want to thank my thesis committee members, Prof. Jia Yuan Yu, Prof. Mingyuan Chen, Prof. Jamal Bentahar for their helpful and insightful comments, as well as the valuable questions which motivates me to widen my research from various perspectives. Special thanks go to Prof. Jia Yuan Yu for guiding me in finishing the first research project in my Ph.D. program. Thank you for taking the time to review my papers. Your valuable comments and suggestions helped a lot on this thesis and other research projects.

I would like to express my appreciation for my colleagues. This thesis has benefited from their comments and discussions with them. Thank you for your friendship and all your help.

Finally, I would like to express my sincere gratitude to my parents for their unconditional support and love from the beginning. Thank you for everything.

Contents

[1] Gao, J., Wong, T., Wang, C., Yu, J. Y. (2021). A Price-Based Iterative Double Auction for Charger Sharing Markets. IEEE Transactions on Intelligent Transportation Systems. (Early Access). DOI: 10.1109/TITS.2020.3047984

[2] Gao, J., Li, X., Wang, C., Huang, X. (2021). BM-DDPG: An Integrated Dispatching Framework for Ride-Hailing Systems. IEEE Transactions on Intelligent Transportation Systems. (Early Access). DOI: 10.1109/TITS.2021.3106243

List of Figures

List of Tables

Chapter 1

Introduction

The sharing economy has rapidly gained momentum across many industry sectors, such as transportation, hospitality, energy, financing, and human resources, which is essential in developing inclusive, collaborative, ecological and mobile next-generation cities. As defined by Daniel Schlagwein (Schlagwein, Schoder, & Spindeldreher, 2020), "The sharing economy refers to an IT-facilitated peer-to-peer model for commercial or noncommercial sharing of underutilized goods and service capacity through an intermediary without a transfer of ownership." Due to its social, economic and environmental benefits to the society, the sharing economy has maintained a strong growing trend. A study by PricewaterhouseCoopers (PwC) shows that the sharing economy is projected to grow from roughly $15 billion in 2014 to around $335 billion in 2025 (Hawksworth & Vaughan, 2014). Statistics Canada reports that the sharing economy has now become an annual $1.3 billion industry in Canada (Foundation, 2019).

As one segment of the sharing economy, *shared mobility* has gained popularity in recent years, serving as an innovative transportation strategy that enables users to gain short-term access to transportation modes on an "as-needed" basis (S. Shaheen, Cohen, Zohdy, et al., 2016). Since 2010, more than $330 billion has been invested in shared mobility systems (Holland-Letz, 2021). Shared mobility is an umbrella term that encompasses a variety of transportation modes, including car-sharing, bike-sharing, micro-transit, on-demand ride-hailing, ride-pooling, and community ride-sharing (Machado, de Salles Hue, Berssaneti, & Quintanilha, 2018), as shown in Figure 1.1. Compared with traditional transportation strategies, the use of shared mobility systems presents a number of environmental, social, and transportation-related benefits (Mourad, Puchinger, & Chu, 2019; S. Shaheen et al., 2019). For example, ride-sharing systems match drivers and riders with similar itineraries and time schedules, providing significant societal and environmental benefits by reducing the total miles traveled in the transportation network, the number of cars used for personal travel and improving the utilization of available seat capacity (Agatz, Erera, Savelsbergh, & Wang, 2012; Tafreshian, Masoud, & Yin, 2020). Recent studies of taxi trips in New York City (NYC) indicate that sharing trips could reduce traffic by 40% or more and could reduce the fleet size by 30% compared with the current transportation operation (Vazifeh, Santi, Resta, Strogatz, & Ratti, 2018). The Transportation Sustainability Research Center at UC Berkley (Nicoll & Armstrong, 2016) also indicates that ride-sharing could reduce carbon emissions between 34% and 41% per year for one household.

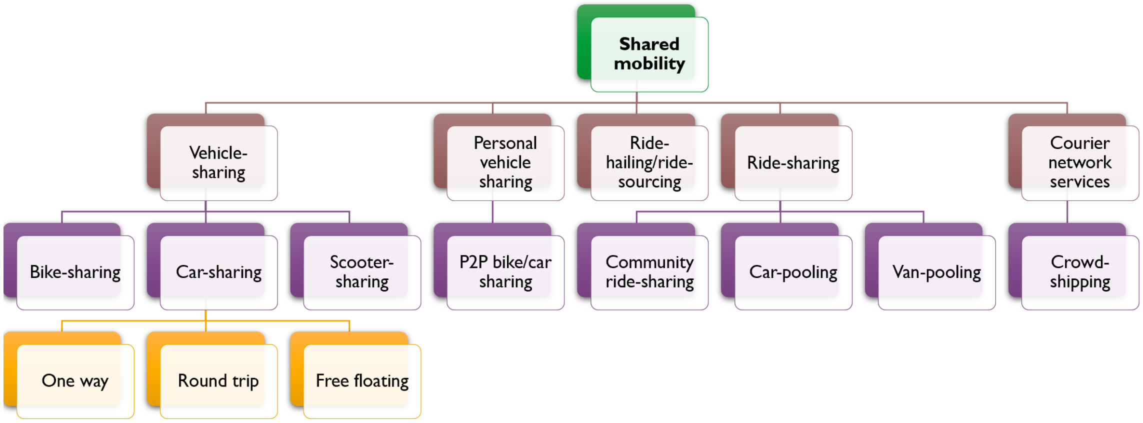


Figure 1.1: Key areas of shared mobility

In general, shared mobility systems can be classified into two categories: one-sided sharing systems and two-sided sharing systems, as shown in Figure 1.2. In a one-sided sharing system such as car-sharing and bike-sharing, the system operator has full control over the transportation resources and is responsible for allocating the resources to and coordinating the schedules of users who share the resources. All users are considered at one side of the market while the operator is at the other side. On the other hand, a two-sided sharing system contains two distinct groups of users, for example drivers and riders. One group provides shared resources to the other group. These two groups of users provide each other with network benefits. In such a system, the operator plays the role of match-maker who provides matching services for both sides of the market. Typical examples of two-sided sharing systems include ride-hailing and ride-sharing services provided by Uber, Lyft and DiDi Chuxing and courier network services such as crowd-shipping. Two-sided shared mobility systems can be modeled as *two-sided markets* where the system operator aims to match drivers from the supply side of the market with riders from the demand side of the market (Furuhata et al., 2013; Jia, Xu, & Liu, 2017; Rochet & Tirole, 2004).

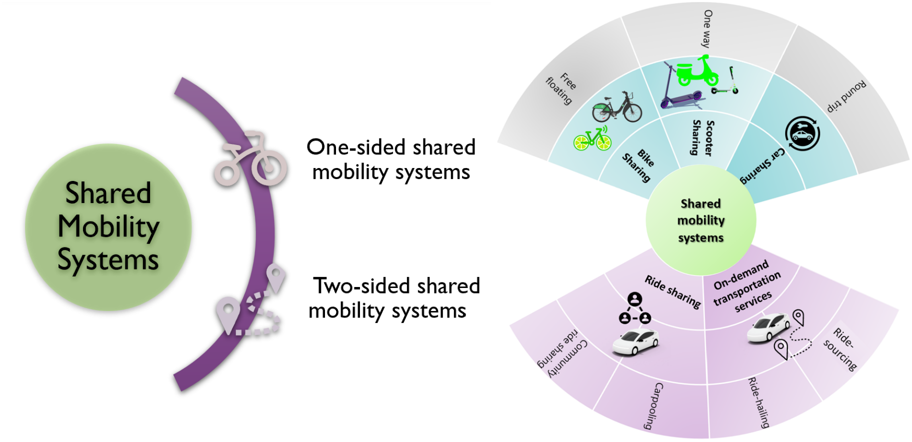


Figure 1.2: One-sided shared mobility systems and two-sided shared mobility systems

1.1 Matching in two-sided markets

Matching in two-sided markets is a decision making process which deals with allocating customer demand to service availability under imposed constraints from two sides of the market. This decision-making process is the core part of building a successful shared mobility system. To that end, a considerable amount of research has been devoted to the design and analysis of matching approaches for two-sided shared mobility systems. In a recent comprehensive review (H. Wang & Yang, 2019), matching approaches are classified into two categories: *greedy matching* and *batched-matching*. Greedy matching algorithms, such as those proposed in (G. Feng, Kong, & Wang, 2020; Lee, Wang, Cheu, & Teo, 2004), find the nearest driver or the shortest-travel-time driver for each individual rider request. Although these methods are easy to implement and manage, they are myopic in the sense that they prioritize immediate individual rider satisfaction over efficient resource utilization across many riders, which jeopardizes rider satisfaction at a larger scale. In addition, these matching methods do not explicitly address privacy concerns and strategic behaviors of the participants in the two-sided markets, which hinders their practical application to the real-world shared mobility systems. On the other hand, batched matching methods accommodate the needs of more riders at a time by optimizing the matching among a group of drivers and riders accumulated in a pre-determined batching window. However, existing batched matching methods used by ride-hailing platforms are mostly reactive in nature. For example, they use price surge strategies to guide idle drivers to high demand areas after the demands are realized, which, as shown in Chapter 5, does not reduce much waiting times of riders and introduce other adversarial effects such as surge chasing.

This dissertation aims to design matching mechanisms for computing high-quality matching solutions that benefit drivers, riders, and the shared mobility system operator. These matching mechanisms will improve the satisfaction of participants from both sides of the market. From the perspective of the systems, they will contribute to the improvement of the matching efficiency and user retaining rate, thus ensuring the sustainable growth of the systems.

1.2 Challenges of designing matching mechanisms for two-sided markets

One common challenge in matching mechanism design is the huge computation time demanded. This is due to the market thickness and the NP-hard nature of the considered matching problems (Pinedo, 2012). This challenge is inherited in the matching problems in two-sided markets. In addition to that, designing matching mechanisms for two-sided markets present two other challenges attributable to *decentralized* and *dynamic* market environment which are described below:

Decentralized Environment The challenge derived from the decentralized environment is to compute matching solutions in the presence of asymmetric information and self-interested agents in the market. Markets are naturally decentralized, in which matching related information is distributed and controlled by self-interested agents. There is no single fully-informed operator that is entitled to hierarchically allocate resources and determine the courses of action for the agents.

Agents from both sides of the market are motivated by different and conflicting objectives and have *private* preferences of matching with a counterparty. Due to privacy and ethical concerns, they may be reluctant to reveal their private information or may choose to reveal incomplete, and perhaps untruthful information about their preferences to the market if that leads to an individually beneficial outcome. Therefore, classic centralized matching approaches which assume that participants reveal complete and truthful matching information are impracticable under the market setting. As the winners of Nobel Economic Sciences Prizes, Paul R. Milgrom and Robert B. Wilson[1] pointed out: "Market design is difficult, because agents always act on their best response based on privately held information, their preferences and their beliefs about the outcomes of their actions." Thus how to compute high-quality matching solutions for shared mobility systems with asymmetric information and strategic behaviors of self-interested agents is challenging.

Dynamic Environment Matching problems in two-sided markets are highly dynamic as users from either side leave and enter the market over time, not all relevant information are known at the time the central operator executes a matching algorithm. Thus, in addition to the challenge arising from the decentralized market environment, matching has to be robust in accommodating the contingencies caused by the uncertainty of future demand and supply. Uncertainties regarding customer demand and service availability make the matching a complex dynamic process and impose significant challenges when service operators try to optimize the matching solution.

1.3 Outline of the dissertation

This dissertation aims to design matching mechanisms to deal with the challenges mentioned above. First, to deal with the **asymmetric information and strategic behaviors** of the self-interested agents in two-sided markets, in Chapter 3, we propose a price-based iterative double auction as a matching mechanism for allocating buyers to sellers in a charger sharing market. The auction computes social welfare maximizing matching solutions by bringing asking and bidding prices to a competitive equilibrium through iterative bidding without requiring buyers' and sellers' private information. This solution approach is rather general and we believe it can be applied to any

[1] https://www.nobelprize.org/prizes/economic-sciences/2020/press-release/

two-sided market with multiple buyers and sellers. The iterative double auction relies on a price discrimination strategy to find an optimal matching solution. While in some shared mobility systems, such as ride-sharing among colleagues, service price is pre-set and does not change on a daily basis. This demands non-price matching approaches. In Chapter 4, we propose a non-price voting based negotiation mechanism to compute near-optimal matching solutions for drivers and riders, while preserving their privacy. This mechanism implements an improvement-based searching process inspired by the simulated annealing meta-heuristic, allowing drivers and riders to negotiate matching solutions iteratively according to their individual preferences.

To deal with **customer demand uncertainties**, we propose a dispatching framework which integrates batched matching with data-driven proactive guidance in Chapter 5. Data-driven proactive guidance computes optimal open driver guidance strategies based on predicted rider demand and open driver supply for various regions. The demand prediction is generated through a machine learning algorithm based on historical data. The open driver supply is obtained from the matching solutions computed by batched matching during previous batching windows. The optimal guidance strategies are computed by solving a mixed integer program with the objectives of minimizing the open driver idle driving cost and the supply-demand gap of the region. Taking the computed guidance strategies from data-driven proactive guidance as inputs, batched matching computes the optimal bipartite matching among open drivers and waiting riders in the batch. By concurrently optimizing the guidance strategies and batched matching, the proposed framework possesses the ability to compute optimal matching results that maximize the social welfare of the two-sided market.

To deal with **supply availability uncertainties**, in Chapter 6, we propose a pricing mechanism to align the interests of the drivers and the platform, which leads to an overall better acceptance rate and platform profitability. This mechanism computes a set of payments tailored to each driver based on the unique characteristics of the assigned rides and the estimated acceptance probabilities of the drivers. We first build a binary choice model to estimate drivers' probabilities of accepting a ride based on the discrete choice analysis and the random utility theory. Taking the predicted acceptance probabilities as inputs, a stochastic optimization model is designed to compute payments for individual drivers. By leveraging behavioral economics and optimization techniques, the proposed pricing mechanism has the ability to compute optimal personalized payments that improve the overall ride acceptance rate and the platform profit. We foresee the proposed matching mechanisms can have considerable impact on many real world shared mobility systems, such as ride-hailing, ride-sharing and charger sharing platforms which require efficient and practical decentralized matching approaches. Finally, Chapter 7 summarizes the thesis and presents future research directions

Chapter 2

Matching problems in two-sided markets

This chapter introduces the definitions and characteristics of two-sided markets and the matching problems in the two-sided markets. At its core, we are trying to solve a matching problem, that is *Which buyer should be matched to which seller at what time?* Different from classical matching problems studied in the literature, the matching problem exists in a market environment, in which buyers and sellers are motivated by different and conflicting objectives. They have their private preferences and goals and will behave strategically in a market to advance their own benefits, rather than the system-wide efficiency. To recognize this autonomous nature of buyers and sellers, in microeconomics, they are modeled as *self-interested* agents (Mas-Colell, Whinston, Green, et al., 1995). In the following, we first introduce the definitions of a two-sided market and its unique characteristics. Then we define the matching problem in the context of two-sided markets.

2.1 Two-sided markets: definitions and characteristics

This section provides a brief introduction to definitions for two-sided markets.

2.1.1 General definitions

A two-sided market is an economic platform that enables direct interactions between two distinct customer groups that provide each other with network benefits. In 2003, Evans (Evans, 2003) defined a *two-sided market* as: “A two-sided market is a market in which a firm act as a platform: it sells two different products to two groups of consumers, while recognizing that the demand from one group of consumers depends on the demand from the other group and, possibly, vice versa. In other words, the demands on the two sides of the market are linked by indirect network effects, and the firm recognizes the existence of (that is, internalizes) these indirect network effects.” The *indirect network effect* means the value that a customer on one side realizes from the market increases with the number of customers on the other side. In their seminal paper, Rochet and Tirole (Rochet & Tirole, 2006) gave a more general definition for a two-sided market: “Two-sided (or more generally multi-sided) markets are roughly defined as markets in which one or several platforms enable interactions between end-users and try to get the two (or multiple) sides “on board” by appropriately charging each side. That is, platforms court each side while attempting to make, or at least not lose, money overall.” More definitions can be found in (Rysman, 2009). In line with these definitions, a two-sided market in this thesis is defined as:

Definition 1. *A two-sided market is defined as a platform that*

(1) has two distinct groups of customers;

(2) enables these two groups of customers to negotiate on the provision of the services; and

(3) has a significant indirect network effect across customer groups.

Here, the indirect network effect refers to the phenomenon that the value of the service increases for one customer group when a new customer of a different customer group joins the market (Rysman, 2009).

2.1.2 Market environment characteristics

After defining the two-sided market, in this subsection, we introduce the environment characteristics of the two-sided market: decentralized and dynamic.

Decentralized market environment

A two-sided market is a market, which is naturally decentralized. In decentralized market environments, the matching related information is distributed and controlled by self-interested agents. To maximize their own benefit or to preserve their own privacy, a rational self-interested agent may be reluctant to reveal his or her preference or may choose to reveal incomplete and perhaps untruthful information about his or her preference if that leads to an individually preferable outcome. In the context of this thesis, we specify the decentralized environment using a description from (C. Wang, 2007): a decentralized environment consists of *self-interested* agents with *private information* and objectives, and no agent has control of other agents. Thus, a decentralized market environment refers to a market with:

- *Asymmetric information* in which matching related information is distributed among agents.
- *Self-interested agent* with private information and game theoretic behavior.

Dynamic market environment

Two-sided markets are highly dynamic as participants from either side leave and enter the market over time. In a dynamic market environment, matching decisions are made over time and future demand and supply are unknown in advance to the service operator. We call it *dynamic* in the sense that the supply and demand of the market are *uncertain*. This is because new participants from both sides of the market continuously arrive; not all relevant offers and requests may be known when the service operator executes an algorithm for matching. Thus, a dynamic market environment refers to a market with *uncertainty of supply and demand*.

2.2 Matching problems in two-sided markets

After introducing the characteristics of the market environment, we model the matching problem in two-sided markets in terms of the following elements:

- **Agents**: An instance of a matching problem contains a service operator and two distinct groups of agents: buyers and sellers. Let B be the set of buyers. Let S be the set of sellers. Each buyer $b \in B$, from the demand side of the market has a service request needs to be served by a seller $s \in S$ who provide a service offer from the supply side of the market. We slightly abuse notations and use b and s to refer to both buyer and seller and his or her service request and service offer, respectively.

- **Constraints**: The constraints in the matching problem can be divided into two groups: matching-based constraints and time-based constraints. Matching-based constraints are matching restrictions related to agents' personal requirements, such as vehicle types. Time-based constraints are matching restrictions related to service time such as precedence constraints and service availability constraints. Specifically, precedence constraints define the order in which buyers can be served by sellers and service availability constraints define the time window during which an agent is available to be matched with a counterparty.

- **Matching solution**: The result of the matching problem is a matching solution which contains the matched buyer-seller pairs and the schedules (time and itinerary). A matching solution is *feasible* if it satisfies all the matching constraints. Let M be a set of feasible matching solutions.

- **Preference and utility function**: Each agent has a preference over a set of feasible schedules. According to the utility theory (Fishburn, 1970), this preference can be quantified by designing a *utility function*. The utility function evaluates an agent's satisfaction levels to a feasible matching solution. We assume agents have *quasi-linear utility functions* (Shoham & Leyton-Brown, 2008), such that the overall utility function of an agent can be formulated as a linear combination of two sub-utility functions, namely *matching utility function* and *scheduling utility function*. These two functions are designed for quantifying an agent's satisfaction levels to a matched counterparty and a travel schedule, respectively. The quasi-linear utility function of an agent takes the form of $u_a(m) = u_a^{match}(m) + u_a^{sched}(m), a \in B \cup S, m \in M$, where $u_a^{match}(m)$ is the matching utility function for agent a over m and $u_a^{sched}(m)$ is the scheduling utility function for agent a over m. The utility is the abstraction of what the agent desires and aims to maximize.

- **Measure of matching solution quality**: The quality of a matching solution m is measured by the total utilities of buyers and sellers, which is the *social welfare* of the market. Thus, the objective of the matching problem is to maximize the social welfare. Social welfare can be used to represent the overall benefits of market participants. We argue that social welfare maximization is a suitable objective for designing two-sided sharing markets as it focuses on the benefits of the whole society, which is well-aligned with the motivations of the sharing economy. In addition, social welfare is also considered as a measure of market efficiency in the microeconomics theory (Mas-Colell et al., 1995). Parkes and Kalagnanam (Parkes & Kalagnanam, 2005) suggest that market efficiency is well suited for the design of stable long-term markets. In sharing economies, buyers and sellers repeatedly rate each other after transactions, which provides strong motivation for both sellers and buyers to improve each others' benefits. By maximizing social welfare, the market will attract more transactions, allowing the charging platforms to take a small percentage of cut from sellers' and, sometimes, buyers' utilities as their revenue. Therefore, social welfare maximizing (or efficient) matching solutions are necessary for long-term revenue sustainability.

Chapter 3

A price-based iterative double auction for charger sharing markets[1]

The unprecedented growth of demand for charging electric vehicles (EVs) calls for novel expansion solutions to today's charging networks. Riding on the wave of the proliferation of sharing economy, Airbnb-like charger sharing markets open the opportunity to expand the existing charging networks without requiring costly and time-consuming infrastructure investments, yet the successful design of such markets relies on innovations at the interface between game theory, mechanism design, and large scale optimization. In this chapter, we propose a price-based iterative double auction for charger sharing markets where charger owners rent out their under-utilized chargers to the charge-needing EV drivers. Charger owners and EV drivers form a two-sided market which is cleared by a price-based double auction. Chargers' locations, availabilities, and unit time service costs as well as drivers' time and location preferences are considered in the allocation and scheduling process. The goal is to compute social welfare maximizing schedules which benefit both charger owners and EV drivers and, in turn, ensure the continuous growth of the market. We prove that the proposed double auction is budget balanced and individually rational. In addition, results from our computational study show that the proposed auction achieves on average 94% efficiency compared with that of the optimal solutions and is suitable for a larger day-ahead charger sharing market

[1] Gao, J., Wong, T., Wang, C., Yu, J. Y. (2021). A Price-Based Iterative Double Auction for Charger Sharing Markets. IEEE Transactions on Intelligent Transportation Systems. (Early Access). DOI: 10.1109/TITS.2020.3047984

setting in terms of running time.

3.1 Introduction

Energy spent on charging electric vehicles (EVs) will grow tremendously in the next decade. As estimated by the International Energy Agency, annual charging energy demand for the EV population is projected to increase from 58 billion kilowatt-hours to 640 billion kilowatt-hours from 2020 to 2030 (Agency, 2019). This surging demand places an unprecedented strain on existing charging networks which need to be substantially expanded in terms of the total amount of energy delivered and their geographical coverage. However, traditional methods to expand the charging networks such as building new charging stations and upgrading to high speed DC chargers are often costly and time-consuming (Hall & Lutsey, 2017; Howell et al., 2017; Schroeder & Traber, 2012). In recent years, *charger sharing* has emerged as one of the cost-effective and immediate solutions to expand the existing charging networks (Plenter et al., 2018; Vanrykel, Ernst, & Bourgeois, 2018). Online charger sharing platforms are being built to connect private charger owners and EV drivers. Some popular ones include PlugShare[2], EVMatch[3], ChargeHub[4] and Share&Charge[5]. Using such platforms, private charger owners aim to rent out under-utilized chargers to recoup their installation and operation costs and EV drivers procure the charging services to satisfy their energy needs.

The success of charger sharing platforms hinges on two major issues: (i) attracting both charger owners and EV drivers to the platform by providing added values to both of the groups, and (ii) retaining them by computing charging scheduling and pricing solutions which maximize the utilities of both EV drivers and charger owners. In the context of sharing economy, designing charger sharing platforms with the objective of benefiting all of its participants is essential to ensure their sustainable growth.

At the present time, the main scheduling mechanisms used by the charger sharing platforms are variants of the First-Come-First-Served (FCFS) mechanism with the "take-it-or-leave-it" pricing schemes. A FCFS mechanism schedules EV drivers in sequence according to their arrival times

[2] https://www.plugshare.com/

[3] https://www.evmatch.com/

[4] https://chargehub.com/en/

[5] https://shareandcharge.com/

to the market. While these mechanisms do motivate charger owners and EV drivers to participate if the price is right, they do not optimize the charging schedule across all market participants. Furthermore, a FCFS mechanism does not possess important economic properties which deal with game-theoretic behaviors of participants in a market. This allows the room for market speculation and further jeopardises the system wide performance.

Charger sharing scheduling mechanism design is a relatively new research topic and the literature on it is limited. As a more general research area, EV charging scheduling has attracted increased attention in the past years. Comprehensive reviews can be found in (Rahman, Vasant, Singh, Abdullah-Al-Wadud, & Adnan, 2016) and (Q. Wang, Liu, Du, & Kong, 2016). Several studies (Darabi, Fajri, & Ferdowsi, 2016; de Hoog, Alpcan, Brazil, Thomas, & Mareels, 2014; Q. Kang, Feng, Zhou, Ammari, & Sedraoui, 2017; Q. Kang, Wang, Zhou, & Ammari, 2015; Korkas, Baldi, Yuan, & Kosmatopoulos, 2017) have tackled EV charging scheduling problems by applying centralized approaches, which assume that a central scheduler is responsible for all allocation decisions. However, these centralized approaches cannot be applied to EV charging scheduling problems in the context of a charger sharing market. This market is naturally decentralized (Mas-Colell et al., 1995; Wellman, Walsh, Wurman, & MacKie-Mason, 2001), in which scheduling information are distributed and controlled by different self-interested agents.

Market based approaches, such as auctions, have gained popularity in providing socially desirable solutions to decentralized EV charging scheduling problems. These approaches respect autonomy and private information inherited from a distributed system and can provide incentives for agents to reveal truthful information (Parkes & Ungar, 2001a). For example, P. Samadi *et al.* (Samadi, Schober, & Wong, 2011) propose a Vickrey-Clarke-Groves (VCG) based mechanism for EV charging scheduling with the objective of maximizing the social welfare. J.de Hoog *et al.* (de Hoog, Alpcan, Brazil, Thomas, & Mareels, 2015) design a market mechanism for smart charging that optimally allocates available capacity and, at the same time, ensures network stability. However, these mechanisms address the setting of one-sided markets with one charger supplier, which cannot be directly applied to two-sided charger sharing markets. In (Gerding, Stein, Robu, Zhao, & Jennings, 2013), an EV charging scheduling problem is studied in a two-sided market setting. The authors propose a VCG payment rule to ensure truthfulness of EV drivers and charging stations.

Following the same payment rule, Yassin *et al.* (Yassine, Hossain, Muhammad, & Guizani, 2019) propose a double auction for energy trading of autonomous electric vehicles. Although the VCG mechanism is well known for being truthful and socially optimal, implementations of VCG-type mechanisms generally suffer from excessively high computational costs (Ausubel, Milgrom, et al., 2006) and are impractical for charger sharing markets with large numbers of charger owners and EV drivers.

Two-sided markets which involve two distinct groups of players, e.g., stock markets, are normally cleared by double auctions. In his seminal paper, McAfee (McAfee, 1992) proposes a trading reduction rule to achieve truthfulness in two-sided markets with homogeneous single unit goods. For the same problem, Chu and Shen (Chu & Shen, 2008) propose an agent competition mechanism by applying shadow prices to achieve strategy proof. More recently, some research has attempted to design double auctions for multi-unit heterogeneous trading problems. For example, Y. Chen *et al.* (Y. Chen, Zhang, Wu, & Zhang, 2013) extend McAfee's mechanism to multi-unit heterogeneous settings. They apply the proposed mechanism to spectrum allocation problems. In (Chichin, Vo, & Kowalczyk, 2016), a two-sided combinatorial greedy allocation mechanism is applied to multi-unit heterogeneous cloud exchange markets. In addition, iterative double auctions based on a decomposition scheme have been proposed for multi-unit heterogeneous energy trading environments (Deng, Yang, Hou, Chow, & Chen, 2014; Faqiry & Das, 2018; Iosifidis, Gao, Huang, & Tassiulas, 2015; J. Kang et al., 2017; Majumder, Faqiry, Das, & Pahwa, 2014). In these double auctions, trading goods are distinct, indivisible items. In order to apply these double auctions to the charger sharing scheduling problem, the continuous scheduling time line has to be discretized, such that the charging time period can be converted to a set of distinct time units (Kutanoglu & David Wu, 1999; Wellman et al., 2001). However, to maintain time accuracy, the discretized time unit cannot be too large. Therefore, this approach can generate a large number of distinct time units, which inflicts heavy computational burden on double auctions in terms of bids evaluation, communication, and winner determination (C. Wang, 2012; C. Wang, Ghenniwa, & Shen, 2009). In (C. Wang, Wang, Ghenniwa, & Shen, 2011) and (C. Wang & Dargahi, 2013), the authors use scheduling specific bidding language for decentralized scheduling problems, which models scheduling related constraints naturally and reduces computational costs. However, both papers focus only on one-sided settings where there is only one seller in the market.

In this chapter, we propose a price-based iterative double auction for matching charger owners (sellers) with EV drivers (buyers) in a charger sharing market with the objective of maximizing overall utilities of charger owners and EV drivers. In each round of the auction, sellers submit their offers to indicate their locations, available service times and prices per unit time they would like to charge (asking price). EV drivers (buyers) place bids to express their charging requirements and prices they would like to pay (bidding price) for using the services offered by the sellers. The sellers and buyers then update their asking and bidding prices as the auction proceeds into further rounds. The auction terminates when sellers and buyers no longer update their prices, which indicates the state that a market equilibrium has been reached. The proposed auction design possesses several economic properties, such as social welfare maximization, individual rationality, budget balance, which are desirable for such a two-sided market in the context of sharing economy. In Section 3.2, we will first introduce definitions and give more explanations of these properties. We then present the market environment of the auction based on which the iterative bidding structure are designed. In Section 3.3, we describe the proposed two-sided iterative bidding structure in details. A theoretical analysis on the properties of the auction is provided in Section 3.4, followed by a computational study in Section 3.5. We conclude our work and discuss future improvements in Section 3.6.

3.2 The auction and its environment

In this section, we first explain the concept of using auctions as a resource allocation mechanism in market settings with an emphasis on introducing desired auction properties. Then, we describe the charger sharing auction environment.

3.2.1 Auctions as a market for resource allocation

At its core, the problem we are trying to solve in this chapter is a resource scheduling problem, to which its solutions answer the question: *Which buyer should be allocated to which seller at what time?* Different from classical scheduling problems studied in the literature, the charger sharing scheduling problem exists in a market environment, in which buyers and sellers are motivated by different and conflicting objectives. Sellers want to maximize their revenues while buyers want to minimize their charging costs. They will behave strategically to advance their own benefits rather than the system wide performance. To recognize this autonomous nature of buyers and sellers, in the microeconomics literature, they are modeled as *self-interested* agents[6]. Scheduling problems that involve self-interested agents are called decentralized scheduling problems (Wellman et al., 2001).

Auctions are probably the most successful application of mechanism design theory to decentralized scheduling problems (Shoham & Leyton-Brown, 2008). They provide a level of abstraction based on which a system wide performance can be achieved through the simple interactions of bidding between individual agents and the auctioneer. In addition, auctions are natural and intuitive in terms of implementation in many real world domains. However, these benefits do not automatically accrue as a result of setting up an auction market. To make an auction market effective, certain economic properties need to be present. In the rest of this section, we will introduce these properties in the context of charger sharing markets. We start with two fundamental concepts based on which agents make decisions in an auction, namely *value* and *utility*.

- *Value*. In the context of charging services, value is the intrinsic worth of a service to the buyer, in other words, the highest price the buyer is willing to pay for the service. In this chapter, we follow the *private value model* as adopted by the classic theories of the Vickrey auction (Vickrey, 1961) and the VCG mechanism. In this model, a buyer has a value for each of the charging services defined by their available charging time and location. Note that a buyer can compute the driving distance to a seller based on his or her location. Therefore, the cost of driving to a location can be implicitly modeled in a driver's value on the location. These values do not depend on the private values of other buyers. Each buyer knows his or her own values but not the values of others. It is important to note that buyers may have different values over the same charging service offering. Different from their bidding prices, buyers' values do not change during the bidding process. Buyers are motivated to maximize their utilities not their values.

[6] Self-interest is an assumption of classical economic theory meaning that individuals are motivated in their actions by self interest. In The Wealth of Nations, Adam Smith makes the claim that, within the system of capitalism, an individual acting for his own good tends also to promote the good of his community. He attributed this principle to a social mechanism that he called the "invisible hand".

- *Utility*. The utility of a buyer for a charging service is the difference between the value of the service to the buyer and the price that the buyer needs to pay for it. For a seller, that is the difference between the price he or she can charge for and the cost of offering the service. The utility is the abstraction of what the agent desires and aims to maximize. Both buyers and sellers want to maximize their utilities in market transactions.

- *Social welfare maximization*. Social welfare is defined as the aggregated utilities of all agents. It can be used to represent the overall benefits of market participants. We argue that social welfare maximization is a suitable objective for designing charger sharing markets as it focuses on the benefits of the whole society, which is well aligned with the motivations of the sharing economy. In addition, social welfare is also considered as a measure of *market efficiency* in the microeconomics theory (Mas-Colell et al., 1995). Parkes and Kalagnanam (Parkes & Kalagnanam, 2005) suggest that market efficiency is well suited for the design of stable long-term markets. In sharing economies, buyers and sellers repeatedly rate each other after transactions, which provide strong motivation for both sellers and buyers to improve each others' benefits. Given the definition of utility above, the aggregated utilities of all agents (social welfare) in our charger sharing market is the difference between the sum of buyers' values and the sum of sellers' costs calculated on a charging schedule. The same definition of social welfare is also adopted in other double auctions proposed in the literature (Deng et al., 2014; Faqiry & Das, 2018; Iosifidis et al., 2015; J. Kang et al., 2017). It is also worth mentioning that social welfare maximization is different from revenue maximization. The idea here is that, by maximizing social welfare, the market will attract more transactions, which allows the charging platforms better opportunity to take a small percentage of cut from sellers' and, sometimes, buyers' utilities as their revenue. Therefore, social welfare maximizing (or efficient) auctions are necessary for long term revenue sustainability.

- *Budget balance.* Budget balance implies that the total price that the platform charges all winning buyers is not less than the total payment that the platform rewards all winning sellers, so that the platform neither accumulates surplus, nor runs in a deficit and does not need to be subsidised by outside payments. In a practical context, such as the charger sharing market, this property can also allow the platform to make profits as stated later in Remark 2 of Section 3.4.

- *Individual rationality*. Individual rationality ensures that all participants in the market should obtain non-negative utility, so that agents have incentives to take part in the auction, which is important in terms of attracting charger owners and EV drivers to join the market.

Assuming that maximizing social welfare is the objective, then the task of the auction is to compute an optimal schedule given the sellers' costs and buyers' values on available charging services. However, buyers and sellers are not willing to reveal their values and costs due to their conflicting objectives. To maximize their utilities, buyers always want to pay less than their values and sellers always want to charge more than their costs. In terms of game theory, buyers' values and sellers' costs are considered as their *private information* which will not be revealed to the auctioneer. The challenge for the auction now is how to design the bidding and pricing rules such that when the market reaches an equilibrium meaning that the supply meets the demand given a price vector on services, the resulting schedule maximizes the social welfare. Since the charger sharing market is a two-sided market, in our setting, we design the bidding rules which are similar to the double auctions used in the stock markets. The bidding rules dictate that buyers gradually move their bidding prices up and sellers gradually move their asking prices down so that the equilibrium prices can be reached at which transactions occur.

Since the charger sharing market is usually a large system in which the effect of an agent's own strategy on the state of a market is small, we adopt the competitive market condition by assuming agents' behavior as *price-taking* or *myopic best-response* Parkes and Ungar (2001b). Under this assumption, an agent plays a best-response to the current price and allocation conditions in the market without modeling either the strategies of other agents or the effect of its own actions on the future state of the market. A double auction can be defined by its market environment and bidding rules. We will describe the environment of the double auction in the following section and the iterative bidding rules in Section IV.

Table 3.1: Summary of notations

Notation	Description
$\mathcal{M}$	The set of sellers, indexed by m
$\mathcal{N}$	The set of buyers, indexed by n
$[s_m, e_m]$	Available time window of seller m
c_m	Unit time service cost of seller m
$[a_{n,m}, d_{n,m}]$	Feasible time window of buyer n for charging at seller m
$r_{n,m}$	Required charging duration of buyer n for charging at seller m
$v_{n,m}$	Valuation of buyer n for charging at seller m
K	An overall charging schedule
$k_{n,m}$	Decision variable: starting time of buyer n at seller m $k_{n,m} \in \mathbb{R}^+$ if n is allocated to m, and $k_{n,m} = -1$ otherwise
$\mathbb{1}_{k_{n,m}}$	Indicator function: $\mathbb{1}_{k_{n,m} \neq -1} = 1$ if $k_{n,m} \neq -1$ is true
t	A biding round number
α^t_m	Ask of seller m at round t
$\theta^t_{n,m}$	Bid of buyer n for seller m at round t
$p^{s,t}_m$	Asking price of seller m at round t for a unit time of service
$p^{b,t}_{n,m}$	Bidding price of buyer n for seller m at round t for a unit time of service
ϵ	Increment for buyers and decrement for sellers

3.2.2 The charger sharing double auction environment

The charger sharing double auction consists of a group of sellers, a group of buyers and an auctioneer (called the platform thereafter). Each seller offers a charging service characterized by his or her location and available time window for charging. For each unit time of charging service offered, the seller incurs a fixed cost which can be a combination of the cost of electricity consumed and the proportionate cost related to maintaining the parking space. Buyers have constraints regarding when and where they can charge. For those sellers who can accommodate their constraints, buyers will calculate values over their services based on how far the chargers are located and how convenient their available time windows are. The goal of the platform is to allocate buyers to sellers, such that their charging constraints are satisfied and the social welfare of all buyers and sellers is maximized.

Let $\mathcal{M}$ be the set of sellers and $[s_m, e_m]$ be the available charging time window of seller $m \in \mathcal{M}$, during which the seller is available for providing his or her charging service to the buyers. The cost for seller m to offer a unit time of charging service is denoted as $c_m \in \mathbb{R}^+$. Let $\mathcal{N}$ be the set of buyers. For each buyer $n \in \mathcal{N}$, let $a_{n,m}$ be his or her earliest arrival time to seller m and $d_{n,m}$ be his or her latest departure time from seller m. $[a_{n,m}, d_{n,m}]$ defines the feasible time window during which buyer n has to complete his or her charging. Note that a buyer may have different feasible charging time windows for different sellers depending on the current location of the buyer. For buyer m to be allocated at seller n, $[a_{n,m}, d_{n,m}]$ must be within $[s_m, e_m]$. That is $a_{n,m} \geq s_m$

and $d_{n,m} \leq e_m$. Let $r_{n,m} \in \mathbb{R}^+$ denote the required charging duration of buyer n at seller m with $r_{n,m} + a_{n,m} \leq d_{n,m}$.

We assume that a charging service is non-preemptive, i.e., once buyer n has started charging at seller m, the service cannot be interrupted. Let $v_{n,m} \in \mathbb{R}^+$ be the value of buyer n for charging at seller m. The result of the charger sharing double auction is a *schedule* which can be represented by a matrix $K \in \mathbb{R}^{|\mathcal{M}|\times|\mathcal{N}|}$, where each element $k_{n,m}$ denotes the starting time of buyer $n \in \mathcal{N}$ at seller $m \in \mathcal{M}$. Let $k_{n,m} \in \mathbb{R}^+$ if n is allocated to m, and $k_{n,m} = -1$ otherwise. For a schedule K to be *feasible*, each $k_{n,m}$ has to satisfy the following constraints:

(i) A buyer can only start charging after his or her arriving time, i.e., $k_{n,m} \geq a_{n,m}$, $\forall n \in \mathcal{N}$, $\forall m \in \mathcal{M}$,

(ii) A buyer has to finish charging before his or her departure time, i.e., $k_{n,m} \leq d_{n,m} - r_{n,m}$,$\forall n \in \mathcal{N}$, $\forall m \in \mathcal{M}$,

(iii) A buyer can only be allocated to one seller, i.e., $\forall m, m' \in \mathcal{M}$: if $k_{n,m} \neq -1$ and $k_{n,m'} \neq -1$ then $m = m'$, $\forall n \in \mathcal{N}$,

(iv) If two buyers n and n' are allocated to the same seller, either n must be finished before the start time of n' or n' must be finished before the start time of n, i.e., $\forall n, n' \in \mathcal{N}$: if $k_{n,m} \neq -1$ and $k_{n',m} \neq -1$, then $k_{n,m} + r_{n,m} \leq k_{n',m} + H \cdot (1 - Y_{n,n',m})$ and $k_{n',m} + r_{n',m} \leq k_{n,m} + H \cdot Y_{n,n',m}$[7],

(v) If a buyer is allocated to a seller, the charging time should be within the seller's available time window, i.e., $s_m \leq k_{n,m} \leq e_m - r_{n,m}$, $\forall n \in \mathcal{N}$, $\forall m \in \mathcal{M}$,

(vi) If a buyer is allocated to a seller, the buyer's value cannot be less than the cost of the seller, i.e., if $k_{n,m} \neq -1$, then $v_{n,m} \geq r_{n,m} \cdot c_m$, $\forall n \in \mathcal{N}$, $\forall m \in \mathcal{M}$.

Let $\mathbb{1}_{k_{n,m} \neq -1}$ be the indicator function that equals 1 if $k_{n,m} \neq -1$ is true and 0 if otherwise. Let $p_{n,m}$ be the unit price paid by the buyer n for charging at seller m. Then the utility of buyer n

[7] $Y_{n,n',m}$ is the disjunctive variable: $Y_{n,n',m} = 1$ when n is scheduled before n' on m and $Y_{n,n',m} = 0$ n' is first. H is a large positive constant which is used for the linearization of the logical constraint "if" (Rardin & Rardin, 1998).

is defined as:

$$u_n^b = \sum_{m \in \mathcal{M}} \mathbb{1}_{k_{n,m} \neq -1} \cdot (v_{n,m} - p_{n,m} \cdot r_{n,m}) \tag{1}$$

and for a seller m as:

$$u_m^s = \sum_{n \in \mathcal{N}} \mathbb{1}_{k_{n,m} \neq -1} \cdot (p_{n,m} \cdot r_{n,m} - c_m \cdot r_{n,m}) \tag{2}$$

The objective of the platform is to compute a feasible schedule K that maximizes social welfare, which is formulated as:

$$\begin{aligned} \max \sum_{n \in \mathcal{N}} u_n^b + \sum_{m \in \mathcal{M}} u_m^s = \\ \max \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} (v_{n,m} - c_m \cdot r_{n,m}) \cdot \mathbb{1}_{k_{n,m} \neq -1} \end{aligned} \tag{3}$$

However, as discussed in Section 3.2.1, buyers and sellers are modeled as self-interested agents in the market setting. They cannot be assumed to reveal their values and costs to the platform due to their conflicting objectives. In the next section, we will present a priced-based iterative double auction (P-IDA) as a scheduling mechanism for allocating buyers to sellers. The auction computes social welfare maximizing schedules by bringing asking and bidding prices to a competitive equilibrium through iterative biding without requiring buyers' values and sellers' costs. The general structure of the auction can be applied to other sharing markets where multiple buyers and sellers are involved. The charger sharing market environment can be used to demonstrate the implementation and the performance of the proposed auction.

3.3 Price-based Iterative Double Auction

In this section, we present a price-based iterative double auction (P-IDA) for the aforementioned charger sharing auction environment. The auction serves as an iterative negotiation process between sellers and buyers. The process is mediated by the platform which matches buyers' charging requests to sellers' service offerings. In double auction literature and real world double auction markets, such as stock exchanges, a seller's offering is usually called an *ask* and a buyer's request a *bid*. In our charger sharing auction, an ask of seller m, denoted by α_m, is a three tuple (s_m, e_m, p^s_m), where p^s_m indicates the asking price for *a unit time* of service. A buyer n's bid on seller m's service offering, denoted by $\theta_{n,m}$, is also represented as a four tuple $(a_{n,m}, d_{n,m}, r_{n,m}, p^b_{n,m})$, where $p^b_{n,m}$ is the bidding price of buyer n on *a unit time* of charging service offered by seller m.

3.3.1 Iterative bidding procedure

Before the auction starts, for each of the registered sellers, the platform announces a price range for his or her service offering. The purpose for setting up such a range is to speed up the convergence to price equilibrium by preventing sellers posting unnecessarily high asking prices and buyers submitting arbitrarily low biding prices. The price range for a seller is the average of historical transaction prices at his or her location plus-minus a certain percentage set by the platform. The platform will not accept an asking price higher than the upper bound of the range and a biding price lower than the lower bound of the range. A high level description of P-IDA is given in Algorithm 1. Before the iterative bidding starts, the sellers set the upper bounds of the price ranges as their initial asking prices and submit their asks to the platform. Buyers set the lower bounds of the price ranges as their initial bidding prices for the sellers' service offerings (Lines 2-3 in Algorithm 1). The bidding then proceeds in rounds. Each round contains three steps, namely price updating, termination checking and winner determination. In the following, we will first give a general description of the three steps which is applicable to any round $t(t > 1)$. We then describe round $t = 1$ as a special case of the general bidding procedure.

Price updating

At round $t(t > 1)$, the buyers and sellers update their bidding and asking prices based on the provisional schedule K^{t-1} computed by the winner determination step at round $t - 1$ (Lines 6-19). If a buyer is not included in K^{t-1} i.e. rejected at round $t - 1$, he or she has two price updating options at round t:

Algorithm 1 Price-based iterative double auction

```
Input: $w (0 \leq w \leq 1)$, $\epsilon$, $\forall m \in \mathcal{M}, \forall n \in \mathcal{N}$
Output: $K^t$
$t \leftarrow 1$;                                                      ▷ round index
Each seller $m \in \mathcal{M}$ initialize its asking price and ask;
Each buyer $n \in \mathcal{N}$ initialize its bidding prices;
$ter.flag = 0$;
while $ter.flag = 0$ do                                                 ▷ iterative bidding starts
    for all $n \in \mathcal{N}$ do                                      ▷ price updating for buyers
        if $t > 1 \wedge$ $n$ is not included in $K^{t-1}$ then
            for all bids submitted at round $t - 1$ do
                $p^t_{n,m} = p^{t-1}_{n,m} + w \cdot \epsilon$;
            end for
        end if
        end
        Compute utility-maximization bids by (4);
        Submit utility-maximization bids to the platform;
    end for
    end
    for all $m \in \mathcal{M}$ do                                      ▷ price updating for sellers
        if $t > 1 \wedge$ $m$ has available time in round $t - 1$ then
            $p^t_m = p^{t-1}_m - w \cdot \epsilon$;
        end if
        end
        Send the ask to the platform;
    end for
    end                                                                 ▷ termination checking
     The platform checks the termination condition:
    if termination condition is satisfied then
        $ter.flag \leftarrow 1$;
    end if
    end                                                                 ▷ winner determination
    The platform computes $K^t$ by (5);
    Send $K^t$ to buyers and sellers;
    $t \leftarrow t + 1$;
end while
end
The platform collects payment $p^{b,t}_{n,m}$ from each buyer and reimburse to the corresponding sellers;
```

- The buyer can increase his or her bidding price by ϵ on the rejected bid. Here ϵ is the minimum bid-increment or ask-decrement predetermined by the auction. Buyers do not bid with an increment more than ϵ as they are assumed to be rational in maximizing their utilities.
- The buyer can keep his or her bidding price unchanged or make an increment less than ϵ on the bid rejected at round $t-1$. This happens when the utilities of this buyer for all other sellers are non-positive given his or her current bidding prices. In this case, the platform will consider that this buyer has entered into the final bid status and the buyer is forbidden from increasing the bidding prices on any of his or her bids in the future rounds.

If a buyer is included in the provisional schedule K^{t-1}, this buyer will keep his or her bidding price unchanged at round t. After updating their bidding prices, buyers compute their utility maximizing bids to be submitted in the current round t based on their updated bidding prices. The submitted bid in round t may be different of that in round $t-1$. In computing such bids $m \in \mathcal{M}$, a buyer $n \in \mathcal{N}$ solves the following utility maximization problem:

$$\max_{m \in \mathcal{M}} v_{n,m} - r_{n,m} \cdot p_{n,m}^{b,t} \tag{4}$$

where $v_{n,m}$ is fixed, representing the highest price that buyer n is willing to pay for charging at seller m and $p_{n,m}^{b,t}$ is the bidding price of buyer n for charging at seller m at round t. Note that the maximization problem may have multiple solutions which equally maximize buyer n's utility. In this case, depending on the bidding rule configuration of the platform, the buyer can randomly pick one (Atomic-Bid bidding rule) or join them together as an XOR-bid (XOR-Bid bidding rule) and submits it to the platform. Here an XOR-bid:

$$\theta_{n,1}^t \oplus \theta_{n,2}^t \oplus \cdots \oplus \theta_{n,m}^t$$

indicates that:

- these bids equally maximize buyer n's utility at round t based on the bidding prices and his or her values,
- at most one $\theta_{n,k}^t$ can be included in the current round provisional schedule.

Note that the buyers included in K^{t-1} will not update their bidding prices. Therefore, their utility maximizing bids to be submitted to the platform will be the same as those submitted at round $t-1$. In other words, they will repeat their bids from the previous round. Moreover, the platform can also allow the buyers who have already entered their final bid status to repeat their XOR-Bid (XOR-Bid-Repeating rule).

Meanwhile, sellers also need to update their asking prices at the beginning of round t. If all available time units of a seller are included in the provisional schedule K^{t-1} , which means this seller has sold all of his or her available charging time at round $t-1$, the seller repeats his or her ask at round t. On the other hand, if a seller still has unsold charging time at round $t-1$, this seller has the following two asking price updating options in round t:

- The seller can decrease his or her asking price by ϵ on the ask submitted at round $t-1$. As sellers are assumed to be rational in maximizing their utilities, they do not bid with a decrement more than ϵ.
- The seller can keep the asking price unchanged or make a decrement less than ϵ. This happens when $p_m^{s,t-1} = c_m$ or $p_m^{s,t-1} - c_m < \epsilon$. In this case, the seller is not allowed to decrease his or her asking prices in the future rounds.

After updating their asking prices, sellers submit their updated asks to the platform.

Termination checking

Upon receiving the updated bids and asks from buyers and sellers, the platform then checks the asks and bids against the termination condition (Lines 20-21 of Algorithm 1) stated as follows:

> *Termination condition: all buyers and sellers submit the same bids and asks in two consecutive rounds.*

If the termination condition is satisfied, which indicates that asking and bidding prices will not change in future rounds and a market equilibrium has been reached, the auction terminates and the provisional schedule K^{t-1} becomes the final schedule. An awarded buyer pays his or her charging time in K^{t-1} at his or her current bidding price. A seller gets paid by all buyers allocated to him or her in K^{t-1}. The amount is the sum of the allocated buyers' payments. On the other hand, if the termination condition is not satisfied, the platform proceeds to winner determination.

Winner determination

In this step, the platform computes a new provisional schedule K^t taking the updated bids and asks from sellers and buyers as inputs. The winner determination model at round t is formulated as:

$$\max \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} r_{n,m} \cdot (p^{b,t}_{n,m} - p^{s,t}_{m}) \cdot \mathbb{1}_{k^t_{n,m} \neq -1} \tag{5}$$

subject to the feasible schedule constraints same as described in Constraints (i)-(v). In order to ensure that the bidding price of a buyer is equal or higher than the asking price of the charging time allocated to the buyer, an additional constraint is added: $\forall n, m$: if $k^t_{n,m} \neq -1$, then $p^{b,t}_{n,m} \geq p^{s,t}_{m}$. As more than one optimal solutions may exist, ties are broken in favour of first maximizing the number of buyers and sellers in K^t and then at random.

During the first round of the auction ($t = 1$), since there is no provisional schedule computed from the previous round, as a special case to the general bidding procedure, buyers and sellers do not need to update their bidding and asking prices. Buyers just compute their utility maximizing bids based on their initial bidding prices on the sellers. For the same reason, termination checking is also not needed during the first round. Winner determination just takes the initial bids and asks as inputs to compute the first provisional schedule K^1.

3.3.2 Approximate winner determination

The winner determination model can be solved using commercial integer programming optimization packages, such as IBM CPLEX. However, since the model is NP-hard, it is not practical to compute optimal provisional schedules for larger scale charger sharing market in a timely manner. In order to improve the practicality of the proposed auction, we design a meta-heuristic winner determination algorithm based on simulated annealing (SA) (Kirkpatrick, Gelatt, & Vecchi, 1983). Due to space limitations, only a brief description is presented here. The algorithm starts by randomly generating an initial schedule based on the bids and asks submitted by buyers and sellers, then it proceeds by taking the form of iterative "cooling" process. At each cooling cycle, multiple schedules are generated by a random permutation of the current incumbent schedule. Then a subset of the generated schedules are selected based on the values of their winner determination functions. A generated schedule is always selected if it has a superior value than that of the incumbent schedule. Otherwise a schedule is selected with a given decreasing possibility based on Boltzmann distribution (Kirkpatrick et al., 1983). The size of the selected subset decreases along the cooling process. The process converges at the cycle when the size of the selected subset reduces to one. This last selected schedule then becomes the provisional schedule computed by winner determination at current bidding round. It is worth noting that the proposed iterative bidding auction equipped with this approximate winner determination algorithm maintains the same incentive for buyers and sellers to follow myopic best-response bidding strategies in larger market settings (Parkes & Ungar, 2000).

3.3.3 A worked example

This section presents a worked example of the iterative double auction procedure. This example contains two sellers and three buyers. We intentionally keep the example oversimplified for the purpose of clearly illustrating the steps of the bidding process. The available time windows and the costs of a time unit for seller 1 and seller 2 and the feasible time windows, required charging duration and valuations of the three buyers are shown in the third row of TABLE 3.2. As shown in this table, buyer 1 can be charged at seller 1 and seller 2, while buyer 2 can only be charged at seller 1 and buyer 3 can only be charged at seller 2. Assume that the upper bound of the asking price is \$5 an hour, the lower bound of the bidding price is \$3 an hour and the minimum bid-increment or ask-decrement $\epsilon = \$1$. For the sake of simplicity, we assume that sellers use \$5 as their initial asking prices and buyers use \$3 as their initial bidding prices. The asking and bidding prices and allocation of each round of the iterative bidding are shown in TABLE 3.2. The iterative bidding proceeds as follows.

1) Round 1: Buyer 1 submits the utility maximizing bid $\theta_{1,2}$ to the auctioneer and buyer 2 and buyer 3 submit bids $\theta_{2,1}$ and $\theta_{3,2}$ to the auctioneer, respectively. At the same time, seller 1 and seller 2 submit asks α_1 and α_2, respectively to the auctioneer. After solving the winner determination

Table 3.2: Bidding process of a two-sided iterative auction example. The utility maximizing bids are in **bold**. The allocated bids are indicated by *.

	Asking and Bidding prices						Provisional schedule
	Seller1	Seller2	Buyer1		Buyer2	Buyer3	
	[18:00,22:00], $3	[16:00,20:00], $3	[18:00,19:00],1h,$4.5	[17:00,19:00],1h,$5	[19:00,22:00],2h,$6	[17:00,18:00],1h,$4	
Round	α_1	α_2	$\theta_{1,1}$	$\theta_{1,2}$	$\theta_{2,1}$	$\theta_{3,2}$	
1	$5	$5	$3	$**3**	$**3**	$**3**	
2	$4	$4	$**3**	$4	*$**4**	*$**4**	B2-S1; B3-S2
3	$3	$3	$4	*$**4**	*$**4**	*$**4**	B1-S2; B2-S1; B3-S2
4	$3	$3	$4	*$**4**	*$**4**	*$**4**	B1-S2; B2-S1; B3-S2

model, there is no trade between buyers and sellers resulted in the first round.

2) Round 2: Sellers decrease their asking prices to $4. Accordingly, buyer 2 and buyer 3 increase their bidding prices to $4. Buyer 1 increases its bidding price on seller 2 to $4. Given the updated bidding prices, buyer 1 bids on seller 1 at this round because this bid maximize his or her utility. The auctioneer allocates buyer 2 to seller 1 and buyer 3 to seller 2 in the provisional schedule.

3) Round 3: Seller 1 decreases its asking price to $3 and seller 2 decreases its asking price to $3. Buyer 2 and buyer 3 in this round repeat their bids as they were included in the provisional schedule in Round 3. Buyer 1 increases its bidding price on seller 1 to $4 and bid on seller 2 as $\theta_{1,2}$ maximizes his or her utility in this round. The auctioneer allocates buyer 1 to seller 2 in this round.

4) Round 4: Both sellers and buyers repeat their asks and bids. The iterative bidding terminates with an optimal schedule.

3.3.4 A day-ahead charger sharing market implementation

We now describe an implementation scenario of the proposed P-IDA in a day-ahead charger sharing market. Day-ahead markets are commonly used for trading electric energy, in which market participants commit to buy or sell electricity one day before the operating day. While electricity markets are general commodity markets in which buyers do not distinguish the supply from different sellers, our charger sharing market accommodates buyers' preferences over sellers and requires social welfare maximizing matching between buyers and sellers. In this charger sharing market, buyers can book their charging services for an operating day through participating in the auction which terminates before the operating day starts. To spare buyers and sellers the trouble of continuously monitoring the auction process and repeatedly placing their asks and bids, the auction can be implemented as an autonomous multi-agent system, in which buyers and and sellers are represented by their software proxy agents which place asks and bids on behalf of them. A buyer's preferences can be programmed into his or her proxy agent or learned from previous bidding data. A seller can also configure his or her proxy agent with the cost and availability information before the auction starts. In the meantime, the agents should be equipped with the algorithm to update asking and bidding prices and select the utility maximization bids along the bidding process. For easy access, buyers and sellers may install their proxy agents on a personal computer, a smart phone, or other mobile devices.

While we call it "day-ahead" market, the operating time interval is not necessarily a day, it can be half a day or even an hour. For shorter operating intervals, a fast winner determination algorithm is required. We will elaborate more on the running time performance of various algorithms in Section 3.5.

3.4 Theoretical analysis

In this section, we analyze the economic properties of P-IDA. As discussed in Section II, three properties are of importance in terms of setting up an effective charger sharing auction, namely social welfare maximization, budget balance and individual rationality. We will evaluate the auction's performance on social welfare maximization in the next section through a computational study. We focus on budget balance and individual rationality in the rest of this section.

If an auction is budget balanced, it requires no outside subsidy to ensure social welfare maximizing allocations. Given the proposed iterative double auction design, we have the following remarks regarding its budget balance property:

Remark 1. *P-IDA is budget-balanced because the total payments collected from the buyers is equal to the total reward assigned to the sellers.*

Remark 2. *If the platform take a percentage of cut from buyers' payments, P-IDA can be considered weakly budget-balanced in the sense that the total payments collected from the buyers is equal to the total reward assigned to the sellers and the platform.*

Individual rationality is another necessary property to sustain the continuous growth of a charger sharing platform by providing incentives for both buyers and sellers to participate in the market. It implies that participants are never worse off by participating in the auction. As stated in the following theorem, the proposed auction possesses this property.

Theorem 1. *P-IDA is individually rational for all participating buyers and sellers.*

Proof. This theorem will be established separately for the buyers and the sellers. Note that not participating in the auction leads to a zero utility for any buyers and sellers as they will not receive or offer any charging services nor they will pay any kind of fee. For an arbitrary buyer $n \in \mathcal{N}$, upon termination of the auction, if he or she is not included in the final schedule, then the utility of buyer n is zero: $u_n^b = 0$ (See Equation (1)).

If on the other hand, buyer $n \in \mathcal{N}$ is allocated to an arbitrary seller $m \in \mathcal{M}$ when the auction terminates at time t^T, then the utility of buyer n is $v_{n,m} - r_{n,m} \cdot p_{n,m}^{b,t^T}$, where $p_{n,m}^{b,t^T}$ is buyer n's last round bidding price. Since $v_{n,m} \geq r_{n,m} \cdot p_{n,m}^{b,t^T}$ always holds, it follows that: $u_n^b = v_{n,m} - r_{n,m} \cdot p_{n,m}^{b,t^T} \geq 0$. Therefore, it is concluded that P-IDA is individually rational for all participating buyers.

If an arbitrary seller m is not included in the final schedule, then $u_m^s = 0$ (See Equation (2)). If, on the other hand, seller m is included in the final schedule, to satisfy the feasible schedule constraint (vi), we have $p_{n,m}^{b,t^T} - p_m^{s,t^T} \geq 0$ always holds, where p_m^{s,t^T} is seller m's asking price in the last round. Since $p_m^{s,t^T} \geq c_m$, it follows that: $p_{n,m}^{b,t^T} - c_m \geq p_{n,m}^{b,t^T} - p_m^{s,t^T} \geq 0$. Therefore, $u_m^s = \sum_{n \in \mathcal{N}} (p_{n,m}^{b,t^T} - c_m) \cdot r_{n,m} \geq p_{n,m}^{b,t^T} - c_m \geq 0$. Thus we conclude that P-IDA is individually rational for all participating sellers. □

Iterative bidding is a type of heuristic search implementation of VCG. Compared with VCG mechanisms (de Hoog et al., 2015; Gerding et al., 2013; Samadi et al., 2011), the iterative bidding structure of the proposed auction promises reduced computation at the platform side (Nisan, Roughgarden, Tardos, & Vazirani, 2007; Parkes & Ungar, 2001b). This is because a VCG requires the platform to solve $|\mathcal{N}| - 1$ NP-hard instances to compute critical payments for buyers and sellers. However, the iterative bidding structure of P-IDA can help distribute the computation across all participants of the auction (Parkes & Ungar, 2000, 2001a). Although the winner determination problem during each bidding round remains NP hard, the problem instances in P-IDA are much smaller than that in VCG as buyers only bid for a small subset of sellers in each round, which largely reduces the computational complexities (Parkes & Ungar, 2001b). In addition, P-IDA preserves the privacy of buyers and sellers. In the VCG, agents need to submit their complete valuations to compute a final schedule. However, in P-IDA, agents are not required to submit their private information since iterative biding is essentially a price system, not a *direct revelation mechanism*. The bidding and asking prices do not necessarily correspond to their values and costs. In addition, compared with single-sided multi-round auctions, such as those proposed in (Parkes, 2006; C. Wang & Dargahi, 2013; C. Wang et al., 2011), the two-sided structure of P-IDA facilitates trade between two groups in one market, which is more efficient than combining several single-sided auctions (Xia, Stallaert, & Whinston, 2005).

3.5 Computational study

In this section, we conduct a computational study to verify the performance of the proposed auction in terms of three evaluation metrics which are important to charger sharing market operators, namely efficiency, profit ratio and running time. First, we present the definitions of the evaluation metrics. Next, we describe the design of testing data. After that, we evaluate P-IDA by comparing its results with optimal solutions and that of two other scheduling heuristics. Finally, we evaluate P-IDA with the SA based approximate algorithm (P-IDA-SA) against larger problem instances to verify the scalability of the proposed auction.

3.5.1 Evaluation Metrics

The evaluation metrics are defined as follows:

- **Efficiency** of scheduling, $eff(K)$, is measured as the ratio of the social welfare of the final schedule K to the social welfare of the optimal schedule K^*

$$eff(K) = \frac{\sum_{(n,m)\in K}(v_{n,m} - r_{n,m} \cdot c_m)}{\sum_{(n,m)\in K^*}(v_{n,m} - r_{n,m} \cdot c_m)} \tag{6}$$

where K is the final schedule generated by the auction and K^* is the optimal schedule which maximizes the social welfare.

- **Profit ratio** of the auction, $pro(K)$, is measured as the sum of sellers' payoff in the final schedule K, as a fraction of the sum of the payoff in the optimal solution K^* that maximizes the social welfare

$$pro(K) = \frac{\sum_{(n,m) \in K} (p^b_{n,m} - c_m) \cdot r_{n,m}}{\sum_{(n,m) \in K^*} (v_{n,m} - r_{n,m} \cdot c_m)} \tag{7}$$

where $p^b_{n,m}$ is the bid price of buyer n for seller m in the final schedule K and $(p^b_{n,m} - c_m) \cdot r_{n,m}$ is the payoff of seller m in K. The profit ratio metric is designed to measure the degree to which the sellers make money by applying the auction.

- **Running time** of the auction refers to the computation time needed to terminate the auction on a charger sharing scheduling problem instance.

3.5.2 Design of Testing Data

We consider a day-ahead charger sharing scheduling setting with a 15 hour scheduling horizon on the next day from 07:00-22:00. We designate half-hour as our unit of time in the experiments. So the entire scheduling horizon can be divided into 30 half-hour charging units. For each seller m, the service start time s_m is randomly drawn from a uniform distribution in the range of 07:00-14:00. We assume that a seller offers at least 16 charging units. The number of charging units offered by m is randomly drawn from $[16, \min\{30, 2 \cdot (22 - s_m)\}]$. The service end time e_m of seller m can be determined based on s_m and the number of charging units offered. For a seller, the cost of a charging unit varies from \$1 to \$2.5 with a step \$0.1. For each buyer n, the arriving time $a_{n,m}$ is randomly drawn from a uniform distribution in the range of 07:00-21:30. According to a survey (Santos, McGuckin, Nakamoto, Gray, & Liss, 2011), the peak intervals of EV charging include 08:00-10:00, 12:00-14:00 and 18:00-20:00. Thus, we vary the relative proportions between the buyers to represent varying levels of heterogeneity in the buyer population. Specifically, our data is designed to ensure that during each of these three peak time intervals, around 20% of all buyers for the day arrive. A departure time $d_{n,m}$ of each buyer n is randomly drawn from a uniform distribution in the range of $[a_{n,m} + 1, \min\{a_{n,m} + 8, e_m\}]$ and required charging duration follows a uniform distribution over the interval $[1, \min\{d_{n,m} - a_{n,m}, \frac{C_n}{R_m}]$, where C_n is the battery capacity (kWh) of buyer n and R_m is the charging rate (kW) provided by seller m. For the sake of simplicity, we assume $C_n = 80$ kWh for all buyers and $R_m = 10$ kW for all sellers. The value of each buyer for a charging unit varies from \$0.1 to \$5, with a step \$0.1. For each buyer, there are multiple sellers in the market whose available time windows can accommodate the buyer's arriving and departure time. We assume the number of these sellers for each buyer is a random number from interval $[1, 0.4 \cdot M]$, where M is the number of sellers.

Given the above mentioned charger sharing scheduling setting, we generated 16 groups of testing problem instances with different sizes. The size of a problem instance is determined by the numbers of buyers and sellers considered. The numbers of buyers and sellers in all 16 groups are shown in TABLE 3.3. Each group contains 10 instances. The sizes of problem instances in Groups 1-13 are relatively small with the number of sellers ranging from 4 to 20 and the number of buyers ranging from 5 to 100. These instances are used to evaluate the efficiency of P-IDA against optimal solutions. Groups 14-16 contain much larger problem instances which are used to evaluate the scalability of P-IDA-SA.

Table 3.3: Numbers of buyers and sellers in all 16 testing groups

Num.of	Testing Groups															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sellers	4	4	4	4	5	5	5	5	6	6	6	6	20	40	100	500
buyers	5	10	15	20	5	10	15	20	5	10	15	20	100	200	500	1000
instances	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

3.5.3 Performance Evaluation of P-IDA

We evaluate the performance of P-IDA and P-IDA-SA by comparing their results with the optimal solutions computed by Cplex[8] solver assuming complete information of sellers' costs and buyers' values. The optimal solutions are the upper bound of the efficiency level that could be possibly achieved by any scheduling algorithms. In addition to that, we compare the performance of

[8] https://www.ibm.com/analytics/cplex-optimizer

P-IDA with a *First-Come-First-Served* (FCFS) algorithm and a *greedy allocation* algorithm. FCFS is a common resource allocation heuristic used in sharing economies. The implementation of FCFS in this charger sharing experiment gives the buyers who arrive earlier in the market priorities to choose their desired sellers first. The greedy heuristic is adopted from (Chichin et al., 2016), which is an efficient strategy for two-sided resource allocation. In our implementation, the greedy algorithm first finds each of the sellers a *feasible buyer list* which contains all the buyers to whom the seller's available time window is feasible. Then, the algorithm sorts the sellers in an ascending order based on their costs. For the first seller in the list, the algorithm assigns buyers in the seller's feasible buyer list to the seller one by one until the seller's charging window is used up. The order of the assignment is based on the buyer's value on the seller. The buyer with highest value goes first. The process is repeated until all sellers' charging windows are filled or all buyers have been allocated.

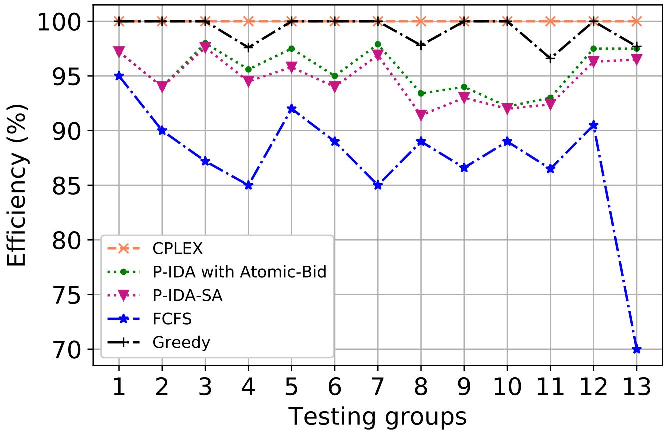
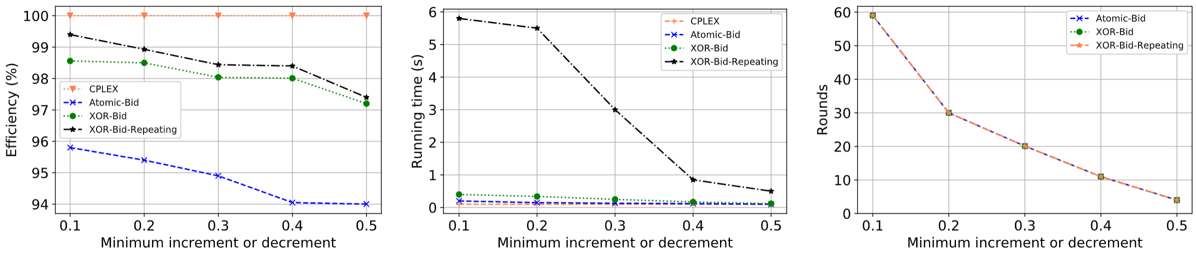


Figure 3.1: Efficiency comparison between FCFS, Greedy algorithm, P-IDA-SA and P-IDA with Atomic-Bid on Groups 1-13.

The average efficiency of schedules computed by FCFS, Greedy, P-IDA-SA and P-IDA over Groups 1-13 are shown in Fig. 3.1. Here, P-IDA is configured with $\epsilon = 0.2, a_{max} = 7, b_{min} = 0.1$ and only accepting Atomic bids. When applying SA to solve the winner determination model, we set the cooling cycle number to 1000 and the number of permutation to 30. It is observed that the P-IDA can achieve on average 94% efficiency compared with that of the optimal solutions (regarded as 100% efficiency). It also outperforms FCFS which has on average 84% efficiency. Furthermore, the P-IDA-SA can achieve on average 92.5% efficiency among all these groups. Our



(a) Efficiency

(b) Running time

(c) Number of rounds

Figure 3.2: The efficiency and running time performance of P-IDA with different ϵ under three bidding rule configurations.

implementation of the greedy heuristic can achieve on average 98.5% efficiency which is higher than that of P-IDA. However, the greedy algorithm requires buyers to reveal the highest prices they can pay and sellers to reveal their true service costs to the platform, which is not practical in a competing market environment. In the greedy algorithm implemented in (Chichin et al., 2016), the buyers are motivated to report their highest prices (values) with the introduction of critical value payments computed for each of the buyers. However, sellers have to be assumed to report their true costs, which does not make it a suitable scheduling mechanism for the charger sharing market. The comparison results indicate that P-IDA configuration is suitable and performs well in the charger sharing market setting. In fact, as shown in Fig. 3.2, we can further boost the efficiency of P-IDA by allowing XOR bids.

Fig. 3.2 (a) shows the efficiency of P-IDA with different values of ϵ under three bidding rule configurations including Atomic-Bid, XOR-Bid and XOR-Bid-Repeating. The results are averaged over the first 13 groups. It is shown that XOR-Bid has better efficiency (on average 97%) than that of Atomic-Bid (on average 94%). XOR-Bid-Repeating has the highest efficiency at 98%. However the cost comes with the high efficiency is more computation time due to the increased complexity of solving the winner determination model containing XOR bids. XOR-Bid-Repeating results in the largest winner determination model since it requires losing buyers to repeat their XOR-Bids even they have already reached their final bidding state. As shown in Fig. 3.2 (b), the P-IDA with XOR-Bid-Repeating has a relatively longer computation time than that of XOR-Bid and Atomic-Bid, especially when the value of ϵ is small. It is also shown in Fig. 3.2 (b) and (c) that the running times and number of rounds needed decrease when increasing the value of ϵ. Since ϵ controls the rate at which the prices of buyers and sellers are increased and decreased across rounds, with lower value of ϵ, the auction needs more rounds to clear the market which in turn, requires more computation time. For example, as shown in Fig. 3.2 (c), the number of bidding rounds needed for P-IDA to terminate drops from 59 with $\epsilon = 0.1$ to around 6 with $\epsilon = 0.5$ under all three bidding rule configurations. With a larger ϵ, the auction terminates quicker. However, it may also over-shoot some price equilibrium points otherwise can be reached if using a smaller ϵ, which results in slightly lower average efficiency. However, even with a large value of ϵ ($\epsilon = 0.5$), the proposed auction can achieve above 94% efficiency with Atomic-Bid and more than 97% efficiency with XOR-Bid or XOR-Bid-Repeating. This verifies that P-IDA is an efficient scheduling mechanism for the charger sharing market.

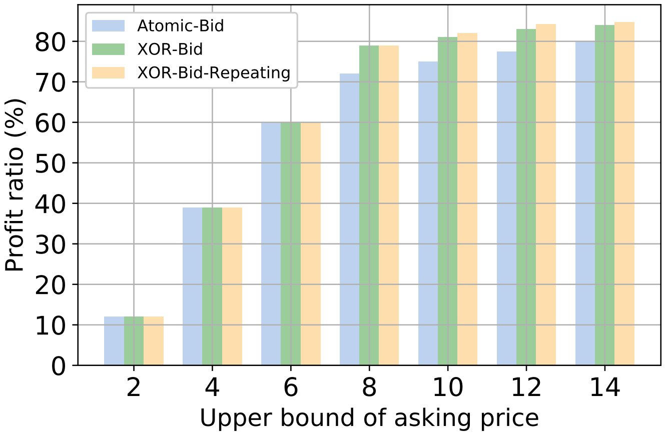


Figure 3.3: Profit ratio performance of the P-IDA averaged over Group 12 on different values of a_{upper}.

Fig. 3.3 shows the profit ratio performance of the P-IDA in relation to the average asking prices over Group 12. An immediate observation is that increasing the upper bound of the initial asking prices will improve the profit ratio that the P-IDA can achieve. The reason for this observation is that a low upper bound has higher probability of limiting the buyers' biding prices from reaching their values, therefore, presses down the equilibrium prices. The profit ratio improvement shown in Fig. 3.3 just indicates a general trend. Actual output will depend on the configurations of the problem instances. However it is observed that the marginal profit ratio gain decreases with larger upper bounds. Since larger upper bounds increase biding rounds which, in turn, incur more computation, it is not always desirable for the platform to prescribe large upper bounds given the demising gains. Again a suitable upper bound depends on problem configurations which may be learned from the transaction histories of a market.

3.5.4 Performance Evaluation of P-IDA-SA

In this subsection, we evaluate the scalability of the P-IDA-SA by comparing its running time and efficiency performance with P-IDA, FCFS and the greedy algorithms over testing groups 14-16. These groups contain large-sized problem instances with number of buyers ranging from 40 to 500 and sellers ranging from 200 to 1000. Optimal solutions could not be practically computed due to the size of the instances. However, as indicated in Fig. 3.1, the greedy allocation algorithm performs very close to the optimal solutions, therefore, makes a suitable benchmark algorithm for efficiency evaluation. Since we do not have optimal solutions for groups 14-16, instead of using the ratio to optimal, we use social welfare as the measure of efficiency.

The the social welfare (SW) and running time (RT) results of P-IDA, P-IDA-SA, Greedy and FCFS over Groups 14-16 are shown in TABLE 6.6. The social welfare performance of P-IDA-SA is noteworthy. It achieves on average 10% higher social welfare than Greedy and on average 40% higher social welfare than FCFS. In addition, P-IDA-SA achieves similar social welfare performance as P-IDA over Group 14. Note that we set a 20 hour time limit for P-IDA in the experiments. P-IDA does not terminate within the time limit over Groups 15 and 16, which explains the absence of social welfare values of P-IDA over Groups 15 and 16 in the table. P-IDA-SA also performs reasonably well in terms of running times. For Group 16 with 500 sellers and 1000 buyers, P-IDA-SA takes on average 5217s to terminate with $\epsilon = 0.5$. This level of responsiveness is acceptable for day-ahead charger sharing markets. The current running time performance is achieved using a PC equipped with a NVIDIA GeForce GPU. If further speedup is required, the SA-based winner determination algorithm can be parallelized and deployed on a cloud computing virtual machine with multiple CPUs and GPUs. In addition, the running times of P-IDA-SA can also be significantly reduced by increasing the value of ϵ without severely sacrificing the social welfare performance. In the auction, the value of ϵ defines the rate at which the prices are changed across rounds. The number of rounds in the auction are inversely-proportional to the value of ϵ, so doubling the value of ϵ will approximately halve the number of rounds needed to terminate. As shown in TABLE 6.6, the longest running time we have for P-IDA-SA is 10217 seconds (around 2.5 hours) with $\epsilon = 0.2$ for Group 16. If we increase the value of ϵ to 0.5, the auction terminates within 5217 seconds (1.45 hours) and the social welfare is only reduced approximately 1%. Depending on the social welfare performance needed, we can further increase ϵ to reduce running time to the level of a few minutes. It is important to note that although FCFS has better running time, its social welfare performance is much worse than that of the proposed auction. As shown in TABLE 6.6, for Group 16, the social welfare achieved by P-IDA-SA is more than four times higher than that of FCFS. Given the trade-off between solution quality and running time, the proposed auction is clearly favorable for a day-ahead market setting. Greedy has a similar social welfare performance as P-IDA-SA, however, it is not suitable for our two-sided market setting since it requires the buyers to reveal their highest paying prices and sellers to reveal their true costs.

Table 3.4: The social welfare (SW) and running time (RT) performance of P-IDA, P-IDA-SA, Greedy and FCFS on Groups 14-16.

Approach	Performance					
	Group 14		Group 15		Group 16	
	SW	RT (s)	SW	RT (s)	SW	RT (s)
Greedy	1431.4	34	3163.5	50	17064.3	55
P-IDA ($\epsilon = 0.2$)	1556.8	3192	-	$\gg$72000	-	$\gg$72000
P-IDA ($\epsilon = 0.5$)	1543.2	1313	-	$\gg$72000	-	$\gg$72000
P-IDA-SA ($\epsilon = 0.2$)	1548.9	921	3515	3450	17999	10217
P-IDA-SA ($\epsilon = 0.5$)	1444.9	395	3370	2361	17802	5217
FCFS	757.7	8	1916.2	15	3953	18

3.6 Summary

The rapid rise of the sharing economy demands the application of efficient resource scheduling mechanisms to two-sided markets. In this chapter, we propose a price-based iterative double auction for charger sharing scheduling with the objective of maximizing the social welfare of all participating drivers and charger owners. The proposed auction is budget balanced, individually rational and suitable for a competitive market environment in which myopic best responses from buyers and sellers are expected. It advances the existing literature by extending one-sided auction-based scheduling to two-sided markets. From practical application perspective, it achieves much better allocative efficiency than the first-come-first-served charger scheduling scheme which is commonly used by charger sharing platforms. It can also address concerns about privacy because bidders only need to reveal partial and indirect information about their valuations. In addition, it scales well to larger problem instances, which validates its potential for large scale markets.

Chapter 4

VOMA: A Privacy-Preserving Matching Mechanism Design for Community Ride-Sharing

In the previous chapter, we present an iterative double auction which serves as a matching mechanism for allocating vehicles to chargers in a charger sharing market. This approach is a price-based mechanism. However, in some markets, prices may not sufficient to determine the allocation such as labor markets and college admission (Haeringer, 2018) or price discrimination strategies is not desirable to some problem settings, like community ride sharing (Eva, n.d.). In this chapter, we propose a non-price negotiation mechanism and apply it to a community ride-sharing market.

Providing high-quality matching between drivers and riders is imperative for sustaining the growth of ride-sharing platforms. A user-focused matching mechanism design plays a key role in terms of ensuring user satisfaction. In this chapter, we consider the matching problem in the community ride-sharing setting, where drivers and riders have strong personal preferences over the matched counterparties. Obtaining high-quality solutions that accommodate drivers' and riders' preferences in such a setting is particularly challenging as drivers and riders maybe reluctant to share with the platform their personal preferences over their ride-sharing counterparties due to privacy and ethical concerns. To this end, we propose a VOting-based MAtching (VOMA) mechanism to compute near-optimal matching solutions for drivers and riders, while preserving their privacy. The mechanism is a distributed implementation of the simulated annealing meta-heuristic, which computes matching solutions by guiding drivers and riders in the distributed search process using an iterative voting protocol. We evaluate the performance of VOMA using test cases generated based on New York taxi data sets. The experiment results show that the proposed matching mechanism achieves on average 90.9% efficiency compared with optimal solutions. We also show that VOMA improves the vehicle miles traveled (VMT) savings by up to 35% compared to an alternative voting-based greedy matching mechanism. System scalability and other practical issues regarding the implementation of such a matching mechanism in community ride-sharing platforms are also discussed.

4.1 Introduction

The sharing economy promotes green consumption by better leveraging idle resources. As one segment of the sharing economy, shared mobility has gained popularity in recent years, serving as an innovative transportation strategy that enables users to gain short-term access to transportation services on an as-needed basis. Typical examples of these services include car-sharing, bike-sharing, on-demand ride services, micro-transit and ride-sharing (S. Shaheen et al., 2016; S. A. Shaheen, 2016). To further promote shared mobility as a service, popular Transportation Network Companies (TNCs) have also added *shared* or *carpool* features to their base ride-hailing services. For example, riders can now use apps such as Uber Pool, Didi Dache and Lyft Line to share rides with other riders. For TNCs, by combining individual trips into shared rides, the fleet size required to serve the same demand is reduced, therefore, saving their capital expenditure and operating costs. For cities, shared mobility helps improve citizens' quality of life by mitigating congestion, road traffic nuisance and air pollution.

In addition to the ride-sharing services provided by well-known TNCs, smaller scale community ride-sharing platforms have also attracted considerable attention as a transportation alternative in recent years, especially in sub-urban and rural areas. These *acquaintance-based* ride-sharing platforms provide shared rides among colleagues, seniors, and commuters within the same community (W. Chen, Mes, Schutten, & Quint, 2019; Payyanadan & Lee, 2018). Typical examples of platforms offering such services include Zimride[1], Eva[2], RideConnet[3] and Pendla[4].

Contrary to large ride-hailing platforms which rely on dedicated drivers to pick up and drop off riders, these "turn-your-neighbors-into-codrivers" platforms operate rather differently. In such platforms, drivers and riders are usually expected to register their personal information and preferences when joining the ride-sharing community (W. Chen et al., 2019). Once the registration is completed and verified by the platform, the commuter becomes a member and is able to submit their trip requests as either a driver or a rider, typically on a daily basis. Each request contains detailed trip information, such as the origin and the destination, and the earliest departure and latest arrival times. The platform uses the received trip request information as well as the commuters' registration information to compute suitable ride-sharing matches considering both distance and time related scheduling constraints and commuters' personal preferences.

Due to the acquaintance-based nature of community ride-sharing, a commuter's perceived quality of matching is largely determined by the *sharing counterparty*. As reported in (Cui, Makhija, Chen, He, & Khani, 2020; Sarriera et al., 2017), the primary concern of commuters in community ride-sharing is to whom they would share the ride. Since they may already know each other and they may share rides on a regular basis, commuters have strong *personal preferences* over the matched pairs (Xing, Warden, Nicolai, & Herzog, 2009; Yousaf, Li, Chen, Tang, & Dai, 2014). In addition to a list of favorite names, a driver or a rider may prefer to share the ride with specific groups of individuals, such as non-smokers or people with the same gender, as opposed to others. To this end, a high-quality matching between drivers and riders needs not only to respect scheduling constraints, but also to accommodate commuters' personal preferences. They may not accept the platform's matching recommendations if the assigned counterparties deviate too far from what they have expected (Agatz et al., 2012; Yousaf et al., 2014).

To achieve this objective, the unique challenge facing community ride-sharing platforms is to obtain complete personal preferences when making matching decisions. However, those preferences are usually considered as *private information* by the commuters. They certainly do not want to share

[1] http://zimride.com/
[2] https://eva.coop/
[3] https://www.rideconnect.com/
[4] https://www.pendla.com/en/

with the platform the list of people they do not like to share the rides with. To a certain extent, they even do not want to share more general preferences such as the gender and age of the matching counterparty due to ethical and privacy concerns (Aïvodji, Huguenin, Huguet, & Killijian, 2018; H. Zhang & Zhao, 2018). This (culturally) sensitive information is distributed among and controlled by individual commuters (Chau, Shen, & Zhou, 2020), to which the service operator does not have access. This distribution of information and control calls for decentralized optimization models which do not rely on centralized information but construct matching solutions through distributed negotiation among drivers and riders.

In this chapter, we propose VOMA, a distributed and privacy-preserving voting-based mechanism to solve the matching problems in community ride-sharing. This mechanism implements an improvement-based searching process which is inspired by the simulated annealing meta-heuristic, allowing drivers and riders to negotiate matching solutions iteratively in accordance with their individual preferences. This distributed negotiation process finds high-quality matching solutions and, at the same time, maintains the commuters' privacy.

The rest of this chapter is organized as follows. Section 4.2 reviews related works and positions the proposed approach in the literature. Section 4.3 describes the ride-sharing matching problem and its mathematical formulation. Section 4.4 formulates the commuters' utility functions, followed by the design of the matching mechanism in Section 4.5. In Section 4.6, we conduct a numerical study to evaluate the performance of the proposed mechanism. Finally we conclude our work in Section 4.7.

4.2 Related work

The design of an effective matching mechanism is essential for the success of a ride-sharing system, which has attracted considerable interest in the transportation research community (Agatz et al., 2012; Furuhata et al., 2013; Tafreshian et al., 2020). In the literature, most attention has been paid to maximizing travel cost savings and matching rate. Agatz *et al.* (Agatz, Erera, Savelsbergh, & Wang, 2011), for example, model the single driver and single rider ride-sharing problem as a max-weight bipartite matching problem and apply rolling horizon approaches to achieve high-quality matching solutions in real time. The objective is to maximize the overall system travel distance cost savings. With the same objective, a graph partitioning-based method is developed in (Tafreshian & Masoud, 2020) to solve a ride-matching problem. Similarly, with the objective of minimizing the total travel costs, Bei *et al.* (Bei & Zhang, 2018) model the ride-sharing problem as a combinatorial optimization problem and design an approximate algorithm to efficiently solve this NP-hard problem. In (Stiglic, Agatz, Savelsbergh, & Gradisar, 2015), Stiglic *et al.* reveal that introducing meeting points benefits ride-sharing systems by significantly increasing the number of successfully matched participants and reducing system-wide driving distances. Meeting points increase the flexibility of the ride-sharing system by expanding the set of feasible matches. In line with addressing the impact of riders' flexibility, Stiglic *et al.* (Stiglicc, Agatz, Savelsbergh, & Gradisar, 2016) use the same model to investigate the impact of matching flexibility, detour flexibility, and scheduling flexibility on matching rates. Masoud and Jayakrishnan (Masoud & Jayakrishnan, 2017) propose an optimal and real-time ride-matching algorithm that maximizes the matching rate in the ride-sharing system.

However, these papers compute the matching solutions between drivers and riders only based on trip-related constraints and preferences. They do not incorporate drivers' and riders' personal preferences, such as the age and gender of their sharing counterparties into the matching process. As indicated by (Agatz et al., 2012; H. Zhang & Zhao, 2018), in addition to the trip-related preference, a good understanding of the commuters' personal preferences is essential for maximizing commuter satisfaction when designing a ride-sharing system. If ride-sharing matching solutions do not satisfy the commuters' personal preferences, the solution may not be accepted, or the commuter may not consider to use the ride-sharing system in the future.

With the consideration of drivers and riders preferences, market-based approaches such as auctions have been proposed to obtain socially desirable solutions to matching problems in ride-sharing systems. Kleiner *et al.* (Kleiner, Nebel, & Ziparo, 2011) propose a second-price auction for a dynamic ride-sharing problem with the objective of balancing the trade-off between the vehicle kilometers travelled and the matching rate. This mechanism incentives drivers and riders to truthfully report their preferences by using a Vickrey–Clarke–Groves (VCG) based payment strategy. Following the same payment rule, Asghari *et al.* (Asghari & Shahabi, 2017) propose a truthful pricing mechanism based on second-price auction with a reserved price for ride-sharing. In a similar fashion, Bian *et al.* (Bian, Liu, & Bai, 2020) propose a novel mechanism, which adapts the traditional VCG mechanism to match drivers and riders in ride-sharing systems with the objective of maximizing the social welfare which is defined as the total preferences of commuters in the system.

Despite VCG's theoretical elegance, the implementations of VCG-type auctions have several limitations, which restrict its application to matching problems in community ride-sharing systems: (i) VCG auctions require drivers and riders to submit their complete preference valuations to compute a final schedule. In community ride-sharing, drivers and riders are often reluctant to do so due to privacy concerns. (ii) Lack of transparency is another practical concern in VCG auctions (C. Wang et al., 2011). It can be difficult to explain to the drivers and riders why a certain matching solution is chosen. (iii) The implementation of VCG auctions generally suffer from excessively high computation costs (Ausubel et al., 2006), which are impractical for ride-sharing systems with large numbers of drivers and riders. In addition to these limitations, the above mentioned matching mechanisms employ price discrimination strategies which are usually not feasible for community ride-sharing systems due to their business model constraints as a community service. In general, the ride-sharing service price is pre-set by the community and does not change on a daily basis (Eva, n.d.).

In this chapter, we propose a non-price negotiation mechanism for matching drivers and riders in a ride-sharing system. This mechanism not only respects the underlying ride-sharing matching constraints but also accommodates commuters' preferences in a privacy preserving way. Other non-price negotiation mechanisms for optimization problems have also been proposed in the literature (Fink, 2004; J. Gao & Wang, 2018; J. Gao, Wong, & Wang, 2019, 2021; Lang, Fink, & Brandt, 2016). However, these mechanisms are designed for one-sided problem settings, which are not suitable for our two-sided community ride-sharing problem. In this work, we propose a more general two-sided ride-sharing markets with two groups of players (drivers and riders). In addition, we do not assume a given preference value for each of the players as in (Fink, 2004; Lang et al., 2016). Instead, we explicitly design utility functions for drivers and riders (in Section VI) to support the negotiation process and to concisely express commuters' personal preferences over the matching solutions.

4.3 The community ride-sharing matching problem

In this section, we formulate the ride-sharing matching problem in the context of community ride-sharing systems.

4.3.1 System overview

We consider a community ride-sharing system where three types of players, namely drivers, riders and a service operator are involved. Drivers and riders first register their personal information, such as age, gender, education level, special interests, etc. After that, drivers and riders who would like to share a ride submit their trip offers and requests to the operator. The *trip offer* of a driver contains the origin and the destination locations and the earliest departure and the latest arrival times. Similarly, for a rider, the *trip request* contains the pick-up and drop-off locations, and the earliest departure and the latest arrival times. Finally, the operator computes an optimal matching solution between a group of trip offers and requests collected in a pre-determined time window. In what follows, we formally define the ride-sharing matching problem. Table 4.1 lists the main notations used in this chapter.

Table 4.1: Summary of main notations

Notation	Description
f	A pre-determined time window
$\mathcal{D}^f$	A set of drivers in f, indexed by d_i
$\mathcal{R}^f$	A set of riders in f, indexed by r_i
e_{d_i}, e_{r_i}	Earliest departure time of d_i and r_i
l_{d_i}, l_{r_i}	Latest arrival time of d_i and r_i
o_{d_i}, o_{r_i}	Origin locations of d_i and r_i
w_{d_i}, w_{r_i}	Destination locations of d_i and r_i
$\delta_{o,w}$	Driving distance between locations o and w
$t_{o,w}$	Driving time between locations o and w
S	A ride-sharing feasible matching solution
dt_{d_i,r_i}	Decision variable: the departure time of driver d_i when matching with rider r_i
$\mathbb{1}_{dt_{d_i,r_i}\in\mathbb{R}^+}$	Indicator function: $\mathbb{1}_{dt_{d_i,r_i}\in\mathbb{R}^+} = 1$ if $dt_{d_i,r_i} \in \mathbb{R}^+$ is true
$u_{r_i}(S), u_{d_i}(S)$	Utility functions of rider r_i and driver d_i on S
T	Number of negotiation rounds
ϱ	Acceptance quota
ψ	Number of candidate matching solutions per round

Let $f \doteq (\omega_f^+, \omega_f^-)$ be a pre-determined time window with length $\omega_f^- - \omega_f^+$, where ω_f^+ is the start time and ω_f^- is the end time. Let $\mathcal{D}^f$ be a set of drivers who submit their trip offers within f. The trip offer of driver $d_i \in \mathcal{D}^f$ is defined by a five-tuple $\alpha(d_i) \doteq (ID_{d_i}, o_{d_i}, w_{d_i}, e_{d_i}, l_{d_i})$, where ID_{d_i} is the driver's identification number, and o_{d_i} and w_{d_i} respectively represent the origin and the destination locations of driver d_i. Moreover, e_{d_i} indicates the earliest time that driver d_i can depart from his or her origin o_{d_i} and l_{d_i} indicates the latest time d_i should arrive at his or her destination w_{d_i}.

Let $\mathcal{R}^f$ be a set of riders who submit their trip requests within f. The trip request of a rider $r_i \in \mathcal{R}^f$ is defined by a five-tuple $\alpha(r_i) \doteq (ID_{r_i}, o_{r_i}, w_{r_i}, e_{r_i}, l_{r_i})$, where ID_{r_i} is the identification number of rider r_i. o_{r_i} and w_{r_i} are the pick-up and drop-off location, respectively. e_{r_i} is earliest pick-up time of rider r_i and l_{r_i} is the latest drop-off time of rider r_i.

Based on the trip offers and requests submitted by drivers and riders, the solution to the ride sharing problem is a matching solution S which contains the matched rider-driver pairs and the corresponding departure time of the drivers. Let dt_{d_i,r_i} be the departure time of driver d_i when matching with rider r_i, where $dt_{d_i,r_i} \in \mathbb{R}^+$ if driver d_i is assigned to pick up and deliver rider r_i, and $dt_{d_i,r_i} = -1$ if otherwise. We say that a matching solution S is *feasible* if and only if the following constraints are satisfied for each $dt_{d_i,r_i} \in \mathbb{R}^+$:

(i) The departure time of a driver to pick-up a rider should be no earlier than the earliest departure time of the driver, i.e., $dt_{d_i,r_i} \geq e_{d_i}, \ \forall r_i \in \mathcal{R}^f, \forall d_i \in \mathcal{D}^f$,

(ii) The arrival time of a driver to his or her destination after delivering a rider should be no later than the latest arrival time, i.e., $dt_{d_i,r_i} + t_{o_{d_i},o_{r_i}} + t_{o_{r_i},w_{r_i}} + t_{w_{r_i},w_{d_i}} \leq l_{d_i}, \ \forall r_i \in \mathcal{R}^f, \forall d_i \in \mathcal{D}^f$, where $t_{o_{d_i},o_{r_i}} + t_{o_{r_i},w_{r_i}} + t_{w_{r_i},w_{d_i}}$ is the time required for driver d_i going through locations o_{d_i}, o_{r_i}, w_{r_i} and w_{d_i}, as shown in the right side of Figure 4.1.

(iii) The pick-up time of a driver (the time a driver arrives at the matched rider location) should be no earlier than the rider's earliest departure time, i.e., $dt_{d_i,r_i} + t_{o_{d_i},o_{r_i}} \geq e_{r_i}, \ \forall r_i \in \mathcal{R}^f, \forall d_i \in \mathcal{D}^f$,

(iv) The arrival time of a rider at his or her destination should be no later than the latest drop-off time, i.e., $dt_{d_i,r_i} + t_{o_{d_i},o_{r_i}} + t_{o_{r_i},w_{r_i}} \leq l_{r_i}, \ \forall r_i \in \mathcal{R}^f, \forall d_i \in \mathcal{D}^f$,

(v) A driver is allowed to serve at most one rider, i.e., if $dt_{d_i,r_i} \in \mathbb{R}^+$ and $dt_{d_i,r_i'} \in \mathbb{R}^+$, then $r_i = r_i', \ \forall d_i \in \mathcal{D}^f$,

(vi) A rider can only be matched to a driver, i.e., if $dt_{d_i,r_i} \in \mathbb{R}^+$ and $dt_{d_i',r_i} \in \mathbb{R}^+$, then $d_i = d_i', \ \forall r_i \in \mathcal{R}^f$,

(vii) A driver only matches to a rider with a positive distance saving, i.e., if $dt_{d_i,r_i} \in \mathbb{R}^+$, then $\delta_{o_{r_i},w_{r_i}} + \delta_{o_{d_i},w_{d_i}} - (\delta_{o_{d_i},o_{r_i}} + \delta_{o_{r_i},w_{r_i}} + \delta_{w_{r_i},w_{d_i}}) > 0 \ \ \forall r_i \in \mathcal{R}^f, \ \forall d_i \in \mathcal{D}^f$, where $\delta_{o_{r_i},w_{r_i}}$ and $\delta_{o_{d_i},w_{d_i}}$ are the estimated distances of rider r_i and driver d_i when traveling alone, as shown in Figure 4.1 (a). $\delta_{o_{d_i},o_{r_i}} + \delta_{o_{r_i},w_{r_i}} + \delta_{w_{r_i},w_{d_i}}$ is the overall length of ride share between rider r_i and driver d_i, as shown in Figure 4.1 (b).

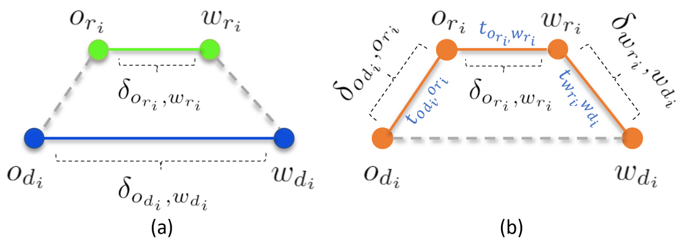


Figure 4.1: An example of a single rider single driver ride-share arrangement. Single paths of rider (green) and driver (blue), overall path required for ride share (orange).

Drivers and riders have preferences over feasible matching solutions regarding the matched counterparties and the arranged times and itineraries. According to the utility theory (Fishburn, 1970), these preferences can be quantified by designing *utility functions*. Let $u_{d_i}(S)$ and $u_{r_i}(S)$ be the utility functions of driver d_i and rider r_i on a matching solution S, respectively. The objective of maximizing the overall utilities across all drivers and riders is defined as:

$$\max \sum_{r_i \in \mathcal{R}^f} \sum_{d_i \in \mathcal{D}^f} \mathbb{1}_{dt_{d_i,r_i} \in \mathbb{R}^+} (u_{r_i}(S) + u_{d_i}(S)) \tag{8}$$

where $\mathbb{1}_{dt_{d_i,r_i} \in \mathbb{R}^+}$ is the indicator function that equals 1 if $dt_{d_i,r_i} \in \mathbb{R}^+$ is true and 0 otherwise. This objective considers the benefits of both drivers and riders, which ensures the continuous growth of ride-sharing systems in the context of the sharing economy.

In what follows, we introduce the design of utility functions of drivers and riders, respectively.

4.4 The design of utility functions

A commuter's preference over a set of matching solutions is determined by the matched counterparty and travel schedule (time and itinerary). Therefore, the utility function which is designed to quantify the degree of a commuter's preference over a set of matching solutions will evaluate a commuter's satisfaction levels to both the matched counterparty and the travel schedule prescribed by the operator. We assume commuters have *quasilinear utility functions* (Shoham & Leyton-Brown, 2008), such that the overall utility function of a commuter can be formulated as a linear combination of two sub-utility functions, namely *matching utility function* and *scheduling utility function*, designed for quantifying a commuter's satisfaction levels to a matched counterparty and a travel schedule, respectively. In what follows, we first present the two sub-utility functions for both drivers and riders, then we propose the design of a commuter's utility function over complete matching solutions.

As observed by the authors in (Cui, Makhija, Chen, He, & Khani, 2021), the perceived satisfaction level of a commuter to his or her matched counterparty is mainly influenced by a set of personal attributes, such as gender, age, education level, smoking and drinking habits, etc. The matching utility function therefore assigns a numerical value to the matched counterparty by evaluating the set of personal attributes of the matched counterparty. Let K be a set of personal attributes for commuters, such as age and gender. Each attribute $k \in K$ can be decomposed into distinct classes based on the characteristic of the attribute. For example, attribute gender can be classified into two classes, female and male. Let X^k be a set that contains all the classes of attribute k. Let $R^{resi}_{c_i}(X^k) \subseteq X^k$ be the *registered* classes of attribute k for commuter $c_i \in \{\mathcal{D}^f \cup \mathcal{R}^f\}$. For example, $R^{resi}_{d_1}(\{female, male\}) = \{female\}$ indicates that the gender of driver d_1 is female. Let $P^{pref}_{c_i}(X^k)$ be the *preferred* classes of attribute k for commuter $c_i \in \{\mathcal{D}^f \cup \mathcal{R}^f\}$, with $P^{pref}_{c_i}(X^k) \subseteq X^k$. For example, $P^{pref}_{d_1}(\{female, male\}) = \{female\}$ indicates that driver d_1 prefers to be matched with a female rider.

Example 1. *To explain the above defined concept, we illustrate an example with two attributes: age and smoking habit. In this example, the first attribute, age attribute, is decomposed into* *three classes: young adulthood (18 to 35 years), middle age (36 to 55 years), and older adulthood (56 years and older) and a^y, a^m and a^o are assigned to the classes, respectively. Thus we have $X^1 = \{a^y, a^m, a^o\}$. Smoking habit, the second attribute, is decomposed into two classes: smoker and nonsmoker. s^y and s^n are assigned to the classes, respectively. Therefore, $X^2 = \{s^y, s^n\}$. If driver d_1 prefers to be matched with young, non-smoking riders, then his or her choices would be: $P^{pref}_{d_1}(\{a^y, a^m, a^o\}) = \{a^y\}$ and $P^{pref}_{d_1}(\{s^y, s^n\}) = \{s^n\}$. Similarly, for riders, $P^{pref}_{r_1}(\{a^y, a^m, a^o\}) = \{a^y\}$ and $P^{pref}_{r_1}(\{s^y, s^n\}) = \{s^y, s^n\}$ indicate that rider r_1 prefers young drivers and has no preference with regard to the smoking habit attribute.*

The matching utility function is formulated based on the *Jaccard similarity coefficient* (Niwattanakul, Singthongchai, Naenudorn, & Wanapu, 2013), which measures the similarities between a commuter's preferred attributes and the matched counterparty's registered attributes. Let $J(P^{pref}_{d_i}(X^k), R^{regi}_{r_i}(X^k))$ be the Jaccard similarity coefficient between the driver's preferred class set and the rider's registered class set of attribute k, which is computed as: $J(P^{pref}_{d_i}(X^k), R^{regi}_{r_i}(X^k)) = \frac{|P^{pref}_{d_i}(X^k) \bigcap R^{regi}_{r_i}(X^k)|}{|P^{pref}_{d_i}(X^k) \bigcup R^{regi}_{r_i}(X^k)|}$. Thus the matching utility function of driver d_i with rider r_i, denoted as $u^{match}_{d_i}(r_i)$, is defined as:

$$u^{match}_{d_i}(r_i) = \sum_{k \in K} w^k_{d_i} \cdot J(P^{pref}_{d_i}(X^k), R^{regi}_{r_i}(X^k)) \tag{9}$$

where $w^k_{d_i}$ is the weight of the attribute k for driver d_i, representing the marginal impacts of the attribute on the preference value. Similarly, rider r_i's matching utility function with driver d_i, denoted as $u^{match}_{r_i}(d_i)$, is defined as:

$$u^{match}_{r_i}(d_i) = \sum_{k \in K} w^k_{r_i} \cdot J(P^{pref}_{r_i}(X^k), R^{regi}_{d_i}(X^k)) \tag{10}$$

where $w^k_{r_i}$ is the weight of the attribute k for rider r_i.

In addition to the matching utility function, commuters also have scheduling utility functions to measure their satisfaction levels to the travel schedules. We assume a driver's satisfaction level to a travel schedule is determined by the length of the detours to pick-up and deliver a rider. As shown in Figure 4.1 (b), the length of the detours of driver d_i to pick-up and deliver rider r_i, denoted as

de_{d_i,r_i}, can be computed by $\delta_{o_{d_i},o_{r_i}} + \delta_{o_{r_i},w_{r_i}} + \delta_{w_{r_i},w_{d_i}} - \delta_{o_{d_i},w_{d_i}}$, where $\delta_{o_{d_i},w_{d_i}}$ is the length of the original trip of driver d_i; and $\delta_{o_{d_i},o_{r_i}} + \delta_{o_{r_i},w_{r_i}} + \delta_{w_{r_i},w_{d_i}}$ is the joint trip length if driver d_i is matched with a rider r_i. Thus the scheduling utility function of a driver d_i with a rider r_i, denoted as $u_{d_i}^{sched}(r_i)$, is defined as:

$$u_{d_i}^{sched}(r_i) = -\alpha_{d_i} de_{d_i,r_i} \tag{11}$$

where $\alpha_{d_i} \in \mathbb{R}^+$ is the detour related cost coefficient of driver d_i. This function indicates that, the driver's satisfaction level decreases with the increase of the length of detours.

On the other side, we define a rider's utility in terms of travel schedule as the wait time before being picked up by a driver. Let $t_{o_{d_i},o_{r_i}}$ be the driving time of driver d_i to pick up rider r_i. The wait time of rider r_i being matched with driver d_i, dented as w_{r_i,d_i} can be computed as $dt_{d_i,r_i} + t_{o_{d_i},o_{r_i}} - e_{r_i}$, where $dt_{d_i,r_i} + t_{o_{d_i},o_{r_i}}$ is the time needed for driver d_i to pick up rider r_i. Thus, the scheduling utility function of a rider r_i with a driver d_i, denoted as $u_{r_i}^{sched}(d_i)$, is defined as:

$$u_{r_i}^{sched}(d_i) = -\lambda_{r_i} w_{r_i,d_i} \tag{12}$$

where $\lambda_{r_i} \in \mathbb{R}^+$ represents the delay tolerance rate of rider r_i. With the increase of wait time, a rider's preference value decreases accordingly.

Given the quasilinear utility function assumption, the utility function of a driver d_i over a matching solution S can be represented as the weighted sum of equations (2) and (4) as follows:

$$u_{d_i}(S) = u_{d_i}^{match}(r_i) + \chi_d u_{d_i}^{sched}(r_i) \tag{13}$$

where χ_d is the weight coefficient of a driver's utility function to align the units of $u_{d_i}^{match}(r_i)$ and $u_{d_i}^{sched}(r_i)$. Similarly, the utility function of a rider r_i over S can be represented as the weighted sum of equations (3) and (5) as follows:

$$u_{r_i}(S) = u_{r_i}^{match}(d_i) + \chi_r u_{r_i}^{sched}(d_i) \tag{14}$$

where χ_r is the weight coefficient of a rider's utility function to align the units of $u_{r_i}^{match}(d_i)$ and $u_{r_i}^{sched}(d_i)$. There are a number of ways to choose such a utility function. For the purposes of this work, however, the formulations in (13) and (14) are chosen for their computational and conceptual simplicity.

As mentioned previously, the utility functions of commuters are their private information which is not accessible to the operator. Given the privacy constraint of commuters, in the next section, we design a privacy-preserving voting-based negotiation mechanism to match drivers and riders.

4.5 Voting-based matching mechanism

In this section, we present VOMA for matching drivers and riders in a ride-sharing system. This mechanism is a distributed implementation of a meta-heuristic through which commuters collaboratively improve the overall utility of a matching solution using a privacy preserving voting procedure inspired by *Simulated Annealing* (Talbi, 2009). This procedure consists of multiple hill climbing iterations during which improving moves are always accepted and a deteriorating move is accepted with a given probability. The acceptance probability is controlled by the cooling process, which is computed using a quota rule similar to that proposed in (Lang et al., 2016). In the following, we first present the design of the overall negotiation mechanism. We then discuss its practical implementation issues by showing its game theoretic properties and a multi-agent systems implementation.

4.5.1 Voting based negotiation

Figure 4.2 shows a high level flow of the proposed negotiation process. First, based on the trip offers and requests submitted by drivers and riders, a set of feasible matching solutions, called candidate matching solutions is randomly generated by the service operator. Upon receiving the candidate matching solutions, drivers and riders compute their utilities over these matching solutions based on their utility functions (defined in equations (13) and (14)). Afterwards, drivers and riders vote on a subset of matching solutions and submit their voting decisions to the operator. This voting decision is restricted by a quota rule, in which drivers and riders have to vote on a certain number of candidate matching solutions so that deteriorating moves are accepted and local optima can be overcome. After receiving all the responses from drivers and riders, the operator has to decide which candidate matching solution can be selected as an incumbent matching solution. The operator proceeds with the generation of new candidate matching solutions until the maximum number of negotiation rounds is reached.

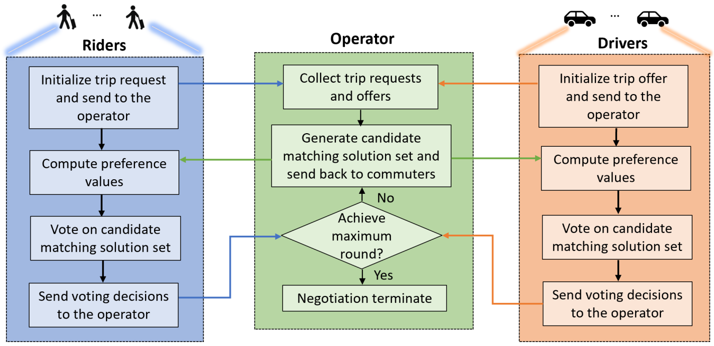

Riders
Initialize trip request and send to the operator
Compute preference values
Vote on candidate matching solution set
Send voting decisions to the operator
Operator
Collect trip requests and offers
Generate candidate matching solution set and send back to commuters
No
Achieve maximum round?
Yes
Negotiation terminate
Drivers
Initialize trip offer and send to the operator
Compute preference values
Vote on candidate matching solution set
Send voting decisions to the operator

Figure 4.2: A high level flow of the negotiation process.

The details of the negotiation process are described in Algorithm 3. As initial inputs, VOMA requires the number of the negotiation round T, the acceptance quota ϱ (how many matching solutions need to be accepted in the first negotiation round), the number of candidate matching solutions per round ψ, a set of trip offers $\alpha(d_i), \forall d_i \in \mathcal{D}^f$ and a set of trip requests $\alpha(r_i), \forall r_i \in \mathcal{R}^f$.

At first, based on the submitted trip offers and requests, the operator matches drivers and riders using a random allocation procedure (see Line 2 of Algorithm 1) to generate an initial permutation $\Pi^{ini} = \{\xi_{d_i,r_i} | \, \forall d_i \in \mathcal{D}^f, \forall r_i \in \mathcal{R}^f\}$, where $\xi_{d_i,r_i} = (\alpha(d_i), \alpha(r_i))$ is a driver-rider pair, binding a driver d_i's offer $\alpha(d_i)$ to a rider r_i's request $\alpha(r_i)$. Given the initial permutation Π^{ini}, an initial matching solution S^{ini} is generated by calling the TIMETABLING procedure (see Algorithm 4 for details). The initial matching solution becomes the final matching solution when the negotiation terminates. For the quota ϱ, in line 4 of Algorithm 1, a cooling factor θ_ϱ, which decreases the quota ϱ in each round is computed (see Line 33 of Algorithm 1). After the initialization, the negotiation procedure starts and is repeated until the Tth round. Each round contains three steps, namely candidate matching solution set generation, voting and incumbent matching solution determination.

- *Candidate matching solution set generation:* At round t, an incumbent permutation Π^{inc} is first generated using an decoding *bijection* function $f(\cdot)$, which maps or converts a matching

Algorithm 3 VOMA

```
Require: $T, \varrho, \psi, \alpha(d_i), \forall d_i \in \mathcal{D}^f, \alpha(r_i), \forall r_i \in \mathcal{R}^f$
Ensure: $S^*$
$\Pi^{ini} \leftarrow \{\xi_{d_i,r_i} | \forall d_i \in \mathcal{D}^f, \forall r_i \in \mathcal{R}^f\}$
$S^{ini} \leftarrow$ TIMETABLING($\Pi^{ini}$)
$\theta_\varrho \leftarrow \varrho^{\frac{1}{1-T}}$
for $t \leftarrow 1$ to $T$ do                                          ▷ Negotiation starts
    AcceptedSolutions $\leftarrow \emptyset$
    $CS \leftarrow S^{ini}$
    $CP \leftarrow \{f(S^{ini})\}$
    for $k \leftarrow 1$ to $\psi - 1$ do                               ▷ Negotiation starts
        $\Pi^c \leftarrow$ SWAP($f(S^{ini})$)
        $CP \leftarrow CP \cup \{\Pi^c\}$
    end for
    for all $\Pi^c \in CP$ do
        $S^c \leftarrow$ TIMETABLING($\Pi^c$)
        $CS \leftarrow CS \cup \{S^c\}$
    end for
    for all $c_i \in \mathcal{D}^f \bigcup \mathcal{R}^f$ do
        $\overrightarrow{Z_{c_i}} \leftarrow$ VOTE$_{c_i}$($CS, v_{c_i}(S^c), \varrho$)
    end for
    for all $S^c \in CS$ do
        if $\sum_{c_i \in \mathcal{D}^f \bigcup \mathcal{R}^f} \overrightarrow{Z_{c_i}}[S^c] \geq Threshold$ then
            AcceptedSolutions $\leftarrow$ AcceptedSolutions $\cup \{S^c\}$
        end if
    end for
    if AcceptedSolutions $\neq \emptyset$ then        ▷ $x$ is a normal distribution number in the range of $[0, 1]$
        $S^{ini} \leftarrow$ RANDOMSELECT(AcceptedSolutions)
    end if
    $\varrho \leftarrow \varrho \cdot \theta_\varrho$
end for
$S^* \leftarrow S^{ini}$
```

solution to its corresponding permutation (see line 9 of Algorithm 3). Then, a set of candidate permutations are generated by applying the *independent swapping* procedure adopted from (Talbi, 2009). This results in the candidate permutation set CP, including the incumbent permutation Π^{inc} and $\psi - 1$ candidate permutations (lines 11-14 of Algorithm 3). ψ is the number of permutations per round. Π^{c} is a candidate permutation. The swap operator is conducted by randomly swapping the locations of two pairs, such as $\xi_{d_i,r_i}(d_i \in \mathcal{D}^f, r_i \in \mathcal{R}^f)$ and $\xi_{d_k,r_k}(d_k \in \mathcal{D}^f, r_k \in \mathcal{R}^f)$ in Π^{inc}, where $i \neq k$. For each permutation in the set CP, the corresponding matching solution is generated by calling the TIMETABLING procedure. Finally, the candidate matching solution set CS is completed by converting all permutations in CP into their corresponding matching solutions (see lines 16-19 of Algorithm 3).

- *Voting:* Drivers and riders evaluate the candidate matching solutions based on their private utilities. Here, this decision is subject to the current acceptance quota ϱ. Specifically, upon receiving the candidate matching solution set CS, each driver d_i and rider r_i needs to vote on the best ϱ matching solutions from it based on their utilities. To compute the value of ϱ at each round of negotiation, we adopt a geometric cooling strategy: $\varrho \leftarrow \varrho \cdot \theta_\varrho$ proposed by (Blum & Roli, 2003) (see line 33 of Algorithm 3). Here θ_ϱ is the cooling factor, initialized as $\theta_\varrho \leftarrow \varrho^{\frac{1}{1-T}}$, where T is the predefined number of negotiation rounds. The cooling strategy corresponds to an exponential decay of ϱ and guarantees that only one matching solution has to be accepted ($\varrho = 1$) for each driver and rider at the last negotiation round. For all voted matching solutions, a commuter $c_i \in \mathcal{D}^f \bigcup \mathcal{R}^f$ sets the value of the corresponding position in the binary decision vector $\overrightarrow{Z_{c_i}}$ from 0 to 1 (lines 21-23 of Algorithm 3) (1 means accept a matching solution and 0 means reject a matching solution). Naturally, commuters only accept improved matching solutions compared with the initial matching solution S^{ini} in terms of utilities. However, in this protocol, each driver or rider has to accept at least ϱ matching solutions from a candidate matching solution set. Specifically, during each round, drivers and riders will order candidate matching solutions based on his or her utilities on them and attribute 1 to the first ϱ matching solutions in $\overrightarrow{Z_{c_i}}$. This indicates that, commuters may need to accept non-improved matching solutions or even deteriorated matching solutions compared with the initial matching solution to reach the quota ϱ.

- *Incumbent matching solution determination:* After the voting process, the operator assesses the responses from the commuters. If a matching solution from a candidate matching solution set CS is accepted by a certain number of commuters (given by *Threshold*, see line 26 of Algorithm 3), it becomes a potential initial matching solution for the next negotiation round and is thus added to the *AcceptedSolutions* set. $\sum_{c_i \in \mathcal{D}^f \bigcup \mathcal{R}^f} \overrightarrow{Z_{c_i}}[S^c]$ represents the sum of the commuters who accept a candidate matching solution S^c. If there are multiple matching solutions in the set *AcceptedSolutions*, one matching solution from the set is randomly selected and becomes the new initial matching solution. Otherwise, the initial matching solution in the current round is carried over to the next negotiation round. The negotiation starts with a new CS generated from the new CP and terminates after T rounds. The last initial matching solution will be the final matching solution S^* generated from the voting-based negotiation process.

4.5.2 TIMETABLING for initial and candidate matching solutions generation

A matching solution is generated by calling the TIMETABLING procedure, which is shown in Algorithm 2. This procedure adopts the priority rule-based scheduling method (Kolisch, 1996), in which an allocation decision is first generated and a specific priority rule is then employed. To be specific, for a pair ξ_{d_i,r_i} in Π^c, the operator will first verify if it satisfies all the ride-sharing constraints (i)-(vii) (see Section III for details). If the constraints are satisfied, the driver and rider are matched and the departure time of the driver in this pair is assigned by the earliest feasible start time (EST) rule (line 4 of Algorithm 2). Otherwise the pair is denied. The process is repeated until all pairs in Π^c are allocated and a matching solution is generated.

4.5.3 Game theoretic property of VOMA

The key factor that ensures the practical implementation of VOMA in community ride-sharing is that commuters follow the procedure by voting for the best ϱ matching solutions out of the candidate matching solution set as prescribed by the negotiation procedure. In the following, we prove that

Algorithm 4 TIMETABLING

```
Require: $\Pi^c$
Ensure: $S^c$
for all $\xi_{d_i,r_i} \in \Pi^c$ do
    if constraints (i)-(vii) are satisfied then
        $dt_{d_i} \leftarrow$ Assign a departure time based on EST
        Else $dt_{d_i} = -1$
    end if
    $S^c \leftarrow S^c \cup \{(\xi_{d_i,r_i}, dt_{d_i})\}$
end for
```

the best strategy for a rational (utility maximizing) commuter is to vote according to the procedure.

Proposition 1. *Given the negotiation procedure, the best strategy for a commuter unaware of other commuters' utilities is to vote on his or her best matching solutions out of the candidate matching solution sets.*

Proof. According to the expected utility theory (Von Neumann & Morgenstern, 2007), a commuter chooses the matching solutions with the highest expected utilities. Let L_{c_i} be a simple lottery of a commuter $c_i \in \mathcal{D}^f \cup \mathcal{R}^f$, which is a probability distribution over candidate matching solutions formulated as: $L_{c_i} = [p^1_{c_i} : S^1, ..., p^m_{c_i} : S^m]$, where m is the number of candidate matching solutions and $p^m_{c_i}$ is the probability measure on candidate matching solution S^m with $p^m_{c_i} \geq 0, \forall m$ and $\sum_m p^m_{c_i} = 1$. In our case $p^m_{c_i}$ indicates the probability of commuter c_i that candidate matching solution $S^m \in CS$ will be selected as an initial matching solution by the operator. Thus the expected utility of the matching solution S^m for commuter c_i is denoted as $p^m_{c_i} \times u_{c_i}(S^m)$. Let $CS^{\varrho}_{c_i}$ be the set of voted matching solutions of commuter c_i. To maximize his or her expected utility, a commuter c_i would vote on ϱ matching solutions with the highest expected value which is:

$$\arg \max_{CS^{\varrho}_{c_i} \subseteq CS} \sum_{S^m \in CS^{\varrho}_{c_i}} p^m_{c_i} \times u_{c_i}(S^m)$$

However, with private information, a commuter is not aware of other commuters' utilities. Thus the probability of which matching solution would be selected as an initial matching solution by the operator cannot be distinguishable. Thus $p^1_{c_i} = ... = p^m_{c_i} = \bar{p}, \forall c_i, m$, where $\bar{p}$ is a constant number.

The decision of each commuter is now reduced to

$$\arg \max_{CS^{\varrho}_{c_i} \subseteq CS} \sum_{S^m \in CS^{\varrho}_{c_i}} \bar{p} \times u_{c_i}(S^m)$$

which is equivalent to $\arg\max_{CS^{\varrho}_{c_i} \subseteq CS} \sum_{S^m \in CS^{\varrho}_{c_i}} u_{c_i}(S^m)$. It immediately follows that, for each commuter, to maximize his or her expected utility, the best strategy is to vote on the best ϱ matching solutions out of the candidate matching solution set CS. □

4.5.4 Multiagent systems implementation

Due to the computation and communication efforts involved, to apply VOMA to real world community ride-sharing settings, an automated negotiation system has to be developed. One possible structure of such automated systems can be a multi-agent systems implementation, where drivers and riders are represented by their software proxy agents, which negotiate on behalf of them. A commuter's utility function can be programmed into his or her proxy agent. In the meantime, the agents should be equipped with the algorithm to compute the utilities of the candidate matching solutions and vote along the negotiation process. As observed in our experiments, the multi-agent system can quickly converge during the negotiation process and find a near-optimal matching solution. For easy access, drivers and riders may install their proxy agents on a personal computer, a smartphone, or other mobile devices.

4.6 Numerical study

In this section, we conduct a numerical study to assess the performance of VOMA in terms of three evaluation metrics that are important to drivers, riders and community ride-sharing platforms. In the following, we first present the evaluation metrics. Next, we describe the testing environment used to evaluate the performance of VOMA. Finally, we evaluate the performance of VOMA by comparing its solutions with optimal solutions and those generated by an alternative matching mechanism against the proposed evaluation metrics.

4.6.1 Evaluation metrics

Social welfare

The social welfare of the ride-sharing system during a pre-determined time window is defined as the sum of utilities of matched drivers and riders, formulated in equation (8).

Efficiency

The efficiency is defined as the ratio of the social welfare of the solution generated by VOMA to the social welfare of the optimal solution.

Vehicle miles travelled (VMT) savings

VMT savings is defined as the difference between the sum of the original trip distances of the commuters when traveling alone and the distances of the shared trips.

4.6.2 Test instance description and parameter setting

We construct the test instances based on the trip information obtained from 2016 New York City green taxi public data[5]. The information available for each trip includes the timestamps and GPS coordinates of the trip origin and destination. In this study, data from February 5th to February 6th are extracted for the purpose of evaluating the performance of VOMA and we randomly assign 50% of the total commuters to be drivers and make the rest riders. The information of the earliest departure time and the latest arrival time is not included in the original data, without loss of generality, we use the pick-up and drop-off times in the data as the earliest departure and the latest arrival times, respectively. We extract the trip records whose earliest departure times fall into evening peak hours ($6:00pm + \Delta T$) to generate the test instances, where ΔT is the length of a pre-determined time window. We vary the length of ΔT so that different numbers of drivers and riders are collected, which are shown in Table II. The reason for varying the length of ΔT is to simulate different market conditions in terms of commuter density and also to test the scalability of VOMA. The travel

[5] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page. The reason for using 2016 data instead of more recent one is that 2016 data includes the commuters' GPS coordinates of pick-up and drop-off locations, which allows us to estimate the traveling distance and time so that we can compute feasible matching solutions and commuters' utility functions

distances between all points are approximated as Euclidean distances between the geographical coordinates which can be obtained from the GPS coordinates in the data. To compute travel times, we assume a driving speed of $30mph$ (we are considering an urban area). According to the American Automobile Association, the operational cost of a personal vehicle ranges from \$0.39 to \$0.86 per mile. Based on this statistic, a driver's per mile detour cost α_{d_i} is randomly drawn from a uniform distribution in the $[\$0.3 - \$0.8]$ range. For each rider r_i, the delay tolerance rate λ_{r_i} is randomly drawn from a uniform distribution in the $[0.1 - 0.8]$ range. This range is chosen based on the customer survey results presented in (Buchholz, Doval, Kastl, Matějka, & Salz, 2020). For the sake of simplicity, we select three personal attributes: age, gender and smoking tendency to formulate the matching utilities of commuters. They are selected because they are the top-three attributes that influence the commuters' ride-sharing behaviors as indicated in (Cui et al., 2021). Each attribute can be decomposed into distinct classes as shown in Table 4.3. For each testing group, we randomly assign the registered and preferred classes of each attribute to the commuters.

Table 4.2: Number of drivers and riders in different time windows

Testing group	ΔT	Number of drivers	Number of riders
1	2min	65	67
2	5min	177	130
3	10min	354	337
4	15min	458	442
5	20min	682	663
6	25min	831	785
7	30min	1402	1496

Table 4.3: Personal attributes and classified classes used in the computation study

Attribute	Class
Age	1 = Young adults; 2 = Middle-aged adults and 3 = Older adults
Gender	1 = Male, 2 = Female
Smoking tendency	1 = Smoking, 2 = Non-smoking

Table 4.4: Performance comparisons between CPLEX, VOMA and the greedy mechanism in terms of social welfare (SW) value, efficiency and computational time for 7 testing groups. We set a 20-hour time limit for CPLEX and "–" marks settings where CPLEX does not terminate within the time limit.

	Approach							
Testing	**CPLEX**		**VOMA**		**Greedy mechanism**			
Group	SW value	Running time (s)	SW value	Running time (s)	SW value	Running time (s)	eff_{VOMA}	eff_{GREEDY}
1	23.35	0.85	22.8	0.23	17.75	0.1	97.6%	76%
2	47.37	3.51	44.1	1.42	36	0.15	93.1%	76%
3	157	168	142	3.42	110	0.5	90.5%	70%
4	248	392	221.2	3.56	173.3	1.43	89.2%	69.8%
5	360	1380	318	5.82	226	1.9	88.4%	62.8%
6	412	5700	357	6.24	222	3.8	86.7%	54%
7	-	-	650	16.5	404	6.76	-	-
Average	-	-	-	-	-	-	90.9%	68.1%

4.6.3 Performance evaluation

In this subsection, we evaluate the performance of VOMA by comparing its solutions with the optimal solutions computed by the CPLEX solver and those computed by a greedy matching mechanism in which drivers and riders only accept improving solutions during each negotiation round. Same as in (Lang et al., 2016), there is no quota rule adopted in this mechanism. The optimal solutions are obtained by solving the matching model formulated in Section III using the CPLEX solver, assuming the operator knows the utilities of drivers and riders. Clearly, this approach is not practical in the community ride-sharing setting, but it serves as an upper bound for the efficiency that could be achieved by any optimization algorithm. VOMA and the greedy mechanism are implemented in Python and run on a GPU. As a single experimental execution involves a certain degree of randomness, we reduce the variance in the results by averaging them over 1000 runs.

Table 4.4 shows the comparison results in terms of the social welfare value and the efficiency of solutions computed by CPLEX, VOMA and the greedy mechanism over Groups 1-7. Here, VOMA is configured with $\varrho = 30$, $T = 1000k$, $\psi = 32$ and the $Threshold$ parameter is set to the number of commuters. The greedy mechanism is configured with $T = 1000k$ and $\psi = 32$. It is observed from Table 4.4 that VOMA computes high-quality matching solutions in terms of social welfare values and efficiency across all testing groups even through the mechanism does not have

Table 4.5: Performance comparisons between CPLEX, VOMA and the greedy mechanism in terms of travel distance savings for 7 testing groups. We set a 20-hour time limit for CPLEX and "–" marks settings where CPLEX does not terminate within the time limit.

	VMT savings (in *Mile*)				
Testing Group	**CPLEX**	**VOMA**	**Greedy mechanism**	Gap_{VOMA}	Gap_{GREEDY}
1	31.5	30.6	26.3	2.8%	16.5%
2	77	73	62	5.2%	19.4%
3	244	219	163.7	10.2%	23.9%
4	392	342	247.5	12.7%	36.9%
5	595	526	345	11.5%	42%
6	633.4	547	358	13.6%	43.4%
7	-	1035	676	-	-
Average	-	-	-	9.3%	30.3%

the commuters' private utilities. Taking group 1 as an example, the matching solution computed by VOMA achieves on average 97.6% efficiency compared to the optimal matching solution computed by CPLEX (regarded as 100% efficiency). Meanwhile, the greedy mechanism only achieves on average 76% efficiency, which is 28% lower than that of VOMA. This makes sense as commuters in the greedy mechanism only accept improved matching solutions compared with the initial one. In this case, the distributed search is equivalent to a local search like hill climbing, which will easily get stuck in local optima. However, in VOMA, commuters have to follow the quota rule which forces drivers and riders to sometimes accept worse matching solutions. This simulated-annealing based guidance can efficiently explore the search space in order to find (near-) optimal matching solutions. Note that we set a 20-hour time limit for CPLEX in the experiments. CPLEX does not terminate within the time limit for group 7, which explains the absence of social welfare values for CPLEX over group 7 in Table 4.4 and Table 4.5.

From column 8 of Table III we also observe that the average efficiency of matching solutions computed by VOMA decreases when the number of commuters grows in the system. This can be explained by the fact that more commuters are more difficult to coordinate and there is a larger conflict of interest due to desperate preferences (Lang et al., 2016). Also the expansion of solution space makes the optimization more challenging for the proposed mechanism and the optimality gap slightly increases under the same negotiation round. However, even for testing group 6 with 831

Table 4.6: Impact of T on the quality of matching solutions

Testing Group	Performance	Negotiation rounds							
		1k	5k	10k	50k	100k	500k	1000k	5000k
1	Efficiency	94.2%	96.4%	96.8%	97.1%	97.4%	97.7%	98%	98.1%
	VMT savings	30.3	30.4	30.5	30.6	30.6	30.6	30.8	30.8
	Running time (s)	0.0002	0.0012	0.002	0.011	0.023	0.11	0.23	1.15
2	Efficiency	78.2%	85.3%	87.4%	88.7%	88.9%	90.8%	91%	92%
	VMT savings	54.3	58	59	59.8	60	61.1	62	63
	Running time (s)	0.0014	0.007	0.014	0.06	0.14	0.646	1.42	5.278
3	Efficiency	71.3%	80.3%	84.1%	88.5%	89%	89.8%	90.5%	91.3%
	VMT savings	167.8	186.7	193.9	204.3	205.8	207.8	208.5	212.6
	Running time (s)	0.003	0.016	0.033	0.169	0.342	1.78	3.42	10.4
4	Efficiency	66.5%	73.8%	77.8%	85.1%	87.1%	88.7%	89.2%	89.6%
	VMT savings	244.7	267.9	280.7	308	315.2	319.5	319.9	324
	Running time (s)	0.0031	0.0158	0.033	0.16	0.324	1.644	3.32	15.2
5	Efficiency	59.7%	66.7%	69.7%	79.2%	80.5%	87.9%	88.3%	89%
	VMT savings	332	361	378	426	443.8	467.3	471	476.3
	Running time (s)	0.0057	0.029	0.058	0.29	0.58	2.93	5.82	29.4
6	Efficiency	53.2%	61.9%	63.1%	75.8%	79.6%	85.5%	86.5%	87.5%
	VMT savings	358	392.7	414.5	477.5	502	540	547	552.3
	Running time (s)	0.006	0.031	0.06	0.315	0.63	3.13	6.24	31.4
7	Efficiency	48%	53.2%	56%	65.7%	70.4%	79.5%	82.3%	84.3%
	VMT savings	642.2	685.2	715	824	881	1001.5	1035	1075
	Running time (s)	0.016	0.08	0.16	0.82	1.66	8.23	16.5	82.4

drivers and 785 riders, VOMA achieves on average 86.7% efficiency compared with CPLEX. Moreover, in contrast to CPLEX which requires the commuters' complete utilities to compute an optimal matching solution, VOMA does not require that information which maintains the commuters' privacy.

The computation times to generate the corresponding matching solutions are also provided in Table 4.4. It is shown that our largest testing group, which contains 1402 drivers and 1496 riders, requires less than 20 seconds to be solved by VOMA. This level of responsiveness is acceptable for a ride-sharing system. In addition, the computation times of VOMA can also be significantly reduced by decreasing the number of negotiation rounds which we will show in Table V. It is important to note that although the greedy mechanism has better computation time performance, its efficiency performance is much worse than that of VOMA. As shown in Table III, for group 7, the social welfare value achieved by VOMA is 60% higher than that of the greedy mechanism. Given the trade-off between solution quality and computation time, VOMA is clearly favorable for a community ride-sharing setting.

The VMT savings of matching solutions computed by CPLEX, VOMA and the greedy mechanism are shown in Table 4.5. The columns Gap_{VOMA} and Gap_{GREEDY} indicate the relative gaps between the matching solutions computed by CPLEX and VOMA and the matching solutions computed by CPLEX and the greedy mechanism, respectively. As observed in this table, the VMT savings obtained by VOMA is close to the optimum, and the average gap is less than 10%. While for the greedy mechanism, the average gap to the optimum solution is more than 30%. For group 7, VOMA can save 1035 miles on average which significantly outperforms the solution computed by the greedy mechanism (676 miles). Note that the distance savings is contributed by an average of 2898 commuters and the daily number of ride-sharing participants of the city exceeds 12000, which may create substantial environmental and economic benefits. The comparison results in terms of social welfare, efficiency and VMT savings indicate that the proposed approach is suitable and performs well in ride-sharing systems. In fact, as shown in Table 4.6, we can further boost the performance of VOMA by increasing the number of negotiation rounds.

Table 4.6 shows the performance of VOMA in terms of the efficiency and the VMT savings with different numbers of negotiation rounds T over testing groups 1-7. It is shown that the efficiency of VOMA increases significantly across all testing groups when increasing the value of T. Also, the amount of VMT savings increases accordingly, as expected. Considering testing group 5 as an example, the efficiency of VOMA increases from 59.7% to 89% and the VMT savings increase from 332 miles to more than 476 miles as the number of negotiation rounds increases from $1k$ to $5000k$. This makes sense since more negotiation rounds will generate more matching solutions, which in turn increases the possibility of finding a better matching solution for each testing group. However the cost comes with the high performance is more computation time. As shown in Table V, negotiating with a greater number of rounds significantly increases the computation time, particularly for the testing groups with a large number of drivers and riders. In spite of that, our largest testing group, which contains 1420 drivers and 1496 riders, requires less than 2 minutes to be solved with $5000k$ rounds. Moreover, VOMA is an anytime algorithm. If so required, VOMA can stop at any pre-defined time limit with an improved matching solution over the initial one.

4.7 Summary

In this chapter, we propose a voting-based negotiation mechanism, serving as a coordination protocol to match drivers and riders in community ride-sharing. This mechanism is a distributed implementation of the simulated annealing meta-heuristic. It is designed to achieve high-quality solutions, while preserving commuters' privacy. The experiment results show that the proposed matching mechanism achieves on average more than 90% efficiency compared with that of the optimal solution. In addition, the vehicle miles traveled savings produced by the proposed matching mechanism are close to the optimal results. Moreover, it scales well to larger problem instances, which makes it a potentially viable matching algorithm candidate for large-scale community ride sharing platforms.

Chapter 5

BM-DDPG: An Integrated Dispatching Framework for Ride-Hailing Systems[1]

In the previous chapters, we have proposed two approaches to deal with the game theoretic behaviors of the participants in two-sided markets. In this chapter, we focus on the rider demand uncertainty in the dynamic market environment. Towards this, we propose a dispatching framework which integrates batched matching with data-driven guidance for driver-rider matching and apply this approach to a Uber-like ride-hailing market.

This chapter proposes an integrated dispatching framework for matching drivers with riders in ride-hailing systems. The goal is to compute matching solutions that maximize social welfare and benefit both sides of the market, such that the sustainable growth of the ride-hailing system is ensured. The proposed framework integrates data-driven proactive guidance strategies with batched matching optimization to increase social welfare, improve matching rate and reduce rider wait time. Proactive guidance strategies are computed by leveraging short-term demand forecasts based on historical data. Taken the resulting guidance strategies as inputs, the batched matching algorithm computes optimal bipartite matching between drivers and riders in a batch. Using New York City taxi data from 2016 March 1st to March 31st as input, we conduct a numerical study to evaluate the performance of the proposed framework and compare it with existing approaches in the literature.

[1] Gao, J., Li, X., Wang, C., Huang, X. (2021). BM-DDPG: An Integrated Dispatching Framework for Ride-Hailing Systems. IEEE Transactions on Intelligent Transportation Systems. (Early Access). DOI: 10.1109/TITS.2021.3106243

Our results show that the proposed framework improves social welfare for up to 50%. It also increases the matching rate by an average of 20% and reduces the average rider wait time by over 15%. This implies a strong potential for the proposed dispatching framework to improve service quality in ride-hailing systems.

5.1 Introduction

Batched matching has been adopted by ride-hailing companies such as Uber, Lyft and DiDi to reduce overall rider wait times (Agency, 2019; Z. Xu et al., 2018). Unlike *immediate matching* (H. Wang & Yang, 2019), a commonly used practice that allocates an open driver to the nearest waiting rider on a first-come-first-served basis, batched matching aggregates all riders within a predetermined time window, i.e., a batch, and minimizes overall rider wait times (Korolko, Woodard, Yan, & Zhu, 2018; Yan, Zhu, Korolko, & Woodard, 2020).

Batched matching makes better use of driver supply by consolidating rider requests. However, it does not address the supply-demand imbalance issue which adversely impacts both the drivers and the riders. For instance, suppose the number of rider requests in a specific region surges within a short period of time. To meet the excess demand, the ride-hailing system would call in open drivers from neighboring regions. These out-of-region drivers then have to travel extra miles to pick up the riders, resulting in longer wait time for the rider. As the drivers become open after dropping off the riders, the present ride-hailing system would not provide any *direct guidance* as to how they should be relocated in expectation of future rider requests. Open drivers will then have to predict the demand based on their own experience and travel to the regions where they believe the next rider requests will likely appear. If the demand falls short of their expectations or too many drivers show up in the same region, the drivers may end up waiting for a long period of time without any rider assignment, following which they may need to re-evaluate the situation and move to other regions. This *ad-hoc* prediction and *uncoordinated* ride hunting behavior lead to excessive operational costs for the drivers and negative social externalities, such as urban congestion and increased emissions (Yuan, Zheng, Zhang, & Xie, 2012).

In response to this issue, ride-hailing systems have adopted *indirect guidance* strategies such as *surge pricing* to attract drivers toward regions where demand outstrips supply. This strategy may reduce expected wait times M. K. Chen (2016). However, with surge pricing, drivers can only estimate the demand based on the level of surge prices. In addition, the relocation decisions among the drivers are not coordinated. In many cases, surge-chasing drivers may oversupply some regions while exacerbating supply shortages in other regions. These uncoordinated driver movements may impose longer overall wait times for the riders across regions and extra idle driving costs for the drivers (He & Shin, 2019; Rosenblat & Stark, 2016).

To address these issues, we propose an enhanced dispatching framework for ride-hailing systems by integrating *direct* driver guidance strategies with batched matching. The integrated framework consists of two modules, namely, *Data-driven Proactive Guidance (DDPG)* and *Batched Matching (BM)*. DDPG computes optimal open driver guidance strategies based on predicted rider demand and open driver supply for various regions. The demand prediction is generated through a machine learning algorithm based on historical data. The open driver supply is obtained from the matching solutions computed by BM during previous batching windows. The optimal guidance strategies are computed by solving a mixed integer program with the objectives of minimizing the open driver idle driving cost and the supply-demand gap of the region. Taking the computed guidance strategies from DDPG as inputs, BM computes the optimal bipartite matching among open drivers and waiting riders in the batch.

The proposed framework contributes to the literature by integrating batched matching models with data-driven proactive guidance strategies, thus yielding lower wait times for the riders, shorter idle driving distance for the drivers and higher matching rate in a batching window. The rest of the chapter is organized as follows. Section 5.2 reviews related works and positions the proposed approach in the literature. Section 5.3 describes the batched matching problem and its mathematical formulation. Section 5.4 presents the integrated dispatching framework followed by a computational study to evaluate the effectiveness of the proposed framework in Section 5.5. Section 5.6 concludes the chapter.

5.2 Related work

A key operational challenge for ride-hailing systems, as for any on-demand transportation systems, is the problem of *matching*, which corresponds to the process of finding a proper driver to serve a rider request. In recent years, a considerable amount of research has been devoted to the design and analysis of matching approaches for ride-hailing systems. In a recent comprehensive review (H. Wang & Yang, 2019), matching approaches are classified into two categories: *greedy matching* and *batched-matching*. Greedy matching algorithms, such as those proposed in (G. Feng et al., 2020; Lee et al., 2004), find the nearest driver or the shortest-travel-time driver for each individual rider request. Although these methods are easy to implement and manage, they are myopic in the sense that they prioritize immediate individual rider satisfaction over efficient resource utilization across many riders, which jeopardizes rider satisfaction at a larger scale. As an alternative, batched matching strives to accommodate the needs of more riders at a time by optimizing the matching among a group of drivers and riders accumulated in a pre-determined batching window. For the benefit of riders, Seow *et al.* (Seow, Dang, & Lee, 2009) propose a multi-agent taxi dispatch structure with the objective of minimizing total rider wait times. This approach concurrently dispatches multiple taxis to pick up a set of riders and allows taxis to exchange their booking assignments. In addition to reducing rider wait times, approaches are also designed to maximize the matching rate in a batched matching solution. A multi-stage stochastic optimization formulation is proposed by Lowalekar *et al.* (Lowalekar, Varakantham, & Jaillet, 2018) to maximize the number of matched requests in a batch with the consideration of future rider demand. Zhang *et al.* (L. Zhang et al., 2017) propose a combinatorial optimization model to solve the order dispatch (matching) problem at Didi Chuxing. Their objective is also to maximize the matching rates. For the benefit of drivers, Zhan *et al.* (Zhan, Qian, & Ukkusuri, 2016) propose two matching algorithms namely optimal matching and trip integration to minimize drivers' idling driving time and distance across all open drivers in the system. The computation results show that these algorithms could find the optimal strategy that minimizes the cost of empty trips, and the number of taxis required to serve all observed trips.

The above-mentioned approaches compute good system-wide matching results. However, they are more reactive in a way that they only respond to the current locations of drivers at the time of matching without providing proactive guidance for the drivers before the rider demands are realized. As reported in (Niu, Wang, Zhou, & Zhou, 2019), when there is no appropriate guidance, drivers are confused with where to go for next rider requests. Some of them would waste much time roaming on the road; while others just wait at locations where they believe they can be dispatched to the next rider quickly.

Stand alone guidance strategies for ride-hailing systems have also been proposed in the literature. These strategies can be categorized into reactive and proactive. Reactive strategies attempt to rebalance vehicles within the system after rider demands in different regions are realized. Typical examples include Fluid model based rebalancing (Braverman, Dai, Liu, & Ying, 2019), Markov transition based rebalancing (Volkov, Aslam, & Rus, 2012) and queueing-theory based rebalancing (R. Zhang & Pavone, 2016). Guidance strategies that anticipate riders' demands before they are realized are proactive strategies. These strategies relocate the vehicles to the regions where high demand is expected.

In the context of Autonomous Mobility-on-Demand systems, a Model Prediction Control algorithm that leverages short term forecast of customer demand is proposed to rebalance the operations of autonomous MoD fleet (Iglesias et al., 2018). The future customer demand is predicted by a forecasting model based on Long Short-Term Memory neural networks. In a similar fashion, Tsao *et al.* (Tsao, Iglesias, & Pavone, 2018) propose a stochastic model-predictive control algorithm to relocate vehicles by leveraging the short-term travel demand forecasts. The aforementioned guidance strategies limit themselves to guidance decisions only. They do not consider the interaction between guidance decisions and matching decisions in a ride-hailing system. In a more realistic setting, the time constraints imposed by batched matching windows will impact the number of available open drivers who can be guided to a certain region. Also, the driver supply of a certain region during a future batching window is also impacted by the matching decisions made before that window. In addition, existing matching approaches only consider benefiting one side of the market, e.g., improving the service quality to riders as in (Lowalekar et al., 2018; Seow et al., 2009; L. Zhang et al., 2017) or reducing drivers' idle driving costs as in (Zhan et al., 2016). This one-sided objective is not suitable for today's ride-hailing systems which primarily operate in the context of the sharing economy. In the proposed framework, we maximize the social welfare of both drivers and riders, which is a suitable objective for designing two-sided matching markets for the sharing economy.

Table 5.1: Summary of existing works on matching problem in ride-hailing systems

Reference	Objective	Matching strategy	Guidance strategy	Note
Seow et al. (2009)	Minimize total rider wait times	Batched matching		Reactive matching No guidance
Lowalekar et al. (2018), L. Zhang et al. (2017)	Maximize the system matching rate	Batched matching		Reactive matching No guidance
Zhan et al. (2016)	Minimize total drivers' driving distance and time	Batched matching		Reactive matching No guidance
Braverman et al. (2019), Volkov et al. (2012), R. Zhang and Pavone (2016)	Minimize supply and demand gap		Reactive guidance	No matching Reactive guidance
Iglesias et al. (2018), Tsao et al. (2018)	Minimize guidance cost		Proactive guidance	No interaction between matching and guidance

5.3 The batched matching problem

In general, batched matching is an optimization problem which assigns drivers to rider requests collected in a batching window such that some social, economic and service quality objective functions are maximized or minimized relative to a set of feasible assignments. In this section, we formulate a batched matching problem which is particularly relevant to on-demand ride-hailing companies in the context of sharing economy with the objective of maximizing the social welfare of both drivers and riders. The social welfare maximization objective of our approach is suitable to ensure long-term growth of ride-hailing systems as it focuses on the benefits of both drivers and riders.

5.3.1 System overview

We consider a typical ride-hailing system with batched matching. As shown in Fig. 5.1, the system consists of a set of drivers, a set of riders and a service operator. Drivers' vehicles are equipped with sensing tools and are connected with the operator via a mobile network. The operator can

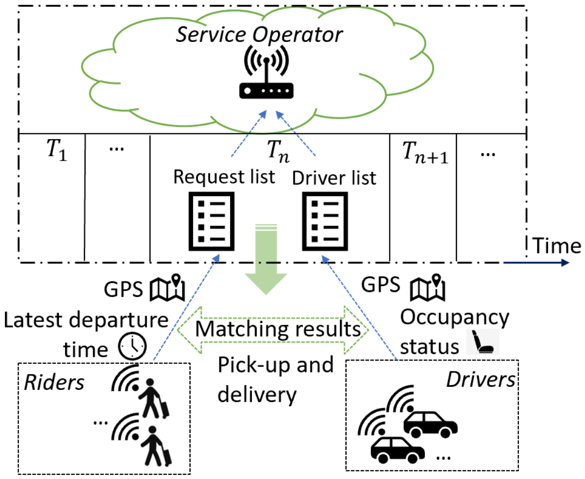


Figure 5.1: Overview of batched matching in a ride-hailing system.

monitor each taxi's geographical coordinates and occupancy status. On the other hand, riders dynamically enter the system and send their ride requests to the operator through mobile devices. The ride request of a rider contains his or her pick-up, drop-off locations as well as the latest departure time. The operator groups such requests within a pre-defined batching window, for example T_n in Fig. 5.1. At the end of the batching window, the operator runs a batched matching algorithm to compute matching results and sends those results to both drivers and riders. After receiving the matching results, drivers are on their way to pick up assigned riders.

5.3.2 Problem formulation

We consider a rather general setting in which the time of a day is divided into a set of batching windows of fixed size ΔT, denoted as $\{T_1, T_2, ..., T_n, ...\}$, where $T_n = [t_n, t_n + \Delta T]$ refers to the nth batching window.

Let $\mathcal{J}^{T_n}$ denote the set of riders in batching window T_n. The ride request r_j of a rider $j \in \mathcal{J}^{T_n}$ is a five-tuple $(ID_j, o_j, w_j, t_j, \bar{l}t_j)$, where ID_j is the ride request identification number, o_j and w_j are the pick-up and drop-off locations. t_j is the time when rider j sends out his or her request, $t_n \leq t_j \leq t_n + \Delta T$. $\bar{l}t_j$ denotes rider j's latest departure time, $t_n + \Delta T \leq \bar{l}t_j$. Here we assume that a rider will leave the ride-hailing system and switch to other alternatives if he or she is not picked up before his or her latest departure time.

Table 5.2: Description of notations

Notation	Description
T_n	A batching window
t_n	The start time of batching window T_n
ΔT	The size of a batching window
$\mathcal{J}^{T_n}$	A set of riders, indexed by j
$\mathcal{I}^{T_n}$	A set of open drivers, indexed by i
$\mathcal{M}$	A set of point of interests, indexed by m
l_i	Location of driver i
o_j, w_j	Pick-up and drop-off locations of rider j
t_j, $\bar{l}t_j$	Request time and latest departure time of rider j
$d_{o,w}$	Euclidean distance between two locations o and w
$\frac{1}{\gamma}$	Driving speed
$U^d_{i,j}$	Utility of driver i when matching with rider j
$U^r_{i,j}$	Utility of rider j when matching with driver i
$W^r_{i,j}$	Wait time of rider j when matching with driver i
$\hat{r}^{T_n}$	Predicted rider demand within a batching window
$z^{T_n}_{i,j}$	$z^{T_n}_{i,j} = 1$ if driver i is matched with rider j $z^{T_n}_{i,j} = 0$ otherwise
$x^{t_n}_{i,m}$	$x^{t_n}_{i,m} = 1$ if driver i is guided to point m $x^{t_n}_{i,m} = 0$ otherwise
λ_j	Delay tolerate rate of rider j
β_i	Driving cost per mile of driver i
γ_j	Utility coefficient of rider j for the ride quality measure
w	Weight coefficient
m, α	Base fare and ride fare per mile
H	A large positive constant
$WT(T_n)$	Average wait time of riders in a batching window
$NP(T_n)$	Net profit of drivers in a batching window
$MR(T_n)$	Matching rate of the ride-hailing system in a batching window

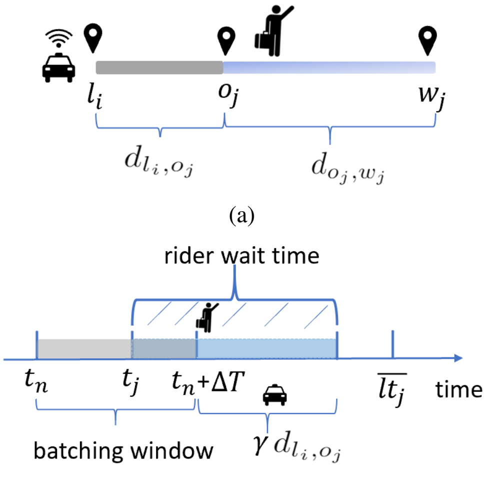
Let $\mathcal{I}^{T_n}$ denote the set of open drivers at the same batching window T_n. Let ID_i be the identification number of driver $i \in \mathcal{I}^{T_n}$. The location of open driver i at the end of batching window T_n is denoted as l_i. We assume all drivers operate with the same fixed speed. Let γ be the time each driver used to travel per *mile*. Thus the speed is $\frac{1}{\gamma}$ *mile* per hour.

As mentioned previously, the objective of the proposed matching framework is to maximize the social welfare of both drivers and riders, which is the sum of their utilities. We define a driver's utility of serving a rider request as the *net profit* that can be obtained from the service. Similar definition of a driver's utility is also adopted in (G. Gao, Xiao, & Zhao, 2016; H. Wang & Yang, 2019). This definition is suitable for drivers in ride-hailing systems as they are freelancers, not paid by company salaries. Their incomes are the profits obtained through providing transportation services to the riders. Let d_{l_i,o_j} and d_{o_j,w_j} denote the distance from location l_i to o_j and from location o_j to w_j in *miles* respectively. The utility of open driver i being assigned to rider j, denoted as $U^d_{i,j}$, is defined as:

$$U^d_{i,j} := m + \alpha d_{o_j,w_j} - \beta_i(d_{o_j,w_j} + d_{l_i,o_j}) \tag{15}$$

where m is the base fare and α is the ride fare per *mile*. Both of them are predetermined by the ride-hailing system. β_i is driver i's driving cost per *mile* which consists of operating cost (including, e.g., fuel, maintenance, repair and tires) per *mile* and ownership cost (including, e.g., depreciation, insurance, license and taxes) per *mile* (Association, 2019). As shown in Fig. 2 (a), a driver needs to travel d_{l_i,o_j} without being paid in order to pick up the rider. Therefore, the total fare paid to the driver is $m + \alpha d_{o_j,w_j}$ and the total cost of driver i for serving rider j is $\beta_i(d_{o_j,w_j} + d_{l_i,o_j})$.

On the other hand, we define a rider's utility as his or her level of satisfaction given the service provided. We assume the general ride fare in a ride-hailing system is fixed. Therefore, the level of satisfaction is largely determined by two factors, namely *wait time* and *ride quality*. As shown in Fig. 2 (b), the wait time of a rider j being matched with a driver i, denoted as $W^r_{i,j}$, can be computed by $t_n + \Delta T - t_j + \gamma d_{l_i,o_j}$, where $t_n + \Delta T - t_j$ is the time difference between the rider submitting his or her request and the operator executing the matching algorithm; and $\gamma d_{l_i,o_j}$ is the time needed for driver i to pick up rider j.



(a)

(b)

Figure 5.2: Location illustration and distance calculation of drivers and riders.

Ride quality can be difficult to measure as it depends on a rider's personal preferences. In this work, we use a common factor which is the physical condition of the vehicle to measure ride quality. This measurement is grounded on the rider rating statistics from ride-hailing platforms like Uber and Lyft. Their data indicates that riders' satisfaction levels are closely related to the physical conditions of the vehicles (Uber, 2020b) (Lyft, 2020). In general, a more high-end and well-maintained vehicle is commonly perceived to be more comfortable and in good condition. These vehicles incur higher ownership and maintenance costs which can be measured by β_i. Thus the utility of a rider j being matched with a driver i, denoted as $U^r_{i,j}$, is defined as:

$$U^r_{i,j} := \gamma_j \beta_i - \lambda_j W^r_{i,j} \tag{16}$$

where $\gamma_j \beta_i$ represents the rider's preference value generated from the ride quality. γ_j is the utility coefficient of rider j for the ride quality measure. λ_j represents the delay tolerate rate of rider j. With the increase of the wait time, a rider's satisfaction level decreases accordingly. We note that γ_j and λ_j are subjective to each individual rider. Accurately evaluating γ_j and λ_j is beyond the scope of this work. One can use machine learning algorithms to learn them from individual rider's rating data in ride-hailing platforms. Although simplified, we believe the current definitions of drivers' and riders' utilities are sufficient in terms of demonstrating our contributions.

After defining the utility functions of drivers and riders, in what follows, we introduce the decision variables, objective function and constraints of the batched matching problem. We define a binary matrix $Z^{T_n} \in \{0, 1\}$ as the matching matrix, where $z_{i,j}^{T_n} = 1$ if driver i is assigned to pick-up rider j, and $z_{i,j}^{T_n} = 0$ otherwise.

The social welfare maximizing objective function[2] of the batched matching problem is defined as follows:

$$\sum_{i \in \mathcal{I}^{T_n}} \sum_{j \in \mathcal{J}^{T_n}} z_{i,j}^{T_n} \left(U_{i,j}^{d} + U_{i,j}^{r} \right) \tag{17}$$

There are also constraints to be considered. If a driver is matched with a rider, the driver's cost of serving the rider should be less than the fare paid by the system. This means the net profit of a driver when matches to a rider should be positive. This constraint is captured by:

$$\begin{aligned} m + \alpha d_{o_j,w_j} + (1 - z_{i,j}^{T_n})H \geq \beta_i (d_{o_j,w_j} + d_{l_i,o_j}) \\ \forall i \in \mathcal{I}^{T_n}, \forall j \in \mathcal{J}^{T_n} \end{aligned} \tag{18}$$

where H is a large positive constant which is used for the linearization of the logical constraint "if" (Rardin & Rardin, 1998). Given rider's latest departure time, if a driver i is matched with a rider j, we must guarantee that the pick-up time should be no later than the rider's latest departure time. This time constraint is represented by:

$$\gamma d_{l_i,o_j} + t_n + \Delta T \leq l\bar{t}_j + (1 - z_{i,j}^{T_n})H \; \forall i \in \mathcal{I}^{T_n}, \forall j \in \mathcal{J}^{T_n} \tag{19}$$

where $\gamma d_{l_i,o_j}$ is driver i's driving time from his or her current location l_i to the rider's pick-up location o_j.

In addition to that, each driver i is allowed to serve at most one rider and each rider j can only

[2] In microeconomic theory, social-welfare is defined as the aggregated utilities of market participants. In this work, the social welfare of the ride-hailing system is defined as the aggregated utilities of drivers and riders. Similar definitions are also adopted in other mobility on demand systems proposed in the literature (J. Gao, Wong, Wang, & Yu, 2021; Karamanis, Anastasiadis, Angeloudis, & Stettler, 2020).

be served by at most one driver, which result in the following two constraints:

$$\sum_{j \in \mathcal{J}^{T_n}} z_{i,j}^{T_n} \leq 1 \; \forall i \in \mathcal{I}^{T_n} \tag{20}$$

$$\sum_{i \in \mathcal{I}^{T_n}} z_{i,j}^{T_n} \leq 1 \; \forall j \in \mathcal{J}^{T_n} \tag{21}$$

The inequalities (20) and (21) imply that the ride-hailing solutions are not compulsory to meet all requests. Given the previously defined decision variables, constraints and the objective function, the batched matching problem for each batching window T_n can be put together as the following integer linear programming model.

$$\begin{aligned} &\text{maximize} \quad (17) \\ &\text{subject to} \\ &(18), (19), (20), (21), \\ &z_{i,j}^{T_n} = \{0, 1\} \qquad\qquad \forall i \in \mathcal{I}^{T_n}, \forall j \in \mathcal{J}^{T_n}. \end{aligned} \tag{22}$$

The above defined batched matching model considers the benefits of both drivers and riders, which ensures the continuous growth of ride-hailing systems in the context of the sharing economy. However, as presented in the proposed framework (Section IV) and demonstrated in the numerical study (Section V), the social welfare of matching solutions can be considerably improved by integrating data-driven proactive guidance strategies into batched matching. In the rest of the chapter, we design and evaluate the integrated dispatching framework to match drivers and riders in ride-hailing systems.

5.4 The integrated dispatching framework

In this section, we present the integrated dispatching framework which consists of two modules: data-driven proactive guidance (DDPG) and batched matching (BM). As shown in Fig. 5.3, DDPG combines mathematical optimization with machine learning to compute open driver guidance strategies. Taken the guidance strategies computed by DDPG as inputs, BM computes social welfare maximization matching solutions. The execution sequence of the demand prediction, guidance strategy computation and batched matching is also illustrated in Fig. 5.3. At the start of batching window T_n, DDPG predicts rider demand for T_n based on historical demand data from previous batching windows for the region and computes optimal guidance for open drivers who are close to the region and can reach the region by the end of T_n. After the rider demand is realized, at the end of T_n, the matching algorithm in BM will compute an optimal matching solution and notify the drivers and riders. Assigned drivers will then on their way to pick up riders. We note that for the purpose of clearly demonstrating the impact of proactive guidance to batched matching, in this chapter, we narrow our scope to one region and one batching window. The proposed framework can certainly be extended to concurrently optimize batched matching solutions across multiple regions and multiple batching windows using rolling horizon decision making approaches. We plan to present those extensions in our future publications. In the following, we begin by delving into details of DDPG and then outline the data-driven batched matching algorithm.

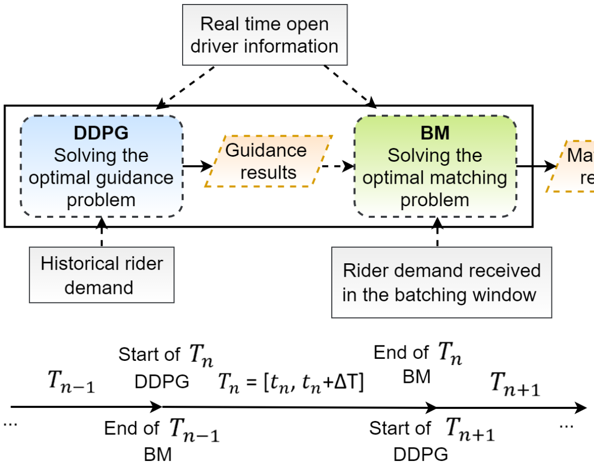

Real time open driver information
DDPG
Solving the optimal guidance problem
Guidance results
BM
Solving the optimal matching problem
Matching results
Historical rider demand
Rider demand received in the batching window
Start of T_n
DDPG
T_{n-1}
T_n = [t_n, t_n+ΔT]
End of T_n
BM
T_{n+1}
...
End of T_{n-1}
BM
Start of T_{n+1}
DDPG
...

Figure 5.3: Overview of the integrated dispatching framework.

5.4.1 Data-driven proactive guidance

In DDPG, the rider demand in each region within a batching window is predicted using a machine learning algorithm based on the rider demand time-series data. At a time point t_n, DDPG needs to predict how many riders will emerge during the future time period $[t_n + \Delta T]$. To do so, time series forecasting models can be used to predict the future demand based upon the previously observed demand. In addition to predicting a single demand value such as in the cases of the time-varying Poisson model (Y. Wang et al., 2019) and the auto-regressive integrated moving average model (ARIMA) (Moreira-Matias, Gama, Ferreira, Mendes-Moreira, & Damas, 2013), we can also estimate the probability distribution of rider demand by using a non-parametric learning approach (J. Gao, Li, Wang, & Huang, n.d.). While various forecasting models can be plugged into our framework, we would not elaborate more on the model selection since it is data dependent. One can select a forecasting model based on the particular characteristics of their demand data. We refer the reader to (Zhao, Tarkoma, Liu, & Vo, 2016) for a recent review on these models in the context of urban mobility. In our numerical study (Section 5.5), we use ARIMA which is widely used in transportation demand forecasting for DDPG.

Given the predicted rider demand and the real time open driver information including their GPS locations and availability status, an integer program model is formulated to compute the optimal guidance strategies for open drivers to balance the supply and demand of a region and, at the same time, minimize the total idle driving costs of the relocated drivers. Let k^{t_n} be the number of open drivers at the region at time t_n. Let p^{T_n} be the number of drivers traveling to the region expected to arrive within batching window T_n. This number is largely determined by the BM in the previous batching windows and can be obtained from the real time sensing information. Hence, the initial supply of the region can be computed by $k^{t_n} + p^{T_n}$. Drivers are guided to a set of point of interests (POI), denoted by $\mathcal{M}$, in the region. Each point $m \in \mathcal{M}$ represents a specific GPS location. Let $\mathcal{I}^{t_n}$ be the set of open drivers outside the region at time t_n. Let $\hat{r}^{T_n}$ be the number of predicted rider demand of the region within the batching window T_n. The rest of the notations used to describe DDPG are defined in the previous section.

We represent the guidance results using a guidance strategy matrix denoted as $X^{t_n} \in \{0, 1\}$, where $x^{t_n}_{i,m} = 1$ if driver $i \in \mathcal{I}^{t_n}$ is guided to point $m \in \mathcal{M}$ at time t_n and $x^{t_n}_{i,m} = 0$ otherwise. DDPG needs to make sure that if a driver is guided to point m to satisfy the demand accumulated within T_n, the driver should be able to arrive at point m before the end of T_n. This constraint is captured by:

$$\gamma d_{i,m} \leq (1 - x_{i,m}^{t_n})H + \Delta T \; \forall i \in \mathcal{I}^{t_n}, \forall m \in \mathcal{M} \tag{23}$$

where $\gamma d_{i,m}$ is driver i's driving time from his or her current location to the location of point m. H is a large positive constant and ΔT is the length of a batching window. Besides, each driver can be guided to at most one point, which results in the following constraint:

$$\sum_{m \in \mathcal{M}} x_{i,m}^{t_n} \leq 1 \; \forall i \in \mathcal{I}^{t_n} \tag{24}$$

Moreover, we aim to design a guidance strategy such that the total number of rider requests (i.e., demand) and the total number open drivers (i.e., supply) within the region are balanced. This can be captured by the following constraint:

$$\sum_{i \in \mathcal{I}^{t_n}} \sum_{m \in \mathcal{M}} x_{i,m}^{t_n} + k^{t_n} + p^{T_n} = \hat{r}^{T_n} \tag{25}$$

where $k^{t_n} + p^{T_n}$ is the initial supply of the region in batching window T_n and $\hat{r}^{T_n}$ is the predicted rider demand within the time window. However, Equation (25) can be too strict if used as a constraint, and there may be no feasible solutions satisfying (25). This is because decision variables are integer matrix, and open drivers' driving speed is limited that they may not be able to arrive at the region before the end of batching window. To that end, we convert the constraint (25) into a soft constraint by introducing a supply-demand mismatch penalty function to narrow the gap between supply and demand within the region and one objective of the DDPG is to minimize the following supply-demand gap function:

$$\sum_{i \in \mathcal{I}^{t_n}} \sum_{m \in \mathcal{M}} \| x_{i,m}^{t_n} + k^{t_n} + p^{T_n} - \hat{r}^{T_n} \| \tag{26}$$

Guiding the open driver from his or her current location to the point introduces a cost, since the driver receives no trip fares and shall pay a certain amount of operating fee. Thus, another objective of DDPG is to minimize the total idle driving costs of all guided open drivers. This cost is assumed to be proportional to the driving distance. Let $d_{i,m}$ be the approximated idle driving distance from the current location of a driver i to point m. The total idle driving costs of the guided open drivers is denoted by:

$$\sum_{i \in \mathcal{I}^{t_n}} \sum_{m \in \mathcal{M}} \beta_i d_{i,m} x_{i,m}^{t_n} \tag{27}$$

where β_i is the driving cost per *mile* of driver i as defined in Section III.

Since there exists a trade-off between two objective functions, we define a weighted sum of (26) and (27) as the objective function of the proactive guidance, represented as:

$$\sum_{i \in \mathcal{I}^{t_n}} \sum_{m \in \mathcal{M}} (\beta_i d_{i,m} x_{i,m}^{t_n} + w \| x_{i,m}^{t_n} + k^{t_n} + p^{T_n} - \hat{r}_{T_n} \|) \tag{28}$$

where w is the weight coefficient to measure the trade-off between two objectives. Given the above defined decision variables, constraints and objective, the optimized guidance strategy can be generated by computing the following binary integer programming model.

$$\begin{aligned} &\text{minimize} \quad (28) \\ &\text{subject to} \\ &(23), (24), \\ &x_{i,m}^{t_n} = \{0, 1\} \ \forall i \in \mathcal{I}^{t_n}, \forall m \in \mathcal{M} \end{aligned} \tag{29}$$

5.4.2 The integrated dispatching framework

The proposed dispatching framework integrates BM with the above designed DDPG to compute matching solutions that maximize the social welfare of drivers and riders. The guidance and matching decisions are interrelated in the sense that the supply of open drivers is determined by the matching decisions computed by BM in the previous matching windows and the number of guided drivers is restricted by the width of the batching window. In addition, which driver could be guided to the region is impacted by the BM since the objective of DDPG is in alignment with the social welfare maximization goal of BM. The details of the proposed framework for one batching window

$[t_n, t_n + \Delta T]$ is described in Algorithm 5.

Algorithm 5 Batched matching with data-driven proactive guidance

```
Require: Batching window: $T_n = [t_n, t_n + \Delta T]$; $\forall i \in \mathcal{I}^{T_n}$;$\forall j \in \mathcal{J}^{T_n}$; $\forall m \in \mathcal{M}$
Initialization: The value of $w$, the real time indicator $t$.
while $t = t_n$ do
    Update the sensor information for current GPS locations of open drivers and occupied drivers
    Estimate the number of drivers $p^{T_n}$ that will arrive at the region before time $t_n + \Delta T$
    Predict the rider demand $\hat{r}^{T_n}$ in the batching window $T_n$
    $X^{t_n} \leftarrow$Solve the optimized guidance problem (29)
    Send guidance strategies to open drivers according to the optimal solution $X^{t_n}$
    if $t = t_n + \Delta T$ then
        Update the sensor information for the current locations of open drivers in the region
        Collect rider requests in the region
        $Z^{T_n} \leftarrow$ Solve the batched matching problem (41)
        Send matching results to drivers and riders according to the optimal solution $Z^{T_n}$
    end if
end while
Ensure: Stored sensor data, guidance solutions and matching solutions.
```

At time t_n, the operator queries the system to obtain information about the drivers' occupancy status and GPS locations. At the same time, the service operator predicts the future rider demand $\hat{r}^{T_n}$ for the next ΔT time steps based on the historical demand data. The operator then computes the optimal guidance strategy X^{t_n} by solving a mixed integer linear program (formulated in Subsection 5.4.1, optimization model (29)) and sends the guidance results to the open drivers. At the end of the batching window, i.e., at time $t_n + \Delta T$, the rider requests are collected and the operator matches the riders and drivers by computing the optimal matching model formulated in optimization model (41) (see Section II.A for details). Open drivers immediately start picking up riders according to the matching results Z^{T_n}. This process is repeated for every batch of requests. If there are riders and drivers that are not matched in this batch, they are carried over and re-solved in the next batching window. The matching model (8) and the guidance model (15) can be solved using general integer programming optimization packages or dedicated algorithms such as the Hungarian algorithm (Kuhn, 1955).

5.5 Numerical study

In this section, we conduct a numerical study to verify the performance of the proposed framework in terms of four evaluation metrics which are important to ride-hailing system operators, participating riders and drivers. First, we present the definitions of the evaluation metrics. Next, we describe the dataset used to conduct the experiments. Finally, we evaluate the framework by comparing its performance with alternative batched-matching approaches.

5.5.1 Evaluation Metrics

We propose four evaluation metrics: social welfare, rider average wait time, driver net profit and matching rate. From the perspective of system operator, social welfare and matching rate are measures of system efficiency which are well suited for the design of long-term stable ride-hailing systems, while average wait time and net profit measure the satisfaction levels of riders and drivers respectively.

- **Social welfare** of the ride-hailing system during a batching window T_n is defined in Equation (17).

- **Average wait time** of riders during a batching window T_n is defined as the ratio of the total rider wait time to the number of successfully matched riders within the batching window,

$$WT(T_n) = \frac{\sum_{j \in Z^{T_n}} W^r_{i,j}}{\sum_{i,j \in Z^{T_n}} z_{i,j}} \tag{30}$$

- **Net profit** of drivers during a batching window T_n is defined as the total driver net profits within the batching window,

$$NP(T_n) = \sum_{i \in Z^{T_n}} U^d_{i,j} \tag{31}$$

- **Matching rate** of the ride hailing system during a batching window T_n is defined as the ratio of the number of successfully matched rider-driver pairs to the number of total rider requests within the batching window,

$$MR(T_n) = \frac{\sum_{i \in \mathcal{I}^{T_n}, j \in \mathcal{J}^{T_n}} z_{i,j}}{R_{T_n}} \tag{32}$$

where R_{T_n} is the number of rider requests within a batching window T_n.

5.5.2 Description of the Data

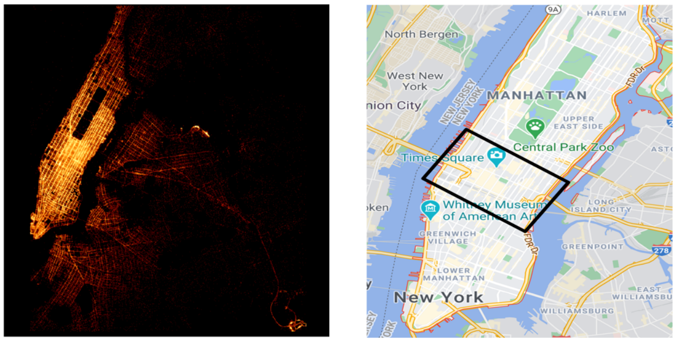


Figure 5.4: A heatmap visualization of the rider demand density computed using the 2016 New York City green taxi trip data. The lighter the region the higher the daily demand density. As shown on the map, the region of Midtown Manhattan has the highest demand density compared with other regions.

We demonstrate the performance of the proposed matching framework using 2016 New York City taxi trip data (OpenData, 2016). The reason for using 2016 data instead of more recent one is that 2016 data includes the riders' GPS coordinates of pick-up and drop-off locations, which allows us to estimate the traveling distance and time so that we can compute the driver's and rider's utilities. In addition, the data contains the pick-up and drop-off timestamps of each trip which are also needed by the proposed framework. By generating a heatmap visualization of the data, as shown in Fig. 5.4, we identified Midtown Manhattan as the region with the highest average rider demand density. We use the demand data from this region for performance evaluation as more data points can be obtained to feed the machine learning algorithm. In total, 362579 trip requests during the one month period from 00:00 on March 1st to 23:59 on March 31st, 2016 are extracted from the data set for the purpose of evaluating the performance of the proposed batched matching framework. The width of batching window ΔT is set to 10 minutes[3], which results in 144 consecutive batching windows in each day. Since the data set does not contain the initial locations of drivers, we randomly assign popular drop-off locations in the regions surrounding Midtown Manhattan as open drivers' initial locations. At the start of each batching window, 300 drivers are assigned to those drop-off locations. The rider demand of the next batching window in Midtown Manhattan is predicted using ARIMA based on the demand data of previous batching windows. The traveling distance between two locations is approximated as Euclidean distance between the geographical coordinates which can be obtained from the GPS data in the data set. To calculate the driving time between each pair of driver and rider, we assume that all drivers drive at a constant speed of $30mph$.

According to the American Automobile Association, the operational cost of a personal vehicle ranges from \$0.39 to \$0.86 per mile (Association et al., 2011). Based on this statistics, in the following experiments, the driver's per mile operational cost β_i is randomly drawn from a uniform distribution in the range of \$0.4-\$0.9. According to the current rates of NYC UberX services (Uber, 2020a), we set $m = \$2.55$ and $\alpha = \$1.8$ per mile for all drivers. For each rider j, the delay tolerance rate λ_j is randomly drawn from a uniform distribution in the range of 0.1-0.8. This range is chosen based on the analysis of the customer survey results presented in (Buchholz et al., 2020). For the sake of simplicity, we assume the preference values generated from the ride quality are the same across riders. The weight coefficient w controls the trade-off between the driver idle travel cost and the system level supply-demand mismatch cost. In the following experiments, we choose $w = 0.8$, which maintains a good balance between these two cost components. The parameter setting of the proposed approach is shown in TABLE 6.4. Note that the values of the above mentioned parameters can be easily adjusted according to the specific operating condition of a ride-hailing system. All numerical experiments are coded in Python and tested on a 2.2 GHz Intel i7 laptop with 16GB RAM. The optimization models (41) and (29) are solved using Gurobi 8.1[4]. For the 4320 problem

[3] In this experiment, the width of the batching window is suitable for the demand density of the region given by the 2016 New York taxi data. It can certainly be narrowed once recent higher demand density data such as Uber's demand data becomes available to us. In that case, the value of $W^r_{i,j}$ will be reduced while still maintaining a meaningful number of riders in a window to ensure matching quality.

[4] https://www.gurobi.com/academia/academic-program-and-licenses/

instances generated from the data, the average time required to solve a single instance of model (41) was 14.2s. For model (29), that was 15.6s. No instance required more than 48 seconds to be solved.

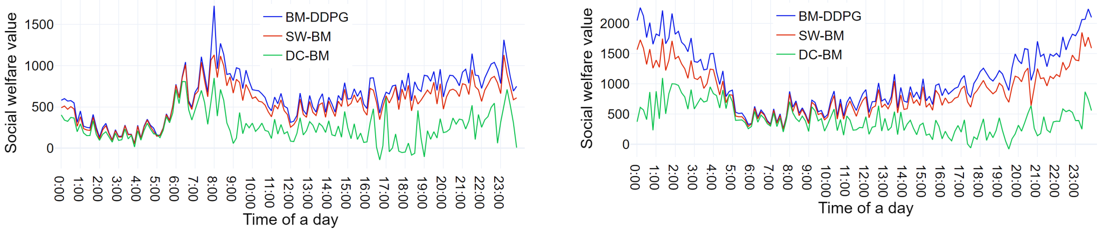
Table 5.3: Parameter setting of the BM-DDPG framework.

Parameter	Variable	Value
The width of batching window	ΔT	10 minutes
Number of open drivers	$\lvert\mathcal{I}^{T_n}\rvert$	300
Driving speed	$\frac{1}{\gamma}$	30 mph
Vehicle operational cost	β_i	\$0.4/mile-\$0.9/mile
Base fare	m	\$2.55/mile
Ride fare per mile	α	\$1.8/mile
Rider delay tolerance rate	λ_j	0.1-0.8
Weight coefficient	w	0.8

5.5.3 Performance Evaluation

Since we do not find an existing framework in the literature that integrates batched matching with proactive guidance, we show the benefits of the proposed framework by comparing its results with those generated by batched matching models without proactive guidance. In light of this evaluation strategy, we picked two batched matching models with different optimization objectives as the benchmark models for the purpose of comparison. The first model maximizes social welfare, which is the same as in the model described in Section III. The second model minimizes drivers' idling driving cost. We call these two models Social Welfare maximization Batched Matching (SW-BM) and Drivers' Cost minimization Batched Matching (DC-BM), respectively. SW-BM fits well in the context of sharing economy, while DC-BM reflects the traditional view of emphasizing on cost reduction in operation. Nevertheless, as argued in (Zhan et al., 2016), the matching assignments that minimize drivers' costs, e.g. idling driving time and empty trip distance, will also take care of the needs of riders since those assignments usually result in shorter wait times.

Fig. 5.5 shows the comparison results in terms of the average value of social welfare over the proposed framework, SW-BM and DC-BM. The results generated by the proposed framework is labeled as BM-DDPG (batched matching with data-driven proactive guidance). The results of SW-BM and DC-BM are optimal solutions obtained using Gurobi solver. A 24-hour day is divided into 144 10-minute batched windows. The social welfare value for each batched window is averaged



(a) weekday

(b) weekend day

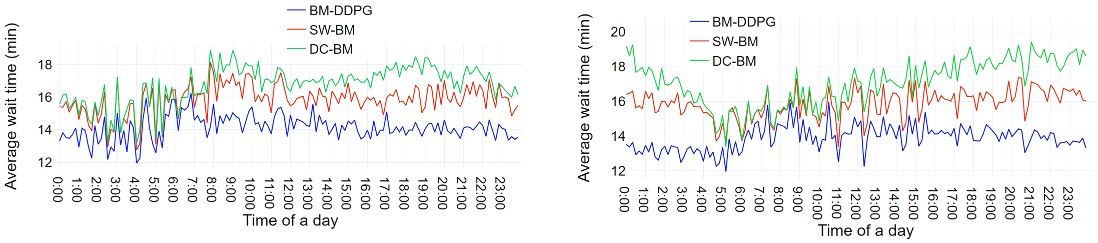
Figure 5.5: Comparison of average social welfare values between SW-BM, DC-BM and BM-DDPG in different batching windows of a day.

over 23 weekdays (Fig. 5.5-(a)) and 8 weekend days (Fig. 5.5-(b)).

It is shown that the performance improvement brought by the proposed framework (BM-DDPG) is consistent in all batching windows with gains in social welfare ranging from 5% to 20% compared with SW-BM and on average more than 50% compared with DC-BM. It is clear from the results that, by integrating proactive guidance with batched-matching, the proposed framework boosts the efficiency of the system in terms of social welfare value. This is because the proactive guidance module of the framework can strategically position the drivers before hand so that they can reduce their idling driving distance and meet the rider demand in a more timely manner.

It is also shown in Fig. 5.5 that DC-BM generates the lowest social welfare value which is lower than that of SW-BM and much lower than that of BM-DDPG. This is because DC-BM's objective function does not include riders' utilities in terms of wait times and ride quality. As mentioned previously, DC-BM can be seen as an approximation to social welfare maximization. While comparing DC-BM with SW-BM and BM-DDPG on social welfare maximization might not seem to be fare since DC-BM has a different objective function, the comparison results indicate that existing operation cost focused batched matching approaches may not be suitable for today's sharing economy based ride-hailing systems.

The computational results in terms of rider wait times are shown in Fig. 5.6. Same as in Fig. 5.5, the results are also averaged over 23 weekdays (Fig. 5.6(a)) and 8 weekend days (Fig. 5.6(b)). As expected, with the integrated dispatching framework, the average rider wait times are reduced in all batching windows compared with DC-BM and SW-BM. Specifically, BM-DDPG provides a 25% reduction in average rider wait times over DC-BM and a 20% reduction over SW-BM. It can be



(a) weekday

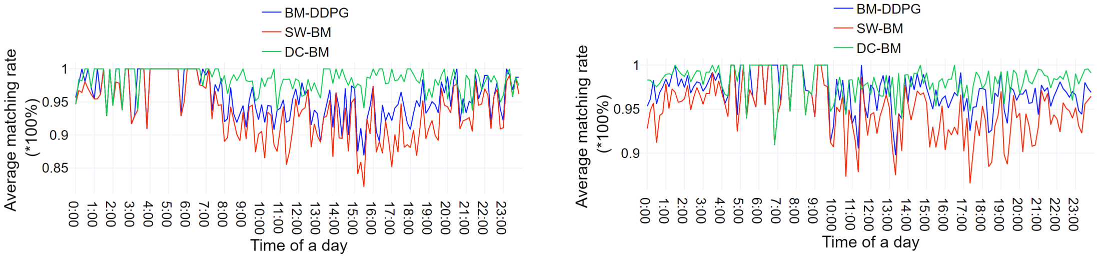
(b) weekend day

Figure 5.6: Comparison of average wait times of riders between SW-BM, DC-BM and BM-DDPG in different batching windows in a day.

observed that this reduction can be even more significant in the batching windows such as rush hours during weekdays and around midnight hours during weekend days when the rider demand density in a window is high. This is due to the fact that to satisfy high rider demand, without any proactive guidance, SW-BM and DC-BM have to call in more drivers far from the region to pick-up riders, which results in much longer rider wait times. In addition to driver wait times, the proposed framework also performs better than SW-BM and DC-BM in terms of driver net profit. As shown in TABLE 5.4, BM-DDPG commands more than 15% of driver net profit compared with SW-BM and DC-BM during week days and more than 10% during weekend days. In addition to rider

Table 5.4: Driver net profit value of BM-DDPG, SW-BM and DC-BM averaged over 23 weekdays and 8 weekend days.

Approach	Avg.weekday	Avg.weekend day
BM-DDPG	1122	1718
SW-BM	908	1472
DC-BM	965	1609



(a) weekday

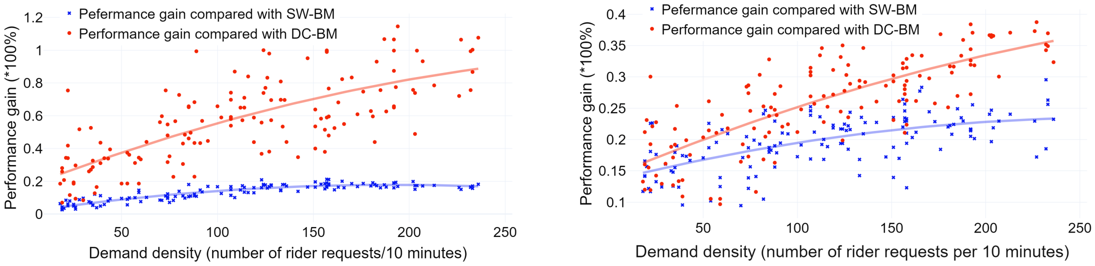
(b) weekend day

Figure 5.7: Comparison of average matching rates between SW-BM, DC-BM and BM-DDPG in different batching windows of a day.

wait times and driver net profit, matching rate is also an important metric to measure the efficiency of a batched matching system. Fig. 5.7 shows the comparison of average matching rate among BM-DDPG, SW-BM and DC-BM. As can be seen, BM-DDPG outperforms SW-BM in all batching windows of both a week day and a weekend day by 20% in average matching rate. This is because, in the proposed framework, BM-DDPG strategically guides a subset of drivers to the region before matching starts, which provides more drivers to meet riders' latest departure time constraints. The increase of feasible drivers will in turn improve the matching rate for the batching window. It is also worth noting that, DC-BM achieves a slightly higher matching rate compared with BM-DDPG. This makes sense since DC-BM only minimizes drivers' costs without considering riders' utilities such as waiting times and ride quality. Unlike the social welfare maximization BM-DDPG, minimizing rider wait times is not part of the objective function in DC-BM. So DC-BM can match more drivers at the cost of longer wait times for riders.

Moreover, it can be observed that, across all three approaches, the average matching rates during quiet hours, e.g. 23:00 pm to 5:00 am the next day during weekdays and 4:00 am to 9:00 am during weekend days, are higher than those during peak hours, e.g. 7:00 am-9:00 am and 17:00 pm-19:00 pm during weekdays. This is because the driver/request ratio is high during quiet hours. However, with the increase in the number of rider requests during peak hours, the average matching rates of all three approaches decrease due to the change of driver/request ratio. Nevertheless, during the peak hours with high demand density, BM-DDPG yields a significantly higher matching rate than SW-BM does, which further highlights the benefits of integrating proactive guidance with batched matching.

We also study the impact of demand density on the performance of the proposed approach. Here, demand density is defined as the number of rider requests aggregated in each 10-minute batching window for a specific area. The results are shown in Fig. 5.8. It is observed that the performance gains brought by the proposed approach in terms of social welfare and average wait time compared with SW-BM and DC-BM are more significant with high demand density than that with low demand density. This observation can be an indication of the performance of the proposed approach in more general problem settings. For example, it is reasonable to predict that the proposed approach could perform better in high density urban areas than in low density rural areas.



(a) Social welfare

(b) Average wait time

Figure 5.8: The performance gains compared with SW-BM and DC-BM in terms of social welfare and average wait time of BM-DDPG under different demand densities. Here, the performance gain is defined as the difference between the value computed by BM-DDPG and SW-BM (or DC-BM) divided by the value computed by BM-DDPG.

5.6 Summary

In this chapter, we propose a dispatching framework which integrates batched matching with data-driven proactive guidance to match riders and drivers in a ride-hailing system. By guiding drivers to the right positions before the matching starts, the proposed framework has the ability to compute matching solutions that maximize the social welfare of both drivers and riders.

Chapter 6

A pricing mechanism design for ride-hailing systems in the presence of driver acceptance uncertainty

Freelance drivers of ride-hailing systems often strategically accept and reject ride requests based on their projection of the profitability of the assigned rides. This driver acceptance uncertainty is mainly caused by the flat-rate pricing and blind acceptance rule adopted by most ride-hailing platforms. High driver rejection rates have negative impacts on the ride-hailing systems in terms of service quality and matching efficiency. In this chapter, we propose a pricing mechanism aiming at improving drivers' average ride acceptance rate by offering personalized payments computed based on the characteristics of individual rides and the estimated acceptance probabilities of the drivers. The proposed ride pricing mechanism is designed based on behavioral economics and stochastic optimization models. First, a binary choice model is formulated to analyze and estimate the ride acceptance probabilities of drivers. Taking the estimated acceptance probability as input, a stochastic optimization model is formulated to compute payments tailored to each of the drivers. Numerical experiments are conducted using Ride Austin trip data collected from June 2016 to April 2017. We evaluate the performance of the proposed pricing mechanism by comparing it with flat-rate pricing and surge pricing strategies. Our results show that the proposed pricing mechanism improves the drivers' average acceptance rate by an average of 80%. It also significantly increases the platform's expected profit and matching rate. This implies a substantial potential for the proposed pricing mechanism to improve service reliability in ride-hailing systems.

6.1 Introduction

Serving as independent contractors, drivers in ride-hailing systems are free to decide whether to accept or reject the ride requests assigned by the platform (**?**). For an accepted ride, the platform computes the driver's payment based on a base fare and the rider's trip distance and trip time. This payment scheme is called *flat rate payment*, which is currently adopted by ride-hailing platforms such as Uber and Lyft (Garg & Nazerzadeh, 2021; Lyft, 2021).

The flat rate payment strategy is easy to implement and manage. However, it only takes a rider's trip distance and trip time into consideration. Other important factors that affect the cost of the driver, such as the distance traveled and the time spent to pick up a rider are not considered. In addition, the *blind ride acceptance rule*[1] currently adopted by most of the platforms makes the driver's estimation of the profitability of a ride before accepting it a guessing game. When receiving a ride request, the driver can only see the pick-up location, not the drop-off location (Rosenblat, 2018; Rosenblat & Stark, 2016). Therefore, they do not have sufficient information to calculate if the expected payment of a ride can cover their cost, let alone make a reasonable profit. To avoid unprofitable rides, a driver would speculate on the profits of ride requests based on their past ride assignment experience with the platform and cherry-pick the rides that, they believe, are more lucrative (Ashkrof, de Almeida Correia, Cats, & van Arem, 2020; Marshall, 2020).

Despite its intended purpose of combating ride cherry-picking behaviors, the blind ride acceptance rule limits drivers' ability to estimate the costs of assigned rides, which may lead to more rejections due to the risk-averse behavior of drivers. Ride-hailing platforms using this rule see, on average, a 50% driver rejection rate (K. Xu, Sun, Liu, & Wang, 2018). It gets even higher if the bind ride acceptance rule is not enforced. For example, from April to November 2020, to promote Prop

[1] Under the blind ride acceptance rule, the assigned riders' drop-off locations and the trip fares are not shown to the drivers before they accept the rides (Rosenblat & Stark, 2016).

22[2], Uber allowed drivers in California to see the ride drop-off locations before accepting them. However, since drivers could see their profitability beforehand, the flat rate payment strategy demotivated drivers from taking on higher cost rides and the rejection rate went up to 80% (Cradeur, 2019).

Another payment strategy of ride-hailing platforms that contributes to the high rejection rate is *surge pricing*. It is intended to attract drivers toward regions where demand outstrips supply. When price surges in a region, drivers are motivated to relocate themselves to that region, hoping they will be assigned to a ride in the surge region and get paid by a higher rate. Once drivers have decided to relocate themselves to surge regions, they normally reject standard price requests along the way in anticipation of being assigned with a ride in the surge region, which will *indirectly* increase the drivers' rejection rate (Ashkrof et al., 2020). The high driver rejection rate experienced by the ride-hailing systems has far-reaching consequences. It increases riders' wait time, thus decreasing their satisfaction and may affect their retention. Meanwhile, it results in a low matching rate in the system, which negatively impacts its profitability and service quality.

In this chapter, we design a pricing mechanism to align the interests of the drivers and the platform, which leads to an overall better acceptance rate and platform profitability. We first build a binary choice model to estimate drivers' probabilities of accepting a ride based on the discrete choice analysis and the random utility theory. Taking the predicted acceptance probabilities as inputs, a stochastic optimization model is designed to compute payments for individual drivers. The pricing mechanism offers personalized payments for drivers based on the characteristics of the assigned rides and the individual acceptance probabilities of drivers. Compared with flat-rate and surge pricing strategies, the proposed approach significantly improves the system's average acceptance rate and the platform's expected profit. The remainder of the chapter is structured as follows. In Section 6.2, we review the literature related to our work. Section 6.3 describes the matching problem and its mathematical formulation. Section 6.4 presents the design of pricing mechanism followed by a numerical study to evaluate the performance of the proposed approach in Section 6.5. Section 6.6 concludes this chapter.

[2] California Proposition 22 (Prop 22), officially known as the "App-Based Drivers as Contractors and Labor Policies Initiative," defines app-based transportation and delivery drivers as "independent contractors" rather than "employees" (Cradeur, 2021).

6.2 Related work

Efficiently matching the demand and supply in ride-hailing systems is the core service provided by the platforms. While maximizing revenue is one of the important objectives of ride-hailing platforms, the quality of the service they provide is usually measured by matching rates, rider wait times and other criteria derived from both drivers and riders. Various optimization models have been proposed for assigning drivers to riders with the objective of maximizing service quality objective functions (Y. Feng, Niazadeh, & Saberi, 2021; G. Gao et al., 2016; J. Gao, Li, Wang, & Huang, 2021; Guo, Caros, & Zhao, 2021; H. Wang & Yang, 2019; Zhan et al., 2016). These models either do not take the impact of pricing into consideration as in (Y. Feng et al., 2021; Guo et al., 2021) or they usually assume static flat-rate pricing where drivers' compensation rate does not change across different regions or time periods as in (G. Gao et al., 2016; J. Gao, Li, et al., 2021; Zhan et al., 2016).

On the other hand, dynamic pricing (also known as surge pricing) strategies are also studied in the literature. In dynamic pricing, rates are "spatial-based" or "temporal-based" depending on the supply-demand condition of a particular region at the time period in question (Garg & Nazerzadeh, 2021; M. Yang & Xia, 2021). Compared with the flat rate pricing strategies, surge pricing, which dynamically determines the payments for drivers based on the supply-demand condition of a region, could achieve a higher system performance in terms of profitability and service quality (Cachon, Daniels, & Lobel, 2017). To maximize the platform's revenue, a spatial surge pricing strategy is designed in (Zha, Yin, & Xu, 2018) to relocate drivers so that the supply-demand gap will be mitigated. They assume that drivers in each zone are homogeneous in terms of ride requests. Integrated with surge pricing, Yang *et al.* (H. Yang, Shao, Wang, & Ye, 2020) propose a reward scheme to allow riders to pay an additional amount to a reward account during peak hours. Then riders use the balance in the reward account to compensate for their trips during off-peak hours. The authors show that the integrated approach in some cases improves rider utility, driver income, and platform revenue compared with approaches without reward schemes. Zha *et al.* (Zha, Yin, & Du, 2018) investigate the impact of surge pricing on the supply and demand gap using a bi-level optimization framework. The lower-level optimization model captures the drivers' choices of working hours under an equilibrium state, while the upper-level optimization model maximizes the total revenue through surge pricing, i.e., differentiated pricing across hours. A time-based surge pricing is computed by (Bai, So, Tang, Chen, & Wang, 2019) to maximize the profit of the platform as well as the social welfare of the system with the consideration of time and price sensitive customers.

In addition to analyzing the impact of surge pricing on revenue and service quality, some researchers focus on how surge pricing affects the relocation behaviors of drivers in ride-hailing systems. Castillo *et al.* (Castillo, Knoepfle, & Weyl, 2017) show that surge pricing is responsible for effectively relocating drivers during periods of high-demand thereby preventing them from engaging in 'wild goose chases' to pick up distant customers. Similarly, Guda *et al.* (Guda & Subramanian, 2019) consider the strategic interaction among drivers in their decisions to move between zones. They found that surge pricing can be profitable even in a zone where the supply of drivers exceeds demands. Considering the drivers' strategic relocation behaviors, Besbes *et al.* (Besbes, Castro, & Lobel, 2021) study how to set prices across locations in a city optimally and what the impact of those prices is on the strategic repositioning of drivers. In a similar fashion, Bimpikis *et al.* (Bai et al., 2019) investigate how surge pricing at different locations affects driver distributions. Jiang *et al.* (Jiang, Kong, & Zhang, 2020) conduct laboratory experiments to study drivers' relocation decisions with surge pricing. The authors observe that surge pricing must be combined with demand information sharing to be the most effective in incentivizing drivers to relocate to a demand surge region.

The above-mentioned matching models and dynamic pricing strategies are important tools for helping ride-hailing systems to achieve high service quality. However, they assume that drivers are fully compliant with the ride requests assigned by the platform, which is not realistic in many ride-hailing settings. In today's sharing economy-based ride-hailing systems, such as Uber and Lyft, drivers can strategically make their acceptance and rejection decisions to boost their personal benefits rather than system-wide service quality. Without addressing drivers' strategic ride acceptance behaviors, the existing pricing approaches encourage drivers' ride profitability speculation and cherry-picking, which results in a high driver acceptance uncertainty in the system. The proposed pricing mechanism fills the gap by computing optimal personalized payments that take ride details and drivers' acceptance uncertainty into account. As demonstrated in the numerical study

(Section 6.5), with the proposed pricing mechanism, all stakeholders, namely drivers, riders, and the platform, are better off compared with the flat rate and surge pricing strategies.

6.3 Matching in a ride-hailing system

Ride-hailing platforms match drivers and riders through a matching algorithm and send the matched ride information to the corresponding drivers. In this section, we formulate a matching problem in a ride-hailing system with the objective of maximizing the social welfare of both drivers and riders. The social welfare maximization objective of our approach is suitable to ensure long-term growth of ride-hailing systems as it focuses on the benefits of both drivers and riders.

6.3.1 The matching process

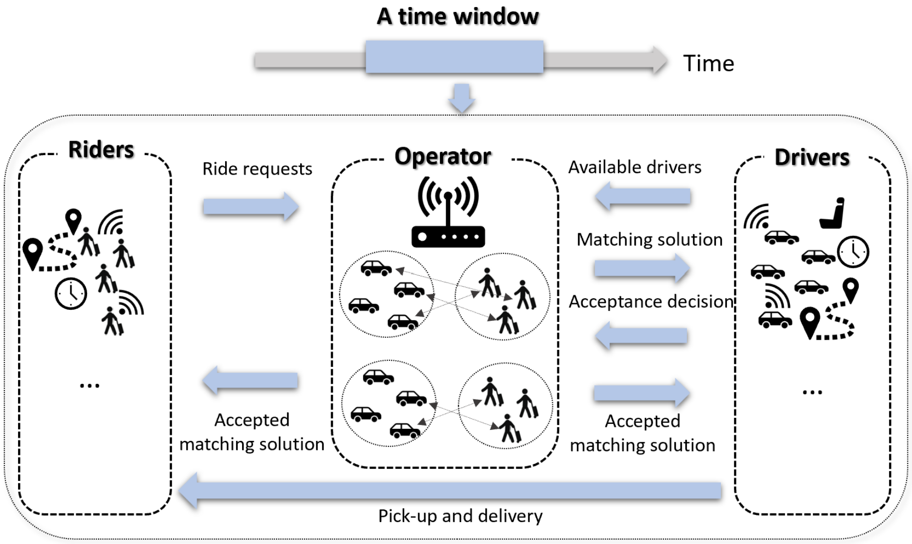


Figure 6.1: Overview of the matching process in a ride-hailing system.

In this work, we consider a typical matching in a ride-hailing system. As shown in Figure 6.1, the system consists of three types of stakeholders: drivers, riders and a service operator. Drivers' vehicles are equipped with sensing tools and are connected with the operator via a mobile network. The operator can monitor each vehicle's geographical coordinates and occupancy status. On the other side, riders send their ride requests to the operator through mobile devices. The service operator aggregates the available drivers and active ride requests within a time window and matches the drivers and riders through a matching algorithm to minimize the riders' wait times and the drivers' idle times between rides. The operator then sends the matching solution to the drivers. In the current ride-hailing system, drivers are given a few seconds (15 seconds in the Uber platform) to decide whether to accept or reject the assigned ride request (Ashkrof et al., 2020). Those drivers who accept the request need to pick up the corresponding riders at their pick-up locations within riders' feasible time intervals. If a driver rejects the request, the driver waits for the next possible ride request provided by the operator. The rejected riders will be re-matched in the following time period if they are still active.

6.3.2 Matching problem formulation

We consider a rather general setting where a day is divided into a set of time windows. A time window f, denoted by $[t_f^-, t_f^+]$ with the length $t_f^+ - t_f^-$, where t_f^- is the start time and t_f^+ is the end time of time window f, respectively.

Let $\mathcal{D}^f$ denote the set of available drivers in time window f. The time that driver $d_i \in \mathcal{D}^f$ is available in this time window is denoted as t_{d_i}, with $t_f^- \leq t_{d_i} \leq t_f^+$. The GPS location of driver $d_i \in \mathcal{D}^f$ at the end of the time window (at time t_f^+) is denoted as l_{d_i}. Each driver has a vehicle type such as *standard* or *luxury* (Austin, 2021). Let $\mathcal{M}$ be the set of vehicle types existing in the ride-hailing system. Let $m_{d_i} \in \mathcal{M}$ be the vehicle type of driver d_i.

Let $\mathcal{R}^f$ denote the set of riders in the same time window f. Each rider $r_j \in \mathcal{R}^f$ has a ride request denoted as $(t_{r_j}, o_{r_j}, w_{r_j}, et_{r_j}, lt_{r_j}, m_{r_j})$, where t_{r_j} is the time when rider r_j submits his or her request to the platform, with $t_f^- \leq t_{r_j} \leq t_f^+$. o_{r_j} and w_{r_j} are the pick-up and drop-off locations of rider r_j, respectively. et_{r_j} is the earliest departure time and lt_{r_j} is the latest departure time, with $t_{r_j} \leq et_{r_j} \leq lt_{r_j}$. Here, $[et_{r_j}, lt_{r_j}]$ defines the feasible time interval during which rider r_j should be picked up by a driver. m_{r_j} is the vehicle type requested by rider r_j[3]. In this work, we slightly abuse notations and use r_j to refer to both rider r_j and his or her ride request.

Given the available drivers and the ride requests within f, the objective of the matching problem is to maximize the social welfare of the platform, which is defined as the total preference of drivers

[3] Riders may request a specific vehicle type for travel, such as regular or SUV based on their individual needs. This service is provided by ride-hailing platforms such as Uber to enhance specialized transit (Uber, 2021).

and riders. Based on the microeconomic theory Mas-Colell et al. (1995), these preferences can be quantified by preference values that measure their satisfaction levels. We assume a driver's preference value of matching with a rider is largely determined by the *idle time* needed to pick up a rider. The idle time of a driver is defined as the wait time plus the en-route time to pick up a rider. It is also called unproductive or down time in the ride-hailing literature Zuniga-Garcia, Tec, Scott, Ruiz-Juri, and Machemehl (2020). The wait time can be computed by $dt_{d_i} - t_{d_i}$, where dt_{d_i} is driver d_i's departure time (the time driver d_i is started to pick up the rider) and t_{d_i} is d_i's avaible time. Let $t_{l_{d_i}, o_{r_j}}$ be the en-route time of d_i to pick up a rider r_j. The idle time of driver d_i, represented as $h^d_{r_j,d_i}$, can be computed as $h^d_{r_j,d_i} = dt_{d_i} - t_{d_i} + t_{l_{d_i}, o_{r_j}}$. For each driver, with the increase of total idle time, his or her satisfaction level decreases accordingly. Thus, the preference of a driver d_i when matching with a rider r_i can be quantified by a preference value function, defined as:

$$g^d_{d_i,r_j} := \beta_{d_i} \frac{1}{h^d_{r_j,d_i} + 1} \tag{33}$$

where $\beta_{d_i} > 0$ is driver d_i's cost coefficient for the idle time, which is subjective to each driver.

On the demand side, riders are impatient, they highly value their time and prefer to be matched to drivers with the shortest waiting time. Let $t_{l_{d_i}, o_{r_j}}$ be the travel time between driver's location l_{d_i} and rider's pick up location o_{r_j}. The wait time of rider r_j, represented as $w^r_{d_i,r_j}$, can be computed as $w^r_{d_i,r_j} = dt_{d_i} + t_{l_{d_i}, o_{r_j}} - et_{r_j}$, where $dt_{d_i} + t_{l_{d_i}, o_{r_j}}$ is the time needed for driver d_i to pick up rider r_j and et_{r_j} is r_j's earliest departure time. Thus the preference value of a rider r_j being matched with a driver d_i, denoted as $g^r_{d_i,r_j}$, is defined as:

$$g^r_{d_i,r_j} := \gamma_{r_j} \frac{1}{w^r_{r_j,d_i} + 1} \tag{34}$$

where $\gamma_{r_j} > 0$ denotes the cost of waiting per unit time for rider r_j. The value of γ_{r_j} is subjective to each individual rider. The preference value can be seen as how much the rider is satisfied with the waiting time $w^r_{r_j,d_i}$. With the increase of wait time, the rider's satisfaction level decreases accordingly.

After defining the preference value functions of drivers and riders, the social welfare maximization objective function in the matching problem is defined as follows:

$$\sum_{d_i \in \mathcal{D}^f} \sum_{r_j \in \mathcal{R}^f} (g^d_{d_i,r_j} + g^r_{d_i,r_j}) x_{d_i,r_j} \tag{35}$$

Here x_{d_i,r_j} is a binary variable that equals to 1 if driver d_i is matched with rider r_j, with respected to the following constraints:

(1) *Time constraint*: as given in Eqs. (36) and (37), a driver d_i is matched with a rider r_j only when the pick-up time $dt_{d_i} + t_{l_{d_i},o_{r_j}}$ within the rider's feasible time interval $[et_{r_j}, lt_{r_j}]$.

$$\begin{aligned} & dt_{d_i} + t_{l_{d_i},o_{r_j}} \geq et_{r_j} + (x_{d_i,r_j} - 1)H \\ & \forall d_i \in \mathcal{D}^f, \forall r_j \in \mathcal{R}^f \end{aligned} \tag{36}$$

$$\begin{aligned} & dt_{d_i} + t_{l_{d_i},o_{r_j}} \leq lt_{r_j} + (1 - x_{d_i,r_j})H \\ & \forall d_i \in \mathcal{D}^f, \forall r_j \in \mathcal{R}^f \end{aligned} \tag{37}$$

where H is a large positive constant which is used for the linearization of the logical constraint "if" [28].

(2) *Vehicle type constraint*: as given in Eq. (38), a driver is matched with a rider only if the assigned vehicle type is consistent with the rider's requested vehicle type.

$$x_{d_i,r_j} m_{d_i} = x_{d_i,r_j} m_{r_j} \quad \forall d_i \in \mathcal{D}^f, \forall r_j \in \mathcal{R}^f \tag{38}$$

(3) *Uniqueness constraint*: as given in Eqs. (39) and (40), a driver d_i can only be matched with one rider r_j, and vice versa.

$$\sum_{d_i \in \mathcal{D}^f} x_{d_i,r_j} \leq 1 \quad \forall r_j \in \mathcal{R}^f \tag{39}$$

$$\sum_{r_j \in \mathcal{R}^f} x_{d_i,r_j} \leq 1 \quad \forall d_i \in \mathcal{D}^f \tag{40}$$

Given the above defined decision variables, constraints and objective function, the optimal matching solutions can be computed by solving the following binary integer programming model.

$$\begin{aligned}
&\text{maximize} \quad (35) \\
&\text{subject to} \\
&(36), (37), (38), (39), (40), \\
&x_{d_i,r_j} \in \{0,1\} && \forall d_i \in \mathcal{D}^f, \forall r_j \in \mathcal{R}^f, \\
&dt_{d_i} \in \mathbb{R} && \forall d_i \in \mathcal{D}^f.
\end{aligned} \tag{41}$$

While the optimal matching solution maximizes the social welfare of all drivers and riders in the matching window, their materialization requires that drivers accept assigned rides. However, as mentioned before, due to the potential misalignment of the cost and payment of an assigned ride, the flat-rate pricing and blind acceptance rule currently adopted by ride-hailing systems do not help in terms of discouraging drivers' cherry-picking behaviors. In the next section, we design an individual ride-based pricing mechanism for addressing this issue.

6.4 The pricing mechanism

In this section, we design a pricing mechanism to incentivize drivers to accept the assigned ride requests. This mechanism consists of two components. First, given a matched ride, a binary choice model is formulated to estimate the driver's acceptance probability based on discrete choice analysis and random utility theory (Louviere, Hensher, & Swait, 2000; Train, 2009). Taking the predicted acceptance probabilities as inputs, a stochastic optimization model is designed to compute a set of payments for the drivers. The objective of the stochastic optimization model is to maximize the platform's expected profit, which is defined as the difference between the expected income and the expected payments. As shown in the numerical study (Section V), this objective makes a balanced payment decision that strives mostly for maximizing the platform's profit and improving the drivers'

Table 6.1: Description of notations

Notation	Description
f	A time window
t_f^-	Start time of time window f
t_f^+	End time of time window f
$\mathcal{D}^f$	A set of open drivers in f, indexed by d_i
$\mathcal{R}^f$	A set of riders in f, indexed by r_j
l_{d_i}	Location of driver d_i
b_{d_i}	Vehicle type of driver d_i
o_{r_j}, w_{r_j}	Pick-up and drop-off locations of rider r_j
lt_{r_j}	Latest departure time of rider r_j
b_{r_j}	Requested vehicle type of rider r_j
h_{r_j,d_i}^d	Idle time of driver d_i when matching with r_j
w_{d_i,r_j}^r	Wait time of rider r_j when matching with d_i
g_{d_i,r_j}^d	Preference value of d_i when matching with r_j
g_{d_i,r_j}^r	Preference value of r_j when matching with d_i
x_{d_i,r_j}	$x_{d_i,r_j} = 1$ if driver d_i is matched with rider r_j $x_{d_i,r_j} = 0$ otherwise
X^f	Matching solution of time window f
C_{d_i}	Choice set of d_i
$U_{d_i}^c$	Utility of d_i when choosing alternative c
$p_{d_i}^c$	Probability of d_i when choosing alternative c
s_{d_i}	Payment offered to d_i

average acceptance probability.

6.4.1 Driver acceptance probability estimation

The discrete choice theory is applied to analyze and predict a driver's binary choice between accept and reject of the assigned ride request. The theory uses the principle of utility maximization economic behavior in which a driver is modeled as a rational, self-interested agent who selects the alternative from a choice set with the highest utility (McFadden, 1981; Train, 2009). Let X^f be the matching solution generated from Section 6.3.2. Let $C_{d_i} = \{a_{d_i,r_j}, n_{d_i,r_j}\}$ be the *choice set* of driver d_i, where alternative a_{d_i,r_j} indicates that the driver d_i accepts the assigned ride r_j in X^f and alternative n represents that the driver rejects the assigned ride r_j in X^f. For ease of readability, we suppress the notations a_{d_i,r_j} and n_{d_i,r_j} as a and n in the following, unless otherwise specified.

Given the choice set, a utility maximization driver will choose the alternative that provides the highest utility. However, the utility of each driver is his or her private information, not known by the operator. To deal with this issue, we adopt random utility theory (Louviere et al., 2000) and formulate the utility functions of choosing alternatives a and n of each driver $d_i \in X^f$ as follows, respectively:

$$U^a_{d_i} = V^a_{d_i} + \epsilon^a_{d_i} \tag{42}$$

$$U^n_{d_i} = V^n_{d_i} + \epsilon^n_{d_i} \tag{43}$$

where functions $V^a_{d_i}$ and $V^n_{d_i}$ are the *systematic (or representative)* components of the utility of choosing alternatives a and n, respectively, which are known or could be estimated statistically by the operator. $\epsilon^a_{d_i}$ and $\epsilon^n_{d_i}$ are the stochastic *disturbances* (*or random components*) which are not observed by the operator, capturing the uncertainty. In the following we first define the systematic components of the utility functions (42) and (43). Then, we describe the formulation of the random components.

The systematic component $V^c_{d_i}$ of an alternative $c \in C_{d_i}$ is a function of *variables* of the alternative. Based on a recent ride-hailing drivers' behavior and preference survey paper (Shokoohyar, Sobhani, & Sobhani, 2020), these variables may include the travel time and travel distance to pick up the rider, the rider pick up location region[4], assigned rider's trip distance and trip time, the inertia comfort, and the *payment* offered by the operator. For the sake of simplicity, we assume the systematic utility function of a driver choosing an alternative $c \in C_{d_i}$ is a linear weighted combination of the variables. Let $\boldsymbol{m}^c_{d_i} \in \mathbb{R}^K$ be a vector which contains a set of variables of choosing an alternative $c \in C_{d_i}$, with $|\boldsymbol{m}^c_{d_i}| = K$. Therefore, the systematic utility functions of driver d_i choosing alternatives a and n can be defined as follows, respectively:

$$V^a_{d_i} = \sum_{k=1}^{K} \varrho_k m^a_{d_i,k} \tag{44}$$

[4] Drivers usually do not want to be matched with a rider in downtown areas especially in peak hours since this crowed area is hard to stand still or park, which has a high possibility of getting fined (Ashkrof et al., 2020).

$$V_{d_i}^n = \sum_{k=1}^{K} \varrho_k m_{d_i,k}^n \tag{45}$$

where $m_{d_i,k}^a$ and $m_{d_i,k}^n$ indicate the values of the kth variable in vectors $\boldsymbol{m}_{d_i}^a$ and $\boldsymbol{m}_{d_i}^n$, respectively. ϱ_k is the coefficient for variable k, representing the marginal impacts of the variable. The coefficients reflect the variables' contributions to the overall utility. Accurately evaluating these coefficients is beyond the scope of this paper. One can use the maximum likelihood method or machine learning algorithms to learn them from individual driver's acceptance data in ride-hailing platforms like the approach proposed in (Yu, Mo, Xie, Hu, & Chen, 2021). Note that the payment offered by the operator is a variable of the systematic utility function of a driver.

After defining the systematic components, we now introduce the specification of the random components. In this work, we assume the random components $\epsilon_{d_i}^a$ and $\epsilon_{d_i}^n$ are independent and identically distributed extreme values with zero mode and $\pi^2/6$ variance. In this case, the difference between disturbances $\epsilon_{d_i}^n - \epsilon_{d_i}^a$ follows standard logistic distribution, namely

$$F(\epsilon_{d_i}^n - \epsilon_{d_i}^a) = \frac{1}{1 + e^{-\mu(\epsilon_{d_i}^n - \epsilon_{d_i}^a)}} \tag{46}$$

Logit is a widely used discrete choice model. Its popularity is due to the fact that the formula for the choice probabilities takes a closed form and is readily interpretable (Train, 2009). Although this kind of modeling approach may seem to be restrictive, it is standard and widely used within the ride-hailing user behavior analysis literature (Urata et al., 2021; Yu et al., 2021).

Under the assumption that $\epsilon_{d_i}^n - \epsilon_{d_i}^a$ is logistically distributed, the probability that driver d_i

chooses alternative a, represented as $p^a_{d_i}$, can be computed as:

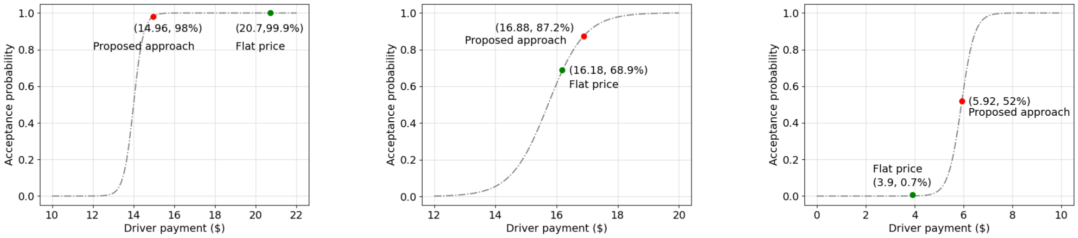
$$\begin{aligned}
p^a_{d_i} &= Pr(U^a_{d_i} \geq U^n_{d_i}) \\
&= Pr(\epsilon^n_{d_i} - \epsilon^a_{d_i} \leq V^a_{d_i} - V^n_{d_i}) \\
&= \int I(\epsilon^n_{d_i} - \epsilon^a_{d_i} \leq V^a_{d_i} - V^n_{d_i}) f(\epsilon_{d_i}) d\epsilon_{d_i} \\
&= \frac{1}{1 + e^{-(V^a_{d_i} - V^n_{d_i})}} \\
&= \frac{1}{1 + e^{\sum_{k=1}^{K}(\varrho_k m^n_{d_i,k} - \varrho_k m^n_{d_i,k})}}
\end{aligned} \tag{47}$$

where $I(\cdot)$ is the indicator function, equaling 1 when the difference between random components $\epsilon^n_{d_i} - \epsilon^a_{d_i}$ is less than or equal to the difference between systematic components $V^a_{d_i} - V^n_{d_i}$, and 0 otherwise. And, the probability that driver d_i chooses alternative n, denoted as $p^n_{d_i}$ is computed as $p^n_{d_i} = 1 - p^a_{d_i}$.

6.4.2 Personalized payment determination

Based on the above estimated choice probability of each driver, a stochastic optimization model is formulated to maximize the platform's expected profit. The expected profit of the platform is defined as the difference between the expected revenue and the expected payment. In the current practise, the revenue or the trip fare contains three parts: base fare, fare per mile and fare per minute (Austin, 2021; OpenData, 2016). Let $\bar{c}$ be the base fare. Let $d_{o_{r_j},w_{r_j}}$ be the trip distance in mile of rider r_j, where o_{r_j} and w_{r_j} are the origin and destination of rider r_j. The distance related trip fare is thus computed as $c^d d_{o_{r_j},w_{r_j}}$, where c^d is the fare per mile determined by the platform. Let $t_{o_{r_j},w_{r_j}}$ be the trip time of rider r_j. The time related trip fare is thus computed as $c^t t_{o_{r_j},w_{r_j}}$, where c^t is the fare per minute determined by the platform. If a driver d_i accepts the matched rider r_j, the trip fare of rider r_j, denoted as z_{r_j}, is computed as:

$$z_{r_j} = \max\{\bar{c} + c^d d_{o_{r_j},w_{r_j}} + c^t t_{o_{r_j},w_{r_j}}, fare^{mini}\} \tag{48}$$



(a) Driver 3 (b) Driver 5 (c) Driver 8

Figure 6.2: Worked example

where $fare^{mini}$ is minimum fare regulated in the ride-hailing system (Garg & Nazerzadeh, 2021).Thus, the expected revenue of the platform is computed as:$\sum_{d_i,r_j \in X^f} p^a_{d_i} z_{r_j}$. Let s_{d_i} be the payment offered to driver d_i. The expected payment is computed as: $\sum_{d_i,r_j \in X^f} p^a_{d_i} s_{d_i}$.

With the above definitions, the payment strategy s_{d_i} of each driver d_i can be computed by solving the following non-linear stochastic optimization model.

$$\underset{s_{d_i}}{\text{maximize}} \quad \sum_{d_i,r_j \in X^f} p^a_{d_i}(z_{r_j} - s_{d_i}) \tag{49a}$$

$$\text{subject to} \quad \sum_{d_i \in X^f} s_{d_i} \leq \sum_{r_j \in X^f} z_{r_j}, \tag{49b}$$

$$s_{d_i} \geq fare^{mini}. \quad \forall d_i \in X^f \tag{49c}$$

The objective function (49a) is to maximize the expected profit of the platform. Constraint (49b) ensures that the total payment is less than or equal to the total fare received by the platform. Constraint (49c) ensures that the price payed to driver d_i should be greater than or equal to the minimum fare regulated in the system.

6.4.3 A worked example

In this subsection, we present a worked example of the proposed approach. This example contains three riders and ten drivers. We intentionally keep this problem oversimplified for the purpose of clearly illustrating the matching and pricing process. Riders submit their ride requests to the service operator with location (origin and destination), latest departure time and required vehicle type information. Thereafter, the operator matches these riders with the open drivers through the

Table 6.2: Results of the worked example

	Approach					
Matched	**The pricing mechanism**			**The flat rate payment**		
pair	Rider trip fare	Driver payment	Acceptance probability	Rider trip fare	Driver payment	Acceptance probability
$\langle d_3, r_2 \rangle$	\$27.63	\$14.96	98%	\$27.63	\$20.7	99.9%
$\langle d_5, r_3 \rangle$	\$21.57	\$16.88	87.2%	\$21.57	\$16.18	68.9%
$\langle d_8, r_1 \rangle$	\$5.21	\$5.92	52%	\$5.21	\$3.9	0.7%
Platform's expected profit		\$16.16			\$8.6	

matching algorithm we designed in section III and obtain a matching result: $(\langle d_3, r_2 \rangle, 18 : 00)$, $(\langle d_5, r_3 \rangle, 18 : 00)$ and $(\langle d_8, r_1 \rangle, 18 : 00)$. In this case, driver d_3 matches to rider r_2, driver d_5 matches to rider r_3 and driver d_8 matches to rider r_1. Both of them will depart from their locations to pick up riders at $18 : 00$. The operator then computes the payment through the pricing mechanism (see Section IV). As shown in Fig. 6.2, the payments offered to driver 3, driver 5 and driver 8 are 14.96\$, 16.88\$ and 5.92\$, respectively. The acceptance probabilities of driver d_3, driver d_5 and driver d_8 under the offered payments are 98%, 87.2% and 52%, with an average of 79.1%. The maximum expected profit is 16.16\$. A detailed matching results can be seen in Table 6.2.

6.5 Numerical experiments

In this section, we conduct experiments to assess the performance of the pricing mechanism in terms of three evaluation metrics which are important to the platform, the participating drivers and riders. In the following, we first present the evaluation metrics. Then, we describe the dataset used to conduct the experiments. Finally, we evaluate the performance of the proposed approach by comparing its results with that generated by the flat rate and the surge pricing strategies against the evaluation metrics.

6.5.1 Evaluation metrics

We evaluate the performance of the proposed approach based on the following metrics.

- **Average acceptance rate** of drivers during a time window f is defined as the ratio of the total acceptance probability of drivers to the number of matched pairs within f,

$$AAR = \frac{\sum_{d_i \in X^f} p^a_{d_i}}{\sum_{d_i, r_j \in X^f} x_{i,j}} \times 100\% \tag{50}$$

where $\sum_{d_i, r_j \in X^f} x_{i,j}$ is the number of successfully matched rider-driver pairs in f.

- **Expected profit** of the system within f is defined in (49a).
- **Expected matching rate** of the ride-hailing system within a time window f is defined as the product of the average acceptance rate and the matching rate within f,

  $$EMR = AAP \times \frac{\sum_{d_i, r_j \in X^f} x_{i,j}}{R^f} \tag{51}$$

  where R^f is the number of ride requests in f.
- **Expected social welfare** of the ride-hailing system within a time window f is defined as the acceptance probability of drivers multiply the social welfare value of the system,

  $$ESW = \sum_{d_i \in X^f} p^a_{d_i} \times SW \tag{52}$$

  where SW is the social welfare value defined in equation (3).
- **Average rider fare** within a time window f is defined as the ratio of the total trip fares received from the matched riders to the total number of matched riders within the time window,

  $$ARF = \frac{\sum_{r_j \in X^f} z_{r_j}}{\sum_{d_i, r_j \in X^f} x_{i,j}} \tag{53}$$

  where z_{r_j} is the trip fare of rider r_j which is defined in (48).

6.5.2 Data description

We demonstrate the performance of the proposed pricing mechanism using RideAustin data, spanning from early-June 2016 to mid-April 2017 (Austin, 2016). There are in total 1,494,125 trip records in this dataset. Each trip record in the dataset includes the rider's request time, the

Longitude and Latitude of pick-up and drop-off locations, and the trip start and end timestamps. The rider's rating and the rider's requested vehicle type are also included in the dataset. For each driver, the dataset contains the driver's vehicle type, available time and the distance (in *mile*) and time (in *minute*) traveled by each driver to pick up and drop off the rider. The dataset has four types of vehicles, that are *Standard*, *SUV*, *Premium* and *Luxury*. All the information is needed by the proposed approach.

The left side of Figure 6.3 shows the number of ride requests aggregated in a day from June 6, 2016 to April 13, 2017. Clearly, March 2017 has the largest number of ride requests. We exact one day trip requests in this month, 2017-03-03, from the dataset for the purpose of evaluating the performance of the proposed pricing mechanism. The geographical distributions of ride requests of the day are illustrated on the right side of Figure 6.3. Based on this data, six instances are created with various commuter quantities by randomly sampling different time ranges from this day. The details of the instances are listed in Table 6.3.

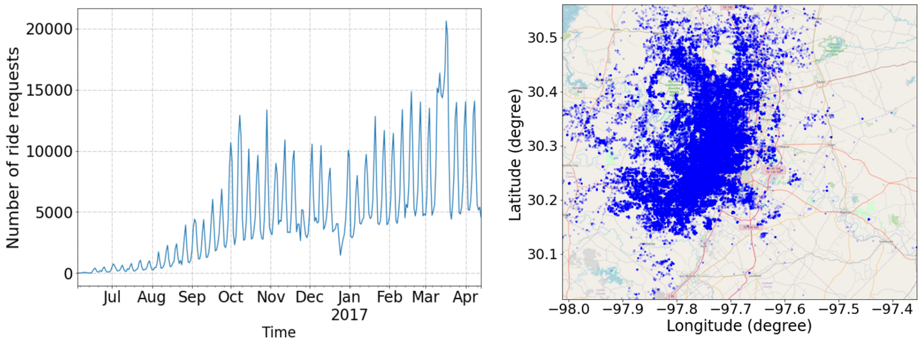


Figure 6.3: Left side, number of ride requests within a day from June 6, 2016 to April 13, 2017. Right side, spatial distributions of rider requests in 2017-03-03.

According to the current rates of RideAustin services (Austin, 2021), we set the base fare $\bar{c} =$ \$2.50, the time rate $c^t =$ \$0.2 per minute, the distance rate $c^d =$ \$0.99 per mile and the minimum fare $fare^{mini} =$ \$4. A report from (Smartrak, 2021) indicates that drivers' idle time costs are closely related to the fuel consumption. Thus, for each driver d_i, the idle time cost coefficient β_{d_i} is randomly drawn from a uniform distribution on $U(0.01, 0.04)$ per minute, where $U(a, b)$ represents a uniform probability distribution between a and b. For each rider r_j, the delay tolerance rate λ_{r_j} is randomly drawn from a uniform distribution in the $U(0.1, 0.8)$ range. This range is chosen based

Table 6.3: Number of drivers and riders in 6 testing instances

Instance ID	Number of drivers	Number of riders	Time range
1	92	50	2:00-2:05
2	90	42	4:00-2:05
3	89	33	9:00-9:05
4	93	29	12:00-12:05
5	186	69	18:00-18:05
6	228	81	19:00-19:05
7	200	78	20:00-20:05
8	300	110	22:00-22:05

on the customer survey results presented in (Buchholz et al., 2020). For the sake of simplicity, we select seven variables to formulate the systematic utility function of a driver. These variables are the travel time and travel distance for a driver to pick up the assigned ride, the assigned rider's trip distance, the assigned rider's location region, the assigned rider's rating, the driver's comfort and the payment offered by the platform. As indicated in (Ashkrof et al., 2020), these are the top seven variables that influence a driver's ride acceptance behavior. The values of the systematic utility coefficients $\varrho_1 - \varrho_7$ are generated based on a ride acceptance behavior analysis paper (Ashkrof, Correia, Cats, & van Arem, 2021). The parameter setting of the proposed matching and pricing is shown in Table 6.4. Note that the values of the above mentioned parameters can be easily adjusted according to the specific operating condition of a ride-hailing system. All numerical experiments are coded in Python and tested on a 2.2 GHz Intel i7 laptop with 16GB RAM. The optimization models (8) and (15) are solved using Gurobi 8.13. For instance 6 with the highest number of drivers and riders, the framework takes less than 20 seconds to compute a matching and individual payment result.

6.5.3 Performance evaluation

In this subsection, we compare the results generated by our approach with those generated by the flat rate payment strategy. Under the flat rate payment strategy, the platform keeps a percentage of the rider's fare (in this experiment, we use 25%) and transferring the rest to the driver. Thus, the platform's profit for a ride request is computed as $25\% \times z_{r_j}$, where z_{r_j} is the trip fare of rider r_i defined in Equation (48). The driver's payment is thus computed as $75\% \times z_{r_j}$. Let $\hat{p}^a_{d_i}$ be the

Table 6.4: Parameter setting of the proposed framework.

Parameters	Description	Values
β_{d_i}	driver idle time cost coefficient	$U(0.01, 0.04)$
γ_{r_i}	rider delay tolerance rate	$U(0.1, 0.8)$
$fare^{mini}$	minimum fare	\$4
$\bar{c}$	base fare	\$2.5
c^t	time rate	\$0.2 per minute
c^d	distance rate	\$0.99 per mile
ϱ_1	travel time coefficient	$U(-1.5, -2)$
ϱ_2	travel distance coefficient	$U(-0.5, -1)$
ϱ_3	pick up location region coefficient	$U(-1.5, -2.5)$
ϱ_4	trip distance coefficient	$U(0.5, 3)$
ϱ_5	rider rating coefficient	$U(-0.5, -0.6)$
ϱ_6	comfort coefficient	$U(16, 18)$
ϱ_7	payment coefficient	$U(1, 3)$

acceptance probability of driver d_i under the flat rate payment strategy. Note that the variables of computing $\hat{p}^a_{d_i}$ are the same as those described in subsection IV-B, except that the driver's payment variable is now computed as $75\% \times z_{r_j}$. Given the above defined, the platform's expected profit under the flat rate payment strategy is computed as:

$$\sum_{d_i, r_j \in X^f} \hat{p}^a_{d_i} \left(25\% \times z_{r_j}\right) \tag{54}$$

In what follows, we compare the results computed by the proposed approach with the above defined flat rate payment strategy in terms of the drivers' average acceptance rate, the platform's expected profit and the system's expected matching rate and social welfare. As a single experimental execution involves a certain degree of randomness, we reduce the variance in the results by averaging them over 20 runs.

Figure 6.4 shows the comparison results in terms of the drivers' average acceptance rate over the pricing mechanism and the flat rate payment strategy. Clearly, by applying the proposed approach, significant performance improvements are achieved in all problem instances with gains in average acceptance rate ranging from 50% (instance 3) to 107% (instance 8) compared with the flat rate payment. Taking instance 5 as an example, the proposed approach can achieve on average an 80% acceptance rate. However, flat rate payment only achieves 41%. It is clear from the results that by

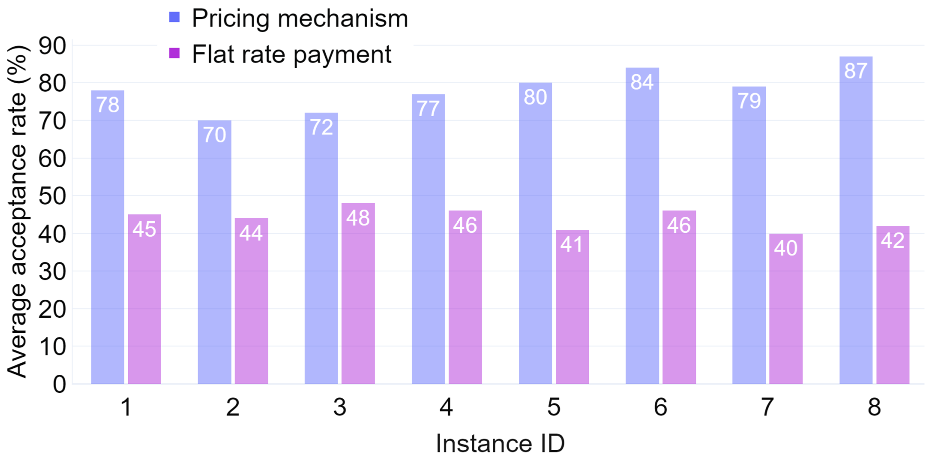


Figure 6.4: Performance comparison in terms of drivers' average acceptance probability between the proposed individual price approach and the flat rate payment strategy.

designing an appropriate payment strategy for each driver, the proposed pricing mechanism boosts the system's efficiency in terms of acceptance rate. This is because the flat rate payment strategy computes the payment for each driver by only considering the length and time cost of the assigned rider's trip. While our proposed approach explicitly models each driver's acceptance behavior so that it can compute a menu of payments tailored to each of the drivers, which directly improves their acceptance rates. The results indicate that while it is simple for the platform to share a fixed base rate with its independent drivers, the platform should adopt a personal payment strategy across different drivers since the underlying operating characteristics of drivers in terms of accepting or rejecting orders can change significantly.

It is important to note that, even the average acceptance rate generated by the proposed approach is much higher than the flat rate payment strategy, the total price paid to the drivers is lower than that of the flat rate payment over each testing instance, as shown in Table 6.5. For example, the drivers' total payment for testing instance 4 is \$375 with the proposed approach, which is more than 18% lower than that of the flat rate payment strategy (in total \$443). This promising result indicates that given the same amount of riders' total trip fares for each testing instance (see column 4 of Table 6.5), the proposed approach has the ability to smartly distribute them among drivers based on drivers' estimated acceptance probabilities so that the platform can spend less money to generate a high average acceptance rate. The improved average acceptance rate and the reduced total payment

Table 6.5: Driver's total payments

	Drivers' total payments ($)		
Instance ID	Pricing mechanism strategy	Flat rate payment strategy	Riders' trip fares ($)
1	465.5	581.2	774.7
2	426	510	630
3	377.5	440.1	587
4	375	443	591
5	720	763	1018
6	833	1063	1417
7	736	980	1205
8	989	1140	1521

directly contribute to the improvement of the platform's expected profit, as shown in Figure 6.5.

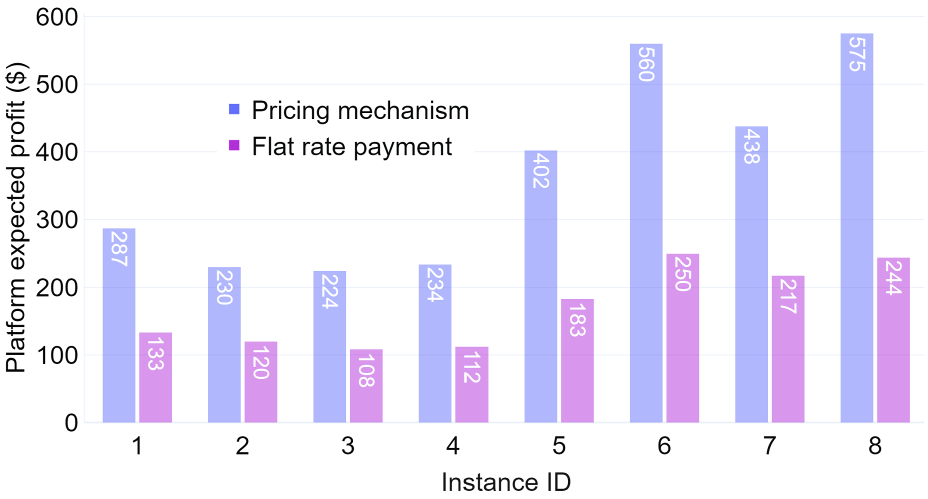


Figure 6.5: Performance comparison in terms of platform's expected profit between the proposed pricing mechanism and the flat rate payment strategy.

The computation results in terms of the platform's expected profit are shown in Figure. 6.5. As expected, with the design of personalized payment, the platform's expected profits are improved significantly in all problem instances compared with the flat rate payment strategy. Specifically, for instance 5, the platform's expected profit under the proposed approach is $402, which is 119% higher than that of the flat rate payment ($183). The considerable gap between the platform's expected profits achieved with the flat rate payment and the pricing mechanism uncovers the significant profit losses caused by the currently applied average pricing rule. Designing individual payments for drivers can benefit the platform substantially since drivers are heterogeneous in terms of the assigned rides. In addition to drivers' average acceptance probability and the platform's profit, by

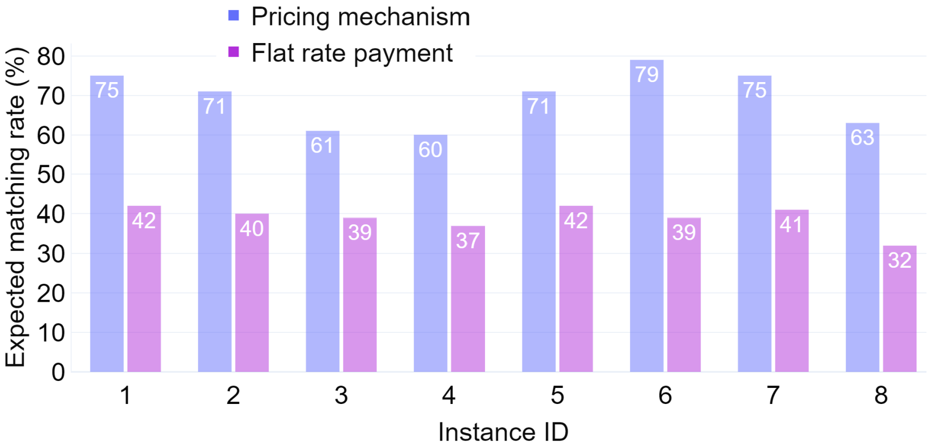


Figure 6.6: Expected matching rate comparison between the proposed approach and the flat rate payment approach.

making the allocation and payment decisions more consistent with drivers preferences, the proposed approach also performs better than the flat rate payment strategy in terms of expected matching rate as shown in Figure 6.6. This makes sense as the proposed approach improves the average acceptance probability which directly improves the expected matching rate. A high matching rate could not only improve riders' satisfaction level, but also ensure the reliability of the ride-hailing system.

In addition to the flat rate payment strategy, surge pricing strategy is also adopted in ride-hailing systems. Under surge pricing, there is a regular price for the entire market region. However, depending on each region's supply and demand condition, the platform can increase the current price at a given region to be higher than the regular price. This higher price is a *surge multiplier*. Let sp be the surge multiplier. The trip fare of a rider r_j under surge pricing strategy is computed as $sp \times z_{r_j}$. To further assess the performance of the proposed pricing mechanism, we compare the results computed by our approach with those computed by the surge pricing strategy under different values of surge multipliers. The results are averaged over the eight testing instances, and the values of surge multipliers are adopted from the RideAustin dataset.

Table 6.6 shows the comparison results in terms of the platform's expected profit, drivers' average acceptance rate, and riders' average trip fare over the proposed approach and the surge pricing strategy across different surge multipliers. From Table 6.6 we have the following observations. The

Table 6.6: Comparison of platform profit and average driver acceptance probability of the proposed approach: *personalized payment strategy* and uber currently applied approach: *surge pricing strategy* under different values of surge multiplier.

Surge Multiplier	Performance					
	Platform expected profit		**Avg. driver acceptance probability**		**Avg.rider trip fare**	
	personalized payment strategy	surge pricing strategy	personalized payment strategy	surge pricing strategy	personalized payment strategy	surge pricing strategy
1.5	$198	$87	60%	38%	$17.6	$17.6
1.75	$282	$116	80%	46%	$21.1	$21.1
2	$363	$148	86%	56%	$24.6	$24.6
2.25	$458	$180	**90%**	65%	**$28**	$28
2.5	$555	$208	91%	70%	$31.6	$31.6
2.75	$651	$238	91.5%	75.4%	$35	$35
3	$749	$266	85%	80%	$38.7	$38.7
3.5	$845	$295	85.2%	83.6%	$42.2	$42.2
4	$1047	$351	93.3%	**90%**	$49.2	**$49.2**
4.5	$1242	$405	93%	94%	$56	$56
5	$1457	$458	97.6%	96%	$63.3	$63.3
5.25	$1654	$511	97.5%	97.4%	$70.3	$70.3
5.75	$1952	$589	96.2	98.8%	$81	$81
6	$2061	$615	98%	99%	$84.4	$84.4

platform's expected profit computed by the proposed approach significantly outperforms the surge pricing strategy across all surge multipliers. Also, the platform's expected profit increases with the increase of the surge multiplier. This makes sense as the platform's profit is deducted from riders' trip fares. When the surge multiplier increases, the rider's average trip fare increases accordingly (as shown in the last two columns of Table 6.6), which directly improves the platform's profit.

On the other hand, the apply of surge multiplier indeed increase the drivers' average acceptance rate. For example, with a surge multiplier of 1.5, the average acceptance probability is 60% under the proposed approach and less than 40% under the surge pricing strategy. When we increase the surge multiplier to 4, the average acceptance probability increases to 93.3% and 90% with the proposed approach and the surge pricing strategy, respectively. However, the cost that comes with the high average acceptance probability is the high-priced riders' average trip fare. As shown in the last column of TABLE 6.6, with a surge multiplier of 4, even the average acceptance probability is 90%, the average trip fare for riders is nearly 50$. This high average trip fare increases the risk of losing riders, which significantly jeopardizes its long-term growth.

However, when incorporating with the proposed approach, as indicated in Table1, with a 2.25 surge multiplier, our approach could achieve on average a 90% acceptance rate. At the same time, the rider's average trip fare is only \$28, which is 40% lower than that of surge pricing strategy (\$49.2). The result indicates that designing an individual payment strategy leads to gains for both the platform, the drivers, and the riders. It is worth noting that, for some cases, surge pricing strategy outperforms the proposed approach in terms of average driver acceptance probability. This is expected as the objective of the payment strategy is to maximize the platform's profit.

6.6 Summary

In this chapter, we design a pricing mechanism that can be used by ride-hailing platforms to compute personalized payments for individual drivers for the purpose of improving system-wide driver average acceptance rate. The proposed mechanism utilizes a behavioral economics-based binary choice model to estimate a driver's acceptance probability given a specific ride request and the associated payment. Taking the estimated acceptance probability as input, a stochastic optimization model is formulated to compute personalized payments for drivers, such that the platform's expected profit, matching rate and the drivers' average ride acceptance rate can be maximized.

Chapter 7

Conclusion and Future Research Plan

This dissertation investigates the key challenges of designing matching mechanisms in two-sided shared mobility systems. Our objective is to design matching mechanisms that are capable of addressing these challenges. In this chapter, we summarize the main contributions of this work and present some future research directions.

7.1 Summary of contributions

The work described in this dissertation makes several important contributions in the area of two-sided shared mobility systems by extending existing matching approaches to more realistic settings involving uncertainties from supply and demand, strategic behaviors of participants, and asymmetric market information. The specific contributions of this dissertation are the following:

- **The design of a price-based iterative double auction** (Chapter 3)

  A price-based iterative double auction is designed to match buyers and sellers in a charger sharing market. The bidding and pricing rules are designed such that when the market reaches an equilibrium meaning that the supply meets the demand given a price vector on services, the resulting matching solution maximizes the social welfare. The proposed auction is budget balanced, individually rational and suitable for a competitive market environment in which myopic best responses from buyers and sellers are expected. It advances the existing literature by extending one-sided auction-based matching to two-sided markets. From a practical application perspective, it achieves much better allocative efficiency than the first-come-first-served charger scheduling scheme which is commonly used by charger sharing platforms. It can also address concerns about privacy because bidders only need to reveal partial and indirect information about their valuations. In addition, it scales well to larger problem instances, which validates its potential for large scale markets.

- **The design of a voting-based matching mechanism** (Chapter 4)

  A non-price voting based matching mechanism is designed to match drivers and riders in a community ride-sharing market. This mechanism is inspired by the simulated annealing meta-heuristic, allowing drivers and riders to negotiate matching solutions iteratively in accordance with their individual preferences. This distributed negotiation process finds high-quality matching solutions and, at the same time, maintains the commuters' privacy. Utility functions for drivers and riders are designed to support the negotiation process and to express commuters' personal preferences over the matching solutions concisely. This mechanism advances the literature by extending the one-sided negotiation mechanism to two-sided markets.

- **The design of a dispatching framework which integrates batched matching with data-driven proactive guidance** (Chapter 5)

  An integrated dispatching framework is designed to match drivers and riders in a ride-hailing market. This framework integrates data-driven proactive guidance strategies with batched matching optimization to increase social welfare, improve the matching rate and reduce rider wait time. Proactive guidance strategies are computed by leveraging short- term demand forecasts based on historical data. Taking the resulting guidance strategies as inputs, the batched matching algorithm computes optimal bipartite matching between drivers and riders in a batch. The proposed framework contributes to the literature by integrating batched matching models with data-driven proactive guidance strategies, thus yielding lower wait times for the riders, shorter idle driving distance for the drivers and higher matching rate in a batching window.

- **The design of a pricing mechanism** (Chapter 6)

A pricing mechanism is designed to improve drivers' average ride acceptance rate by offering personalized payments computed based on the characteristics of individual rides and the estimated acceptance probabilities of the drivers. The proposed ride pricing mechanism is designed based on behavioral economics and stochastic optimization models. Specifically, we model the drivers' ride acceptance rates through a binary choice model and incorporate it into the stochastic optimization problem for the ride-hailing system. This provides personalized payment for each driver in connection with the characteristics of the assigned ride. We then evaluate the performance of the proposed pricing mechanism through extensive numerical experiments based on RideAustin trip data from June 2016 to April 2017. The results suggest that our proposed pricing mechanism improves the drivers' average acceptance rate significantly compared to some commonly used pricing schemes. It also significantly increases the platform's expected profit and matching rate with positive social welfare implications.

7.2 Future Research

Some potential research directions for future work are summarized as follows:

The success of two-sided markets requires a sufficient number of participants, which means market "thickness" is an important factor for a market to work. However, since the matching problem is NP-hard in nature, the designed iterative double auction in Chapter 3 still suffers from high computational cost. Thus, we will continue improving the computational efficiency of the iterative double auction, especially the SA-based winner determination algorithm, so that it can take full advantage of the parallel computing capability offered by the clouds to further reduce the computation time required to reach a market equilibrium. We will also explore the possibility of introducing other approximate and heuristic algorithms for the winner determination problem. While these algorithms can come with different flavors, those that preserve incentive compatibility are worth investigating. In addition, the designed iterative double auction is more suitable for a day-ahead market which only runs during a specified time period the day before the operating day. We will extend the auction to more dynamic online two-sided markets. These markets employ batched-auctions which allow buyers and sellers to enter the market at anytime and get near real-time trade confirmations

Another important future direction is to develop machine learning approaches to learn drivers' and riders' matching preferences. In Chapters 4-6, drivers' and riders' preferences are quantified by the design of utility functions. At the current stage, the parameters of the utility functions are estimated based on some behavior survey papers. We can use machine learning approaches such as deep learning to learn these parameters from individual riders' and drivers' historical behavior data. Moreover, in Chapter 6, we have applied the discrete choice theory to compute drivers' ride acceptance probabilities. Given that the drivers' decisions of whether to accept the rider requests or not are changing over time, developing a pricing mechanism that combines reinforcement learning and mechanism design would be another step toward more practical approaches to real-world decentralized and dynamic matching applications. In addition to that, the proposed drivers' discrete choice model is calibrated using the data from a driver behavior survey. Should real-world driver behavior data collected by ride-hailing systems becomes available, estimating model parameters and calibrating them using that data will provide us with more accurate driver acceptance probabilities.

The currently designed mechanisms in Chapters 5-6 optimize the matching in one region and during one batched window. Two possible extensions in this regard can be considered in future work. First, the proactive guidance and batched matching models can be extended to maximize the social welfare of drivers and riders across multiple regions. Second, we can concurrently optimize consecutive batching windows using a rolling horizon modeling approach. This way, we can exploit the interdependence between matching solutions for adjacent batching windows to further improve the matching efficiency. In addition, to predict rider demand, we have implemented a time-series prediction approach, ARIMA, into our framework. To make the approach more sensitive to the hourly changing trends of the demand, deep-learning-based time series forecasting methods can be developed and evaluated.

References

Agatz, N., Erera, A., Savelsbergh, M., & Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, *223*(2), 295–303.

Agatz, N., Erera, A. L., Savelsbergh, M. W., & Wang, X. (2011). Dynamic ride-sharing: A simulation study in metro atlanta. *Procedia-Social and Behavioral Sciences*, *17*, 532–550.

Agency, I. E. (2019). *Global ev outlook 2019, scaling up the transition to electric mobility.* Retrieved 2019-05, from https://www.iea.org/publications/reports/globalevoutlook2019/

Aïvodji, U. M., Huguenin, K., Huguet, M.-J., & Killijian, M.-O. (2018). Sride: A privacy-preserving ridesharing system. In *Proceedings of the 11th acm conference on security & privacy in wireless and mobile networks* (pp. 40–50).

Asghari, M., & Shahabi, C. (2017). An on-line truthful and individually rational pricing mechanism for ride-sharing. In *Proceedings of the 25th acm sigspatial international conference on advances in geographic information systems* (pp. 1–10).

Ashkrof, P., Correia, G. H. d. A., Cats, O., & van Arem, B. (2021). Ride acceptance behaviour of ride-sourcing drivers. *arXiv preprint arXiv:2107.07864*.

Ashkrof, P., de Almeida Correia, G. H., Cats, O., & van Arem, B. (2020). Understanding ride-sourcing drivers' behaviour and preferences: Insights from focus groups analysis. *Research in Transportation Business & Management*, *37*, 100516.

Association, A. A. (2019). *Your driving costs: How much are you really paying to drive?* Retrieved 2020-07-30, from https://exchange.aaa.com/wp-content/uploads/2019/09/AAA-Your-Driving-Costs-2019.pdf

Association, A. A., et al. (2011). Your driving costs: How much are you really paying to drive? 2011 edition. *Heathrow, FL: American Automobile Association*.

Austin, R. (2016). *Data file and code book. data.world. available.* Retrieved 2010-09-30, from https://data.world/ride-austin/

Austin, R. (2021). *Driver rates.* Retrieved 2021-08-03, from http://www.rideaustin.com/drivers/rates

Ausubel, L. M., Milgrom, P., et al. (2006). The lovely but lonely vickrey auction. *Combinatorial auctions*, *17*, 22–26.

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

Bei, X., & Zhang, S. (2018). Algorithms for trip-vehicle assignment in ride-sharing. In *Thirty-second aaai conference on artificial intelligence.*

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

Bian, Z., Liu, X., & Bai, Y. (2020). Mechanism design for on-demand first-mile ridesharing. *Transportation research part B: methodological*, *138*, 77–117.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, *35*(3), 268–308.

Braverman, A., Dai, J. G., Liu, X., & Ying, L. (2019). Empty-car routing in ridesharing systems. *Operations Research*, *67*(5), 1437–1452.

Buchholz, N., Doval, L., Kastl, J., Matějka, F., & Salz, T. (2020). *The value of time: Evidence from auctioned cab rides* (Tech. Rep.). National Bureau of Economic Research.

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

Castillo, J. C., Knoepfle, D., & Weyl, G. (2017). Surge pricing solves the wild goose chase. In *Proceedings of the 2017 acm conference on economics and computation* (pp. 241–242).

Chau, S. C.-K., Shen, S., & Zhou, Y. (2020). Decentralized ride-sharing and vehicle-pooling based on fair cost-sharing mechanisms. *IEEE Transactions on Intelligent Transportation Systems*.

Chen, M. K. (2016). Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform. In *Proceedings of the 2016 acm conference on economics and computation* (pp. 455–455).

Chen, W., Mes, M., Schutten, M., & Quint, J. (2019). A ride-sharing problem with meeting points and return restrictions. *Transportation science*, *53*(2), 401–426.

Chen, Y., Zhang, J., Wu, K., & Zhang, Q. (2013). Tames: A truthful double auction for multi-demand heterogeneous spectrums. *IEEE Transactions on Parallel and Distributed Systems*, *25*(11), 3012–3024.

Chichin, S., Vo, Q. B., & Kowalczyk, R. (2016). Towards efficient and truthful market mechanisms for double-sided cloud markets. *IEEE Transactions on Services Computing*, *10*(1), 37–51.

Chu, L. Y., & Shen, Z.-J. M. (2008). Truthful double auction mechanisms. *Operations research*, *56*(1), 102–120.

Cradeur, J. (2019). *Big changes could be coming to uber's destination filter.* Retrieved 2020-08-03, from https://therideshareguy.com/changes-to-ubers-destination-filter/

Cradeur, J. (2021). *Uber finally lets drivers see where passengers are going!* Retrieved 2021-08-03, from https://therideshareguy.com/uber-rolling-out-new-driver-features/

Cui, Y., Makhija, R. S. M. S., Chen, R. B., He, Q., & Khani, A. (2020). Understanding and modeling the social preferences for riders in rideshare matching. *Transportation*, 1–27.

Cui, Y., Makhija, R. S. M. S., Chen, R. B., He, Q., & Khani, A. (2021). Understanding and modeling the social preferences for riders in rideshare matching. *Transportation*, *48*(4), 1809–1835.

Darabi, Z., Fajri, P., & Ferdowsi, M. (2016). Intelligent charge rate optimization of phevs incorporating driver satisfaction and grid constraints. *IEEE Transactions on Intelligent Transportation Systems*, *18*(5), 1325–1332.

de Hoog, J., Alpcan, T., Brazil, M., Thomas, D. A., & Mareels, I. (2014). Optimal charging of electric vehicles taking distribution network constraints into account. *IEEE Transactions on Power Systems*, *30*(1), 365–375.

de Hoog, J., Alpcan, T., Brazil, M., Thomas, D. A., & Mareels, I. (2015). A market mechanism for electric vehicle charging under network constraints. *IEEE Transactions on Smart Grid*, *7*(2), 827–836.

Deng, R., Yang, Z., Hou, F., Chow, M.-Y., & Chen, J. (2014). Distributed real-time demand response in multiseller–multibuyer smart distribution grid. *IEEE Transactions on Power Systems*, *30*(5), 2364–2374.

Eva. (n.d.). *Think locally and move easily, year = 2021, url = https://eva.coop//rider, urldate = 2021-07-05.*

Evans, D. S. (2003). The antitrust economics of multi-sided platform markets. *Yale J. on Reg.*, *20*, 325.

Faqiry, M. N., & Das, S. (2018). Double auction with hidden user information: Application to energy transaction in microgrid. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*(99), 1–14.

Feng, G., Kong, G., & Wang, Z. (2020). We are on the way: Analysis of on-demand ride-hailing systems. *Manufacturing & Service Operations Management*.

Feng, Y., Niazadeh, R., & Saberi, A. (2021). Two-stage stochastic matching with application to ride hailing. In *Proceedings of the 2021 acm-siam symposium on discrete algorithms (soda)* (pp. 2862–2877).

Fink, A. (2004). Supply chain coordination by means of automated negotiations. In *37th annual hawaii international conference on system sciences, 2004. proceedings of the* (pp. 10–pp).

Fishburn, P. C. (1970). *Utility theory for decision making* (Tech. Rep.). Research analysis corp McLean VA.

Foundation, C. W. (2019). *The sharing economy takeover.* Retrieved 2019-08, from https://cwf.ca/research/publications/blog-the-sharing-economy-takeover/

Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., & Koenig, S. (2013). Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, *57*, 28–46.

Gao, G., Xiao, M., & Zhao, Z. (2016). Optimal multi-taxi dispatch for mobile taxi-hailing systems.

In *2016 45th international conference on parallel processing (icpp)* (pp. 294–303).

Gao, J., Li, X., Wang, C., & Huang, X. (n.d.). Learning-based open driver guidance and rebalancing for reducing riders' wait time in ride-hailing platforms. In *2020 ieee international smart cities conference (isc2)* (pp. 1–7).

Gao, J., Li, X., Wang, C., & Huang, X. (2021). Bm-ddpg: An integrated dispatching framework for ride-hailing systems. *IEEE Transactions on Intelligent Transportation Systems (Early Access). DOI: 10.1109/TITS.2021.3106243*.

Gao, J., & Wang, C. (2018). Automated negotiation protocol for collaborative diagnostic services scheduling. In *2018 ieee 22nd international conference on computer supported cooperative work in design ((cscwd))* (pp. 140–145).

Gao, J., Wong, T., & Wang, C. (2019). Coordinating patient preferences through automated negotiation: a multiagent systems model for diagnostic services scheduling. *Advanced Engineering Informatics*, *42*, 100934.

Gao, J., Wong, T., & Wang, C. (2021). Social welfare maximizing fleet charging scheduling through voting-based negotiation. *Transportation Research Part C: Emerging Technologies*, *130*, 103304.

Gao, J., Wong, T., Wang, C., & Yu, J. Y. (2021). A price-based iterative double auction for charger sharing markets. *IEEE Transactions on Intelligent Transportation Systems (Early Access) DOI: 10.1109/TITS.2020.3047984*.

Garg, N., & Nazerzadeh, H. (2021). Driver surge pricing. *Management Science*.

Gerding, E. H., Stein, S., Robu, V., Zhao, D., & Jennings, N. R. (2013). Two-sided online markets for electric vehicle charging. In *Proceedings of the 2013 international conference on autonomous agents and multi-agent systems* (pp. 989–996).

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

Guo, X., Caros, N. S., & Zhao, J. (2021). Robust matching-integrated vehicle rebalancing in ride-hailing system with uncertain demand. *Transportation Research Part B: Methodological*, *150*, 161–189.

Haeringer, G. (2018). *Market design: auctions and matching*. MIT Press.

Hall, D., & Lutsey, N. (2017). Emerging best practices for electric vehicle charging infrastructure. *The International Council on Clean Transportation (ICCT): Washington, DC, USA*.

Hawksworth, J., & Vaughan, R. (2014). The sharing economy–sizing the revenue opportunity. *PricewaterhouseCoopers.(available at http://www. pwc. co. uk/issues/megatrends/collisions/sharingeconomy/the-sharing-economysizing-therevenue-opportunity. html)*.

He, S., & Shin, K. G. (2019). Spatio-temporal adaptive pricing for balancing mobility-on-demand networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(4), 1–28.

Holland-Letz, D. (2021). *Mobility's future: An investment reality check.* Retrieved 2021-04, from https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/mobilitys-future-an-investment-reality-check

Howell, D., Boyd, S., Cunningham, B., Gillard, S., Slezak, L., Ahmed, S., ... others (2017). *Enabling fast charging: A technology gap assessment* (Tech. Rep.).

Iglesias, R., Rossi, F., Wang, K., Hallac, D., Leskovec, J., & Pavone, M. (2018). Data-driven model predictive control of autonomous mobility-on-demand systems. In *2018 ieee international conference on robotics and automation (icra)* (pp. 1–7).

Iosifidis, G., Gao, L., Huang, J., & Tassiulas, L. (2015). A double-auction mechanism for mobile data-offloading markets. *IEEE/ACM Transactions on Networking (TON)*, *23*(5), 1634–1647.

Jia, Y., Xu, W., & Liu, X. (2017). An optimization framework for online ride-sharing markets. In *2017 ieee 37th international conference on distributed computing systems (icdcs)* (pp. 826–835).

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

Kang, J., Yu, R., Huang, X., Maharjan, S., Zhang, Y., & Hossain, E. (2017). Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Transactions on Industrial Informatics*, *13*(6), 3154–3164.

Kang, Q., Feng, S., Zhou, M., Ammari, A. C., & Sedraoui, K. (2017). Optimal load scheduling of plug-in hybrid electric vehicles via weight-aggregation multi-objective evolutionary algorithms. *IEEE Transactions on Intelligent Transportation Systems*, *18*(9), 2557–2568.

Kang, Q., Wang, J., Zhou, M., & Ammari, A. C. (2015). Centralized charging strategy and scheduling algorithm for electric vehicles under a battery swapping scenario. *IEEE Transactions on Intelligent Transportation Systems*, *17*(3), 659–669.

Karamanis, R., Anastasiadis, E., Angeloudis, P., & Stettler, M. (2020). Assignment and pricing of shared rides in ride-sourcing using combinatorial double auctions. *IEEE Transactions on Intelligent Transportation Systems*.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671–680.

Kleiner, A., Nebel, B., & Ziparo, V. A. (2011). A mechanism for dynamic ride sharing based on parallel auctions. In *Twenty-second international joint conference on artificial intelligence.*

Kolisch, R. (1996). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, *90*(2), 320–333.

Korkas, C. D., Baldi, S., Yuan, S., & Kosmatopoulos, E. B. (2017). An adaptive learning-based approach for nearly optimal dynamic charging of electric vehicle fleets. *IEEE Transactions on Intelligent Transportation Systems*, *19*(7), 2066–2075.

Korolko, N., Woodard, D., Yan, C., & Zhu, H. (2018). Dynamic pricing and matching in ride-hailing platforms. *Available at SSRN*.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, *2*(1-2), 83–97.

Kutanoglu, E., & David Wu, S. (1999). On combinatorial auction and lagrangean relaxation for distributed resource scheduling. *IIE transactions*, *31*(9), 813–826.

Lang, F., Fink, A., & Brandt, T. (2016). Design of automated negotiation mechanisms for decentralized heterogeneous machine scheduling. *European Journal of Operational Research*, *248*(1), 192–203.

Lee, D.-H., Wang, H., Cheu, R. L., & Teo, S. H. (2004). Taxi dispatch system based on current demands and real-time traffic conditions. *Transportation Research Record*, *1882*(1), 193–200.

Louviere, J. J., Hensher, D. A., & Swait, J. D. (2000). *Stated choice methods: analysis and* *applications*. Cambridge university press.

Lowalekar, M., Varakantham, P., & Jaillet, P. (2018). Online spatio-temporal matching in stochastic and dynamic domains. *Artificial Intelligence*, *261*, 71–112.

Lyft. (2020). *Preferred rides: Experience the next level of ride comfort.* Retrieved 2021-01-28, from https://www.lyft.com/blog/posts/preferred-rides/

Lyft. (2021). *The driver's guide to pay.* Retrieved 2021-11-07, from https://www.lyft.com/driver/pay

Machado, C. A. S., de Salles Hue, N. P. M., Berssaneti, F. T., & Quintanilha, J. A. (2018). An overview of shared mobility. *Sustainability*, *10*(12), 4342.

Majumder, B. P., Faqiry, M. N., Das, S., & Pahwa, A. (2014). An efficient iterative double auction for energy trading in microgrids. In *2014 ieee symposium on computational intelligence applications in smart grid (ciasg)* (pp. 1–7).

Marshall, A. (2020). *Uber changes its rules, and drivers adjust their strategies.* Retrieved 2020-08-03, from https://www.wired.com/story/uber-changes-rules-drivers-adjust-strategies/

Mas-Colell, A., Whinston, M. D., Green, J. R., et al. (1995). *Microeconomic theory* (Vol. 1). Oxford university press New York.

Masoud, N., & Jayakrishnan, R. (2017). A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system. *Transportation Research Part B: Methodological*, *106*, 218–236.

McAfee, R. P. (1992). A dominant strategy double auction. *Journal of economic Theory*, *56*(2), 434–450.

McFadden, D. (1981). Econometric models of probabilistic choice. *Structural analysis of discrete data with econometric applications*, *198272*.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., & Damas, L. (2013). Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, *14*(3), 1393–1402.

Mourad, A., Puchinger, J., & Chu, C. (2019). A survey of models and algorithms for optimizing shared mobility. *Transportation Research Part B: Methodological*, *123*, 323–346.

Nicoll, E., & Armstrong, S. (2016). Ride-sharing: The rise of innovative transportation services. *Retrieved from MaRs: https://www. marsdd. com/news-and-insights/ride-sharing-the-rise-of-innovative-transportation-services*.

Nisan, N., Roughgarden, T., Tardos, E., & Vazirani, V. V. (2007). *Algorithmic game theory*. Cambridge university press.

Niu, K., Wang, C., Zhou, X., & Zhou, T. (2019). Predicting ride-hailing service demand via rpa-lstm. *IEEE Transactions on Vehicular Technology*, *68*(5), 4213–4222.

Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013). Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists* (Vol. 1, pp. 380–384).

OpenData, N. (2016). *2016 green taxi trip data.* Retrieved 2020-02-07, from `https://data.cityofnewyork.us/Transportation/2016-Green-Taxi-Trip-Data/hvrh-b6nb/data`

Parkes, D. C. (2006). *Iterative combinatorial auctions*. MIT press.

Parkes, D. C., & Kalagnanam, J. (2005). Models for iterative multiattribute procurement auctions. *Management Science*, *51*(3), 435–451.

Parkes, D. C., & Ungar, L. H. (2000). Iterative combinatorial auctions: Theory and practice. *AAAI/IAAI*, *7481*.

Parkes, D. C., & Ungar, L. H. (2001a). An auction-based method for decentralized train scheduling. In *Proceedings of the fifth international conference on autonomous agents* (pp. 43–50).

Parkes, D. C., & Ungar, L. H. (2001b). *Iterative combinatorial auctions: Achieving economic and computational efficiency*. University of Pennsylvania.

Payyanadan, R. P., & Lee, J. D. (2018). Understanding the ridesharing needs of older adults. *Travel Behaviour and Society*, *13*, 155–164.

Pinedo, M. (2012). *Scheduling* (Vol. 29). Springer.

Plenter, F., Chasin, F., von Hoffen, M., Betzing, J. H., Matzner, M., & Becker, J. (2018). Assessment of peer-provider potentials to share private electric vehicle charging stations. *Transportation Research Part D: Transport and Environment*, *64*, 178–191.

Rahman, I., Vasant, P. M., Singh, B. S. M., Abdullah-Al-Wadud, M., & Adnan, N. (2016). Review of recent trends in optimization techniques for plug-in hybrid, and electric vehicle charging infrastructures. *Renewable and Sustainable Energy Reviews*, *58*, 1039–1047.

Rardin, R. L., & Rardin, R. L. (1998). *Optimization in operations research* (Vol. 166). Prentice Hall Upper Saddle River, NJ.

Rochet, J.-C., & Tirole, J. (2004). Two-sided markets: an overview. *Institut d'Economie Industrielle working paper*.

Rochet, J.-C., & Tirole, J. (2006). Two-sided markets: a progress report. *The RAND journal of economics*, *37*(3), 645–667.

Rosenblat, A. (2018). *Uberland: How algorithms are rewriting the rules of work*. Univ of California Press.

Rosenblat, A., & Stark, L. (2016). Algorithmic labor and information asymmetries: A case study of uber's drivers. *International Journal of Communication*, *10*, 27.

Rysman, M. (2009). The economics of two-sided markets. *Journal of economic perspectives*, *23*(3), 125–43.

Samadi, P., Schober, R., & Wong, V. W. (2011). Optimal energy consumption scheduling using mechanism design for the future smart grid. In *2011 ieee international conference on smart grid communications (smartgridcomm)* (pp. 369–374).

Santos, A., McGuckin, N., Nakamoto, H. Y., Gray, D., & Liss, S. (2011). *Summary of travel trends: 2009 national household travel survey* (Tech. Rep.).

Sarriera, J. M., Álvarez, G. E., Blynn, K., Alesbury, A., Scully, T., & Zhao, J. (2017). To share or not to share: Investigating the social aspects of dynamic ridesharing. *Transportation Research Record*, *2605*(1), 109–117.

Schlagwein, D., Schoder, D., & Spindeldreher, K. (2020). Consolidated, systemic conceptualization, and definition of the "sharing economy". *Journal of the Association for Information Science and Technology*, *71*(7), 817–838.

Schroeder, A., & Traber, T. (2012). The economics of fast charging infrastructure for electric vehicles. *Energy Policy*, *43*, 136–144.

Seow, K. T., Dang, N. H., & Lee, D.-H. (2009). A collaborative multiagent taxi-dispatch system. *IEEE Transactions on Automation science and engineering*, *7*(3), 607–616.

Shaheen, S., Cohen, A., Randolph, M., Farrar, E., Davis, R., & Nichols, A. (2019). Shared mobility policy playbook.

Shaheen, S., Cohen, A., Zohdy, I., et al. (2016). *Shared mobility: current practices and guiding principles* (Tech. Rep.). United States. Federal Highway Administration.

Shaheen, S. A. (2016). Mobility and the sharing economy. *Transport Policy*, *51*(Supplement C), 141–142.

Shoham, Y., & Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Shokoohyar, S., Sobhani, A., & Sobhani, A. (2020). Impacts of trip characteristics and weather condition on ride-sourcing network: Evidence from uber and lyft. *Research in transportation economics*, *80*, 100820.

Smartrak. (2021). *The costs of fleet idle time.* Retrieved 2021-08-03, from https://smartrak.com/the-costs-of-fleet-idle-time/

Stiglic, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2015). The benefits of meeting points in ride-sharing systems. *Transportation Research Part B: Methodological*, *82*, 36–53.

Stiglicc, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2016). Making dynamic ride-sharing work: The impact of driver and rider flexibility. *Transportation Research Part E: Logistics and Transportation Review*, *91*, 190–207.

Tafreshian, A., & Masoud, N. (2020). Trip-based graph partitioning in dynamic ridesharing. *Transportation Research Part C: Emerging Technologies*, *114*, 532–553.

Tafreshian, A., Masoud, N., & Yin, Y. (2020). Frontiers in service science: Ride matching for peer-to-peer ride sharing: A review and future directions. *Service Science*, *12*(2-3), 44–60.

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.

Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge university press.

Tsao, M., Iglesias, R., & Pavone, M. (2018). Stochastic model predictive control for autonomous mobility on demand. In *2018 21st international conference on intelligent transportation systems (itsc)* (pp. 3941–3948).

Uber. (2020a). *How much does a ride with the uber app cost?* Retrieved 2020-02-07, from https://www.uber.com/us/en/price-estimate/

Uber. (2020b). *Ratings, rider feedback, and what it means.* Retrieved 2021-01-28, from https://www.uber.com/blog/rider-feedback/

Uber. (2021). *Innovative ways to reach more people.* Retrieved 2021-08-03, from https://www.uber.com/ca/en/transit/products/

Urata, J., Xu, Z., Ke, J., Yin, Y., Wu, G., Yang, H., & Ye, J. (2021). Learning ride-sourcing drivers' customer-searching behavior: A dynamic discrete choice approach. *Transportation Research Part C: Emerging Technologies*, *130*, 103293.

Vanrykel, F., Ernst, D., & Bourgeois, M. (2018). Fostering share&charge through proper regulation. *Competition and Regulation in Network Industries*, *19*(1-2), 25–52.

Vazifeh, M. M., Santi, P., Resta, G., Strogatz, S. H., & Ratti, C. (2018). Addressing the minimum fleet problem in on-demand urban mobility. *Nature*, *557*(7706), 534–538.

Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, *16*(1), 8–37.

Volkov, M., Aslam, J., & Rus, D. (2012). Markov-based redistribution policy model for future urban mobility networks. In *2012 15th international ieee conference on intelligent transportation systems* (pp. 1906–1911).

Von Neumann, J., & Morgenstern, O. (2007). *Theory of games and economic behavior (commemorative edition).* Princeton university press.

Wang, C. (2007). *Economic models for decentralized scheduling* (Unpublished doctoral dissertation). Faculty of Graduate Studies, University of Western Ontario.

Wang, C. (2012). Requirement-based bidding language for agent-based scheduling. *IEEE Transactions on Automation Science and Engineering*, *10*(1), 219–223.

Wang, C., & Dargahi, F. (2013). Service customization under capacity constraints: an auction-based model. *Journal of Intelligent Manufacturing*, *24*(5), 1033–1045.

Wang, C., Ghenniwa, H. H., & Shen, W. (2009). Constraint-based winner determination for auction-based scheduling. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *39*(3), 609–618.

Wang, C., Wang, Z., Ghenniwa, H. H., & Shen, W. (2011). Due-date management through iterative bidding. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *41*(6), 1182–1198.

Wang, H., & Yang, H. (2019). Ridesourcing systems: A framework and review. *Transportation Research Part B: Methodological*, *129*, 122–155.

Wang, Q., Liu, X., Du, J., & Kong, F. (2016). Smart charging for electric vehicles: A survey from the algorithmic perspective. *IEEE Communications Surveys & Tutorials*, *18*(2), 1500–1517.

Wang, Y., Lin, X., Wei, H., Wo, T., Huang, Z., Zhang, Y., & Xu, J. (2019). A unified framework with multi-source data for predicting passenger demands of ride services. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *13*(6), 1–24.

Wellman, M. P., Walsh, W. E., Wurman, P. R., & MacKie-Mason, J. K. (2001). Auction protocols for decentralized scheduling. *Games and economic behavior*, *35*(1-2), 271–303.

Xia, M., Stallaert, J., & Whinston, A. B. (2005). Solving the combinatorial double auction problem. *European Journal of Operational Research*, *164*(1), 239–251.

Xing, X., Warden, T., Nicolai, T., & Herzog, O. (2009). Smize: a spontaneous ride-sharing system for individual urban transit. In *German conference on multiagent system technologies* (pp. 165–176).

Xu, K., Sun, L., Liu, J., & Wang, H. (2018). An empirical investigation of taxi driver response behavior to ride-hailing requests: A spatio-temporal perspective. *PloS one*, *13*(6), e0198605.

Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., … Ye, J. (2018). Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining* (pp. 905–913).

Yan, C., Zhu, H., Korolko, N., & Woodard, D. (2020). Dynamic pricing and matching in ride-hailing platforms. *Naval Research Logistics (NRL)*, *67*(8), 705–724.

Yang, H., Shao, C., Wang, H., & Ye, J. (2020). Integrated reward scheme and surge pricing in a ridesourcing market. *Transportation Research Part B: Methodological*, *134*, 126–142.

Yang, M., & Xia, E. (2021). A systematic literature review on pricing strategies in the sharing economy. *Sustainability*, *13*(17), 9762.

Yassine, A., Hossain, M. S., Muhammad, G., & Guizani, M. (2019). Double auction mechanisms for dynamic autonomous electric vehicles energy trading. *IEEE Transactions on Vehicular*

Technology, *68*(8), 7466–7476.

Yousaf, J., Li, J., Chen, L., Tang, J., & Dai, X. (2014). Generalized multipath planning model for ride-sharing systems. *Frontiers of computer science*, *8*(1), 100–118.

Yu, J., Mo, D., Xie, N., Hu, S., & Chen, X. M. (2021). Exploring multi-homing behavior of ride-sourcing drivers via real-world multiple platforms data. *Transportation Research Part F: Traffic Psychology and Behaviour*, *80*, 61–78.

Yuan, N. J., Zheng, Y., Zhang, L., & Xie, X. (2012). T-finder: A recommender system for finding passengers and vacant taxis. *IEEE Transactions on knowledge and data engineering*, *25*(10), 2390–2403.

Zha, L., Yin, Y., & Du, Y. (2018). Surge pricing and labor supply in the ride-sourcing market. *Transportation Research Part B: Methodological*, *117*, 708–722.

Zha, L., Yin, Y., & Xu, Z. (2018). Geometric matching and spatial pricing in ride-sourcing markets. *Transportation Research Part C: Emerging Technologies*, *92*, 58–75.

Zhan, X., Qian, X., & Ukkusuri, S. V. (2016). A graph-based approach to measuring the efficiency of an urban taxi service system. *IEEE Transactions on Intelligent Transportation Systems*, *17*(9), 2479–2489.

Zhang, H., & Zhao, J. (2018). Mobility sharing as a preference matching problem. *IEEE Transactions on Intelligent Transportation Systems*, *20*(7), 2584–2592.

Zhang, L., Hu, T., Min, Y., Wu, G., Zhang, J., Feng, P., … Ye, J. (2017). A taxi order dispatch model based on combinatorial optimization. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining* (pp. 2151–2159).

Zhang, R., & Pavone, M. (2016). Control of robotic mobility-on-demand systems: a queueing-theoretical perspective. *The International Journal of Robotics Research*, *35*(1-3), 186–203.

Zhao, K., Tarkoma, S., Liu, S., & Vo, H. (2016). Urban human mobility data mining: An overview. In *2016 ieee international conference on big data (big data)* (pp. 1911–1920).

Zuniga-Garcia, N., Tec, M., Scott, J. G., Ruiz-Juri, N., & Machemehl, R. B. (2020). Evaluation of ride-sourcing search frictions and driver productivity: A spatial denoising approach. *Transportation Research Part C: Emerging Technologies*, *110*, 346–367.