Reinforced stable matching for Crowd-Sourced Delivery Systems under stochastic driver acceptance behavior

Hou, Shixuan; Wang, Chun; Gao, Jie

DOI
10.1016/j.trc.2024.104916

Publication date
2024

Document Version
Final published version

Published in
Transportation Research Part C: Emerging Technologies

Citation (APA)
Hou, S., Wang, C., & Gao, J. (2024). Reinforced stable matching for Crowd-Sourced Delivery Systems under stochastic driver acceptance behavior. *Transportation Research Part C: Emerging Technologies*, *170*, Article 104916. https://doi.org/10.1016/j.trc.2024.104916

Transportation Research Part C 170 (2025) 104916

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Reinforced stable matching for Crowd-Sourced Delivery Systems under stochastic driver acceptance behavior

Shixuan Hou [a], Chun Wang [a], Jie Gao [b,*]

[a] Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montréal, Canada
[b] Delft University of Technology (TU Delft), Delft, Netherlands

ARTICLE INFO

Keywords:
Crowd-sourced delivery
Reinforced stable matching
Gale–Shapley algorithm
Compensation
Behavior uncertainty

ABSTRACT

Crowd-Sourced Delivery Systems (CDS) depend on occasional drivers to deliver parcels directly to online customers. These freelance drivers have the flexibility to accept or reject orders from the platform, leading to a stochastic and often unstable matching process for delivery assignments. This instability results in frequent rematching, delayed deliveries, decreased customer satisfaction, and increased operational costs, all highlighting the critical need for improved matching stability within CDS. While traditional stable matching theory provides a foundation, it primarily addresses static and deterministic scenarios, making it less effective in the dynamic and unpredictable environments typical of CDS. Addressing this gap, this study extends the classic Gale–Shapley (GS) stable matching algorithm by incorporating tailored compensations for drivers, incentivizing them to accept assigned orders and thus improving the stability of matchings, even with the inherent uncertainties of driver acceptance. We prove that the proposed mechanism can generate reinforced stable matching results based on tailored compensation values. Also, our numerical study shows that this reinforced stable matching approach significantly outperforms traditional methods in terms of both matching stability and cost-effectiveness. It reduces the order rejection rate to as low as 1% and cuts operational costs by up to 18%.

1. Introduction

As the digital marketplace expands, the global e-commerce market is projected to grow at an annual rate of 11.17% between 2023 and 2027, reaching a market size of $5.56 trillion.[1] Within such a sizable market, there is an urgent demand for innovative urban logistics solutions capable of facilitating rapid, cost-effective, and reliable delivery options. Inspired by the sharing economy, a novel business model known as CDS has emerged. The CDS employs ordinary individuals, such as commuters, travelers, and shoppers, as a supplement to the professional fleet for delivering online orders (Archetti et al., 2016). By effectively utilizing existing transportation resources, this model has yielded notable economic, social, and environmental benefits (Le et al., 2019).

The advantages of the CDS have made it popular among major e-commerce sites, retail chains, and logistics companies, such as Amazon Flex,[2] Walmart Spark,[3] and FedEx.[4] Despite its benefits, operating such a system poses challenges. CDS operates as a

* Corresponding author.
E-mail addresses: shou@ivey.ca (S. Hou), chun.wang@concordia.ca (C. Wang), j.gao-1@tudelft.nl (J. Gao).

[1] https://www.statista.com/outlook/dmo/ecommerce/worldwide.
[2] https://flex.amazon.com/.
[3] https://drive4spark.walmart.com/.
[4] https://www.fedex.com/.

https://doi.org/10.1016/j.trc.2024.104916
Received 5 March 2024; Received in revised form 8 September 2024; Accepted 31 October 2024
Available online 15 November 2024

two-sided market where drivers and customers each have their own preferences. Drivers may prioritize orders based on factors such as delivery distance, payment, or convenience, while customers may prefer certain drivers based on criteria such as driver ratings, delivery speed, or reliability (Punel et al., 2018; Le and Ukkusuri, 2019; Al-Saudi and Himpel, 2020). When the matching process does not adequately consider these preferences, it can result in rejected order assignments and low rating scores for the delivery service (Said, 2021; Xu et al., 2018). This highlights the need for a stable matching process to ensure optimal outcomes for both parties. Stable matching algorithms have been successfully applied in various fields to achieve matching stability. In their seminal paper, Gale and Shapley (1962) propose a matching algorithm for school admissions and marriage markets in 1962, also known as the Gale–Shapley (GS) algorithm. Over the years, this theory has significantly evolved, finding substantial applications in various real-world markets—from labor markets (Roth and Peranson, 1999), school choice programs (Balinski and Sönmez, 1999), to kidney exchange (Roth et al., 2005a). In the transportation sector, particularly within shared mobility systems, stable matching algorithms have proven to have good performance in enhancing both efficiency and effectiveness. For instance, they have been applied to ride-sharing systems (Wang et al., 2018), ride-hailing systems (Sühr et al., 2019) and CDS (Zhang et al., 2023). However, the existing papers have a set of matching-related assumptions. First, the assumption of *deterministic preferences*, where participants are expected to have clear and complete preferences. Second, the assumption of *rationality*, where participants will always accept pairings that optimize their utilities. Third, the assumption of *internal characteristic* suggests that the preferences of individuals within a pairing are solely determined by the inherent attributes of the opposing side in a two-sided market, unaffected by external environmental factors.

In practise, these established assumptions are not applicable to CDS due to their highly decentralized, stochastic, and dynamic environment. In a more realistic CDS setting, occasional drivers (are abbreviated as: drivers), also known as crowd-shippers, may not be able to accurately and completely state their preferences, and their order acceptance decisions remain uncertain (Hou et al., 2022). Moreover, external factors such as the compensation offered can influence drivers' preferences. The main contributions of this paper can be summarized as follows: (1). We propose the concept of reinforced stable matching, which ensures the stability of the matching results through externality compensation, thereby reducing the probability of matching failures; (2). We address the uncertainty inherent in the CDS matching process, particularly the uncertainty in driver acceptance behavior. This is quantified through an analysis of survey data; (3). We validate the feasibility and effectiveness of the reinforced stable matching mechanism through a discrete event simulation.

The remainder of this paper proceeds as follows. In Section 2, we review the literature relevant to the matching problem in CDS, followed by a problem description in Section 3. Section 4 presents the proposed reinforced stable matching mechanism. While in Section 5, we introduce the questionnaire design and analyze the collected data. In Section 6, we conduct computational experiments and Section 7 concludes this paper.

2. Literature review

In this section, we first review literature related to the design of matching approaches in CDS system, then we review literature on the theory of stable matching in two-sided markets, followed by the applications of stable matching theory in different transportation fields.

2.1. Matching in CDS

Effective matching, corresponding to the process of finding a proper driver to deliver orders, is the key to the success of CDS. Nowadays, a considerable amount of research has been devoted to the design of matching approaches for CDS. For recent reviews, we refer to the readers to Alnaggar et al. (2021) and Pourrahmani and Jaller (2021), where matching approaches are generally classified as deterministic and stochastic.

To the best of our knowledge, Archetti et al. (2016) was the first to propose the concept of crowd-sourced delivery services. This paper investigates a last-mile/same-day delivery company, employing in-store occasional drivers to deliver parcels. The matching problem is formulated as an integer programming model that minimizes the total delivery costs. Building on this foundation, some researchers have addressed the problem of order-driver matching in more complex delivery systems. Sampaio et al. (2020) introduce benefits of transfer locations to the crowd-sourced delivery system. To solve the matching problem between orders and drivers, a flow-based mixed integer programming model is proposed to minimize the total delivery costs. A similar setting of introducing intermediate depots is investigated by Macrina et al. (2020), who formulate a matching and routing problem of a crowd-sourced delivery system as an integer linear programming model. The objective is to minimize the total delivery costs and carbon emissions. Similarly, with the goal of minimizing the total costs of delivering all the parcels on time, a mixed integer programming is proposed for multi-driver-multi-order matching and scheduling (Chen et al., 2018). Behrend and Meisel (2018) consider a joint decision-making problem that matches supply requests and trip deliveries in an integrated system of crowd-shipping and item sharing. This problem is formulated as an integer programming model that maximizes the total profits of the system. In addition to classic optimization approaches, some articles have considered matching methods within a game-theoretic framework. Le et al. (2021) consider a bidding system for CDS system that order requesters declare willingness to pay (WTP) and crowd-shippers state their expected to-be-paid (ETP). This paper proposes a mixed-integer non-linear model to minimize the total system benefits. Based on a similar bidding system, Boysen et al. (2022) propose a mixed integer programming model to compute the optimal matching between drivers and a bundle of customers, aiming to minimize the total system costs.

Considering the high variability and dynamic of the market, Arslan et al. (2019) formulate a matching problem between ad hoc drivers and parcel delivery tasks to an integer programming model that minimizes the costs of ad hoc driver matches and dedicated vehicle matches. Tu et al. (2019) consider an online crowd-sourced food delivery system. A mixed-integer programming model is proposed to minimize the total delivery costs and delivery delays. This paper uses a short-sight strategy to achieve dynamically order-task allocations. Behrend et al. (2019) extend their previous work (Behrend and Meisel, 2018) by allowing drivers to deliver multiple items at once, for which they develop an exact solution approach. To minimize the total carrier travel time and the deviation from the original travel route, Allahviranloo and Baghestani (2019) formulate a mixed-integer programming model and propose a bidding system to handle the pricing and compensation scheme decisions.

The adoption of crowd-shippers in CDS introduces uncertainties not present in traditional logistics. These uncertainties include driver availability, spatiotemporal characteristics, and behavior. Specifically, Mousavi et al. (2022) investigate the uncertainty surrounding the availability of crowd-shippers and propose a two-stage stochastic integer program to minimize the sum of mobile depot operational costs, penalty costs for postponing deliveries, and crowd-shippers' compensation. Dayarian and Savelsbergh (2020) propose two rolling horizon dispatching mechanisms to dynamically match orders and crowd-shippers, taking into consideration probabilistic information about future online orders and crowd-shippers arrivals. To manage the uncertainty of crowd-shippers' availability, a matching and routing problem is formulated as a two-stage stochastic programming model to minimize total expected delivery costs (Torres et al., 2022). The occasional availability of crowd-shippers is estimated based on historical data. However, crowd-shippers in CDS may not always accept the orders assigned by the system. Acknowledging this acceptance uncertainty, Gdowska et al. (2018) formulate the matching problem between crowd-shippers and packages as a stochastic integer programming model that minimizes the total delivery costs. Recently, Hou et al. (2022) model the uncertainty of crowd-shippers' order acceptance as a discrete choice model, incorporated into a two-stage optimization framework, aiming to compute a cost-effective optimal matching and compensation scheme. This work focuses on designing pricing strategies to encourage drivers to accept orders, employing a two-stage stochastic programming approach. While this method effectively optimized acceptance rates through financial incentives, it did not fully address the stability of the matching process.

The aforementioned approaches compute efficient system-wide matching results in CDS, such as cost minimization, profit maximization, and carbon emission reduction. However, they do not consider individual preferences, which may lead to a scenario where system participants, drivers in our context, strategically reject matches in the hope of obtaining more favorable matching outcomes. This scenario is referred to as *matching instability*, which affect the reliability and stability of the system (Zhang et al., 2023).

2.2. Stable matching

Since Gale and Shapley (1962) first introduced the concept of stable matching in 1962, its theory has matured over time through accumulation and evolution. And it is successful applied in numerous fields, including labor markets (Roth and Peranson, 1999; Hou, 2019), school admissions programs (Balinski and Sönmez, 1999; Biró et al., 2010), and kidney exchanges (Roth et al., 2005a,b).

In recent years, considering the variability and uncertainty of the modern market, the academic community has gradually enriched research on promoting stable matching by adopting incentive mechanisms, for example: Aziz et al. (2020) explore the stable matching problem under three models of uncertainty: the lottery model, the compact indifference model, and the joint probability model. It examines the computational complexity of determining the stability probability of a given matching and finding a matching with the highest probability of being stable. Liu et al. (2014) consider the stable matching problems with one-sided asymmetric information and address the conceptual challenge of defining a blocking pair when considering the inferences an uninformed agent might make. The study proves that the set of stable outcomes in an incomplete information environment is a superset of the set of complete-information stable outcomes. Additionally, it explores the concept of price-sustainable allocations, showing that incomplete-information stable matchings are a subset of such allocations. Rasulkhani and Chow (2019) construct a generalized market equilibrium model based on the assignment game criteria. The model aims to find stable pricing and matching, allowing operators and users to match user groups to route collections in the network based on route capacity. The objective is to maximize the total utility of all matched users and routes, subtracting the related costs. Zhou et al. (2019) consider the optimization of computation resource allocation and task assignment in Vehicular Fog Computing (VFC). A pricing-based stable matching algorithm is proposed for task assignment, which significantly impacts problem-solving by iteratively updating preference lists to achieve a stable matching.

While the previously mentioned studies have acknowledged how external factors, especially pricing, can shape agent preferences, they have primarily focused on devising pricing and incentive strategies to ensure stable matchings. These studies rest on the assumption that agents act rationally, meaning they will choose matches or assignments that optimize their payoff. In this work, we relax the assumption of rationality and study how to design a stable matching solution through employing advanced machine learning methods to quantify irrational behavior.

In recent years, the use of stable matching theory in the transportation sector has gained considerable attention Recently, Zhang et al. (2023) attempt to apply stable matching theory to the matching mechanism design for the CDS system. They formulate the matching problem between pre-planned trips and delivery requests as a non-cooperative game and propose two exact algorithms to find stable matchings for CDS matching problems. Additionally, we have observed some efforts to apply stable matching to the fields like ride-sharing and other urban mobility systems. Wang et al. (2018) investigate a stable matching problem for ride-sharing systems with incomplete information. They formulate mathematical models that generate stable and nearly stable matching solutions. Peng et al. (2020) develop a stable matching model for ride-sharing systems and propose a stable matching mechanism

Table 1
The differences between this paper and the most relevant literature.

Studies	Service platform	Stable matching	Incentive	Behavior uncertainty
Peng et al. (2020), Yan et al. (2021), Wang et al. (2018) and Zhang and Zhao (2018)	Ride-sharing	✓	✗	✗
Shurrab et al. (2021)	EV charging-sharing	✓	✗	✗
Zhao et al. (2019)	On-demand taxi	✓	✗	✗
Rasulkhani and Chow (2019)	Generalized transportation system	✓	✓	✗
Zhou et al. (2019)	Vehicular Fog Computing	✓	✓	✗
Aziz et al. (2020)	Generalized two-sided matching market	✓	✗	✗
Liu et al. (2014)	Labor market	✓	✗	✗
Roth and Peranson (1999) and Hou (2019)	Labor market	✓	✗	✗
Balinski and Sönmez (1999) and Biró et al. (2010)	School admissions	✓	✗	✗
Zhang et al. (2023)	Crowd-sourced delivery	✓	✗	✗
Roth et al. (2005a) and Roth et al. (2005b)	Kidney exchanges	✓	✗	✗
Arslan et al. (2019), Tu et al. (2019), Behrend et al. (2019), Dayarian and Savelsbergh (2020), Macrina et al. (2020), Torres et al. (2022), Behrend and Meisel (2018), Sampaio et al. (2020), Chen et al. (2018), Mousavi et al. (2022), Le et al. (2021), Allahviranloo and Baghestani (2019), Boysen et al. (2022) and Archetti et al. (2016)	Crowd-sourced delivery	✗	✗	✗
Hou et al. (2022) and Gdowska et al. (2018)	Crowd-sourced delivery	✗	✓	✓
This study	Crowd-sourced delivery	✓	✓	✓

that incorporates both a payment incentive mechanism and a deferred acceptance algorithm. Yan et al. (2021) design a matching and pricing mechanism to balance system-wide optimality with matching stability. Shurrab et al. (2021) address an EV-to-EV (V2V) charging energy sharing problem and propose a two-layer matching mechanism. This approach leverages the Gale–Shapley algorithm and a user-satisfaction model to efficiently match EVs. Zhao et al. (2019) propose an online stable matching model to address the preference-aware task matching problem in an on-demand taxi dispatching environment with the goal of maximizing total profit and minimizing the number of blocking pairs. Zhang and Zhao (2018) highlight that while most ride-sharing algorithms focus solely on system optimization objectives, very few explicitly consider passengers' preferences for their peers as a matching criterion. Their proposed preference-based model outperforms efficiency-based models in terms of passenger preference satisfaction, with only a moderate loss at the aggregate level.

The aforementioned literature operates under the assumption that all participants are rational without considering external characteristics and have deterministic preferences, such that they will always accept the matching results that maximize their utility. This utility maximizing behavior is assumed to be consistent across different situations. However, in a more realistic CDS setting, the acceptance decisions of drivers are not so predictable, they may choose not to accept matched orders even if such orders could maximize their utilities (Hou et al., 2022). This could be due to personal preferences, external circumstances like traffic or weather conditions, or other unpredictable human factors (Hou et al., 2022; Ashkrof et al., 2020). The difference between theoretical assumption and practical behavior in CDS highlight the need to consider the behavior uncertainties when designing the stable matching mechanism in CDS.

Our work focus on generating stable matching results with the consideration of drivers' behavior uncertainty. We propose a matching mechanism that includes a tailored compensation scheme aimed at increasing the probability of drivers accepting matched orders, thereby reinforcing the stability of the matching results. Table 1 summarizes the differences between this paper and the related works.

3. Problem description

In this section, we introduce a CDS system that leverages occasional drivers for order deliveries. We begin with an overview of the system's architecture, highlighting the interrelationships among its components. Following this, we employ a Discrete Event Simulation (DES) model to describe the dynamics of the CDS system. By simulating the behavior of each occasional driver, particularly focusing on order acceptance choice behavior, the DES model assists in evaluating the impact of these behaviors on overall system performance. With the insight gained from the DES, we formulate the assignment of orders to drivers as a classic bilateral matching problem. For detailed notations and descriptions, please refer to Table 3.

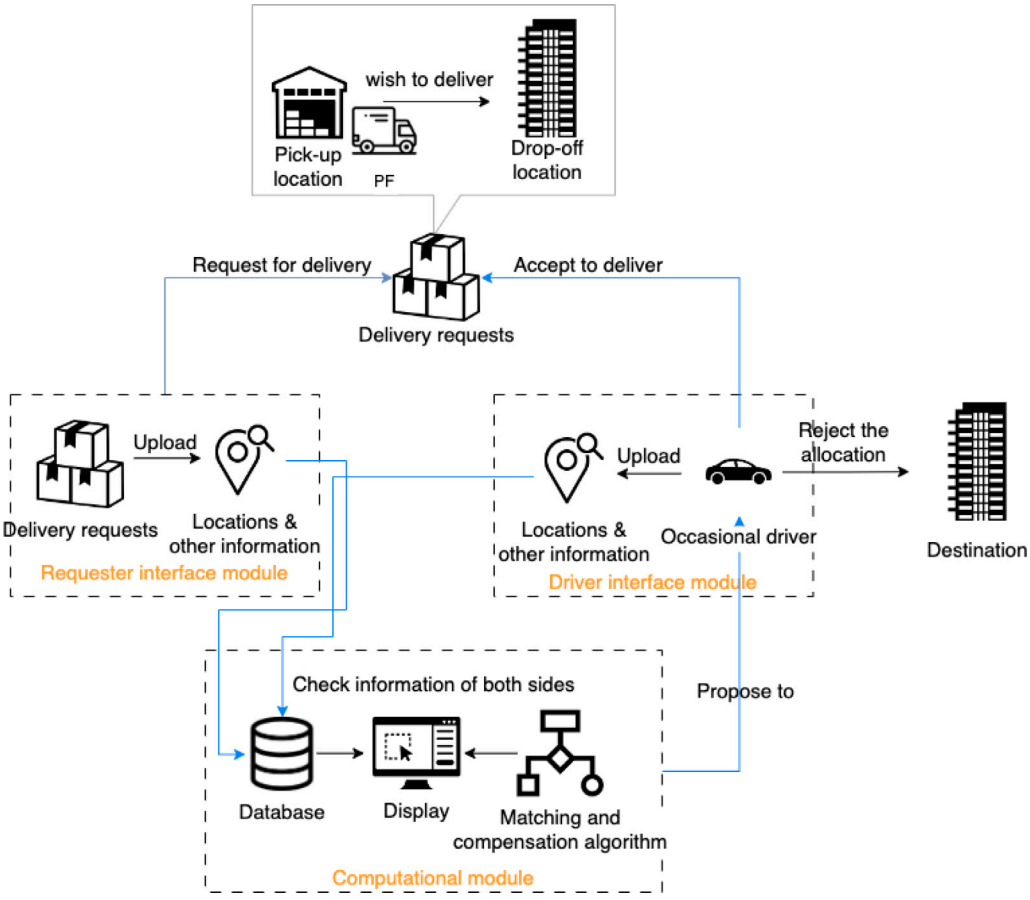


Fig. 1. Components of the CDS system and their relationships.

3.1. System overview

As illustrated in Fig. 1, the CDS system consists of three modules, *computational module*, *requester interface module*, and *driver interface module*. The requester, which can be either an individual or a corporate entity, seeks a proper driver for the delivery of parcels or documents (hereinafter referred to as "orders"). Each requester uploads the order information to the system database via the requester interaction module, with the pick-up location, drop-off location, and the latest delivery time. Drivers, serving as one-time crowd-shippers, deliver the orders in exchange for compensations. Each driver declares her availability to deliver orders and uploads her origin, destination, latest departure time and transportation mode through the driver interaction module to the system database. The computational module, leveraging the aggregated data, calculates a matching solution that aligns with both the order's and driver's preferences. This solution is proposed to drivers through the driver interface module, asking them to decide whether to accept or reject the matched orders. The accepted orders are delivered by occasional drivers, while the rejected orders, as well as the unmatched orders are assigned to professional fleets (PFs).

3.2. Discrete event simulation model

Let $\mathcal{I}$ be the set of all occasional drivers. For a driver $i \in \mathcal{I}$, the main *events* that affect her states are e_i^l, e_i^r, e_i^n, e_i^p, e_i^d, and e_i^s, where e_i^l represents the event that occasional driver i *logs in* the system. e_i^r denotes that an occasional driver i *receives* an order delivery request. e_i^n and e_i^p denote the *order rejection* and *order acceptance* events, respectively. Finally, let e_i^d be the event that occasional driver i *drops off* the accepted order, while e_i^s represents that the driver i *logs out* and leave the system. Please refer to Table 2 for event descriptions.

Let Y be the state space which consists of the set of possible values of the vector $\boldsymbol{y}(k) = [y_1(k) \ldots y_i(k) \ldots y_{|I|}(k)]^T$, where $y_i(k)$ represents the state of the occasional driver i after the kth event happened. The value of the state variable $y_i(k)$ indicates if the driver i:

Table 2
Descriptions of events in the DES model.

Events	Descriptions
e_i^l	Log-in event of occasional driver i, occurs at τ_i^l
e_i^r	Order receive event of occasional driver i, occurs at τ_i^r
e_i^p	Order acceptance event of occasional driver i, occurs at τ_i^p
e_i^n	Order rejection event of occasional driver i, occurs at τ_i^n
e_i^d	Order drop-off event of occasional driver i, occurs at τ_i^d
e_i^s	Log-out event of occasional driver i, occurs at τ_i^s

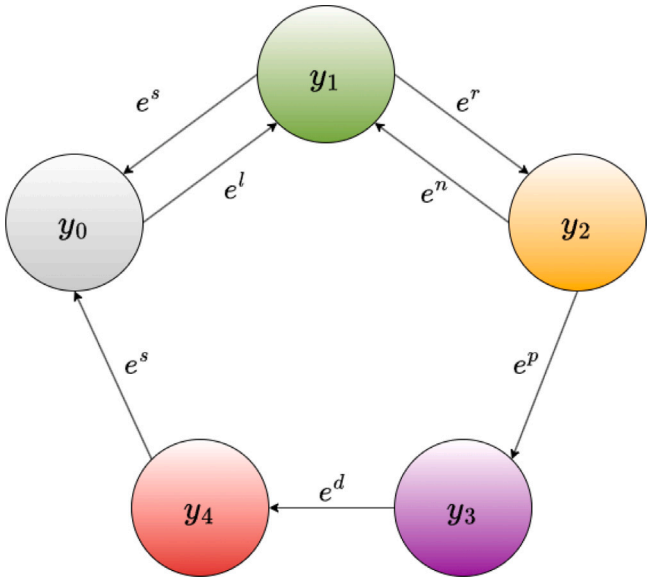


Fig. 2. State transition diagram.

- is unavailable: $y_i(k) = y_0$;
- is available to be matched: $y_i(k) = y_1$;
- is evaluating an order: $y_i(k) = y_2$;
- is in motion: $y_i(k) = y_3$;
- is at order's drop-off location: $y_i(k) = y_4$.

The state transitions diagram of an occasional driver i is shown in Fig. 2. And the details of some important events are illustrated below.

3.2.1. Driver log-in and driver log-out event

For each driver $i \in \mathcal{I}$, the occurrence of her log-in event e_i^l changes her state from y_0 to y_1, meaning that the driver i declares her availability to be matched. As regards to the event timing, the log-in event e_i^l is assume to occur at τ_i^l. Let $I \subset \mathcal{I}$ be the set of available drivers whose states are y_1 at τ_i^l, and let $J \subset \mathcal{J}$ be the set of available orders that release before τ_i^l ($\rho_j \leq \tau_i^l$), where ρ_j denotes the release time of each order $j \in \mathcal{J}$. We assume that the order releases follow the Poisson process, the elapsed time between an order release and the next one results to be an exponential stochastic variable with the rate λ. And let T be a fixed time interval, the due time of each order $j \in \mathcal{J}$ is denoted by $\delta_j = \rho_j + T$. It should be noted that the occurrence of driver log-out event is scheduled in $\tau_i^s = \tau_i^l + \Delta\iota_i$, being $\Delta\iota_i \sim \mathcal{N}(\iota_i, \sigma_i^2)$, a Gaussian stochastic variable with expectation ι_i equal to the average waiting time of driver i in the system and with variance σ_i^2.

3.2.2. Order receive event

Given the set of available drivers I and available orders J, the proposed matching mechanism, described in Section 4.1, generates a set of matched drivers, denoted by $\hat{I}' \subseteq I$, and a set of matched orders, denoted by $\hat{J}' \subseteq J$. Accordingly, an optimal compensation scheme S, indexed by s, for all matched drivers $\hat{I}'$ is determined via an optimization model, described in the following Section 4.2. With reference to the state diagram in Fig. 2, τ_i^r is the instant at which the state of driver i's state changes from y_1 to y_2. Moreover, according to the solution of Algorithm 1, let $\bar{I} \subseteq I$ be the set of unmatched drivers and $\bar{J} \subseteq J$ be the set of unmatched orders.

Table 3
Notation and description.

Notation	Description
$\mathcal{I}$, $\mathcal{J}$	Set of all occasional drivers, indexed by i, and set of all online orders, indexed by j
I, J	Set of available occasional drivers $\subset \mathcal{I}$, and set of available orders $\subset \mathcal{J}$
$\hat{I}'$, $\hat{J}'$	Set of matched drivers $\subset I$, and set of matched orders $\subset J$
$\hat{I}$, $\hat{J}$	Set of drivers accepting to deliver orders $\subset \hat{I}'$, and set of accepted orders $\subset \hat{J}'$
$\bar{I}$, $\bar{J}$	Set of unmatched drivers $\subset I$, and set of unmatched orders $\subset J$
$\dot{I}$, $\dot{J}$	Set of drivers rejecting to deliver orders $\subset \hat{I}'$, and set of rejected orders $\subset \hat{J}'$
$\tilde{J}$	Set of orders assigned to PFs $= \bar{J} \cup \dot{J}$
S	Set of compensation proposed to matched drivers $\hat{I}'$, indexed by s
o_i, d_i	Origin and destination of driver $i \in \mathcal{I}$
o_j, d_j	Pick-up location and drop-off location of order $j \in \mathcal{J}$
ρ_j, δ_j	Release time and due time of order $j \in \mathcal{J}$
T	A fixed time interval indicating the time window of orders
$\Delta\iota$	A fixed interval indicating the patience (waiting time) of drivers in CDS
$L(a)$	Preference list of an agent a
D_i	Distance of original travel route of occasional driver $i \in \mathcal{I}$
D_j	Distance between pick-up location and drop-off location of order $j \in \mathcal{J}$
D_{o_i,o_j}	Distance between origin o_i of driver $i \in \mathcal{I}$ and pick-up location o_j of order $j \in \mathcal{J}$
D_{d_j,d_i}	Distance between drop-off location d_j of order $j \in \mathcal{J}$ and destination d_i of driver $i \in \mathcal{I}$
$D_{i,j}$	Distance of driver $i \in \mathcal{I}$ to deliver order $j \in \mathcal{J}$
$D^d_{i,j}$	Detour distance of driver $i \in \mathcal{I}$ to deliver order $j \in \mathcal{J}$
V_0	Delivery speed of PF
v_i	Delivery speed of driver $i \in \mathcal{I}$
C^l	Penalty for delayed order
C^o_j	Delivery cost of delivering order $j \in \tilde{J}$ by PF
TD^o_j	Delayed order, 1 if PF deliver order $j \in \bar{J}$ late, 0 otherwise
$TD^d_{i,j}$	Delayed order, 1 if driver $i \in \hat{I}$ deliver order $j \in \hat{J}$ late, 0 otherwise
$U_{i,j}$	Utility of driver i deliver order j
$p^{a(n)}_{i,j}(s)$	Probability that driver i accepts (rejects) to deliver order j given compensation s
μ	A mapping indicting the matching result
$x_{i,j}$	Decision variable, 1 if driver i is matched to order j, 0 otherwise
s_i	Decision variable, indicating the compensation paid to driver i
$s'_{i,j}$	Driver i's expected compensation of delivering order j

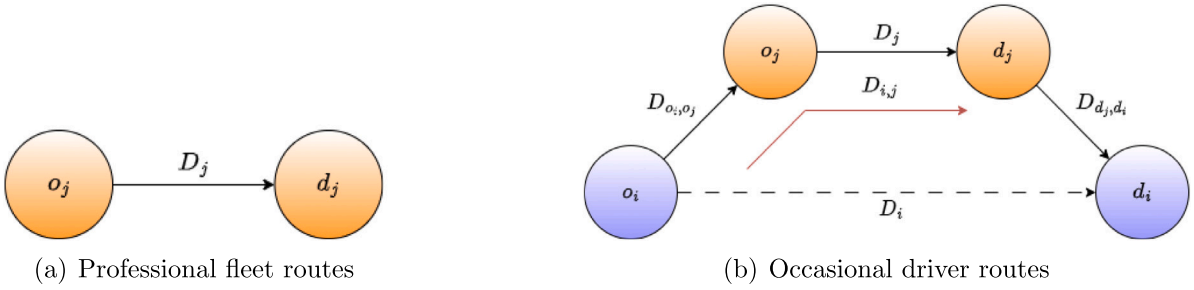


Fig. 3. Delivery routes of occasional drivers and professional fleets.

3.2.3. Order acceptance event

The driver $i \in \hat{I}'$ is free to choose whether to accept the proposal (matched order $j \in \hat{J}'$ with compensation $s \in S$) or not. In the considered model, this freedom is modeled as acceptance probability, which is denoted as $p^a_{i,j}(s)$. If the driver i accepts to deliver order j, meaning the occurrence of event e^p_i, the driver i's state is transferred from y_2 to y_3. The set of drivers who accept to deliver is denoted as $\hat{I} \subseteq \hat{I}'$ and the set of accepted orders is denoted as $\hat{J} \subseteq \hat{J}'$. Note that the drivers who accept to deliver orders coherently update their log-out time corresponding to the order drop-off time.

3.2.4. Order rejection event

For each driver $i \in \hat{I}'$, the probability of rejecting a corresponding matched order $j \in \hat{J}'$ with compensation s is defined as $p^n_{i,j}(s) = 1 - p^a_{i,j}(s)$. If the driver i rejects to deliver order j with compensation s, meaning the occurrence of event e^n_i, the driver i's state is transferred from y_3 back to y_1. Moreover, the set of rejected orders is denoted by $\dot{J}$, where $\dot{J} \cup \hat{J} = \hat{J}'$. Let $\tilde{J} = \dot{J} \cup \bar{J}$ be the orders assigned to PFs.

For simplification, we assume the number of PFs is infinite and they are always on standby at the orders' pick-up locations. Therefore, the delivery time of a PF to deliver a specific unmatched order $j \in \tilde{J}$ is assumed to be $\tau^d_j = \tau^r_i + D_j/V_0$, where D_j denotes the distance between order j's pick-up location o_j and drop-off location d_j, and V_0 denotes the average delivery speed of PFs. Let

D_j/V_0 be the estimated travel time of a PF delivering an order j, the route is illustrated in Fig. 3(a). A late penalty C^l occurs if $\tau_j^d > \delta_j$, regarding a delayed order j by PF, denoted by $TD_j^o = 1$, otherwise $TD_j^o = 0$. The compensation paid to PFs to delivery a rejected order/unmatched order $j \in \tilde{J}$ is assumed to be proportional, via the parameter α_0, to the delivery route distance plus a fix value, and is determined by:

$$C_j^o = C_0 + \alpha_0 D_j \tag{1}$$

3.2.5. Order drop-off event

Let $\tau_{i,j}^d$ be the occurring time that driver $i \in \hat{I}$ drops off the accepted order $j \in \hat{J}$, the occurrence of the order drop-off event e_i^d transfers the driver i's state from y_4 to y_5. In our simulation study, we assume that a specific order drop-off event e_i^d is scheduled at $\tau_{i,j}^d = \tau_i^r + D_{i,j}/v_i$, where $D_{i,j}$ denotes the distance of driver i picks up and drops off an order j, the route refers to Fig. 3(b). Let v_i be the speed of driver i depending on the driver i's transportation mode, introduced in Section 6. A late penalty C^l occurs, if $\tau_{i,j}^d > \delta_j$, regarding a delayed order j, denoted by $TD_{i,j}^d = 1$, otherwise $TD_{i,j}^d = 0$.

The driver log-in event triggers the matching between drivers and orders, which in turn facilitates the order rejection and acceptance events. These events subsequently influence the stability of the matching results. In the following section, we define this two-sided matching problem in more detail and introduce the concept of reinforced matching stability.

3.3. Two-sided matching formulation and reinforced matching stability definition

We formulate the CDS driver-order matching problem as a bilateral matching with two sided preferences (Gale and Shapley, 1962). The two-sided matching, originally designed for a marriage market, aims to match every man with a woman in a stable marriage. A marriage is called stable if and only if there are no agents in any match who would both prefer being matched with each other than their current counterparts (Manlove, 2013). The CDS system consists of two disjoint sets, driver set and order set. Drivers are free to choose appropriate orders for matching, while order requesters have preferences over the driver set. These preferences include factors such as delivery time, delivery cost, and reliability. Based on the similarity drawn between the CDS system and the marriage market, the problem formulation and its associated stability definition are presented as follows.

Assuming a CDS platform, each available order, indexed by $j \in J$, demands a proper available driver $i \in I$ to deliver it, and each driver prefers at most deliver a appropriate order for compensation. The two finite and disjoint sets of agents in the CDS platform are the set $I = \{1, 2, \dots, |I|\}$ of drivers, and $J = \{1, 2, \dots, |J|\}$ of orders. Each driver has preferences over the orders, and each order has preferences over the drivers.

Preferences of both orders and drivers are the basis for the platform to create a stable matching between these two parties. Specifically, the preference of each driver i is represented by an ordered list, $L(i)$, on the set J. That is, a driver i's preferences can be of the form $L(i) = 1 \succ 3 \succ 2, ..,$ indicating that her first choice is to deliver the order 1, her second choice is to deliver the order 3, and her third choice is to deliver order 2.

Given that drivers may not be able to evaluate all available orders to generate a comprehensive and accurate preference list, we propose that each driver's preferences among all orders are shaped by two key factors: the detour distance denoted as $D_{i,j}^d$, and the expected compensation represented as $s'_{i,j}$. The detour distance factor is formulated in Eq. (2).

$$D_{i,j}^d = D_{i,j} + D_{d_j,d_i} - D_i \tag{2}$$

where $D_{i,j}$ denotes the travel distance of driver i delivers order j, defined as $D_{i,j} = D_{o_i,o_j} + D_j$, where D_{o_i,o_j} denotes the distance between driver i's origin and order j's pick-up location, the delivery route can be found in Fig. 3(b). D_{d_j,d_i} is the distance between order j's drop-off location d_j and driver i's destination d_i, and D_i is the distance of driver i's original travel route from origin o_i to destination d_i. The expected compensation is formulated in Eq. (3).

$$s'_{i,j} = C_1 + \alpha_1 D_{i,j}^d \tag{3}$$

where C_1 represents the flat fee rewarded to drivers, and α_1 is the fee rate per kilometer. Furthermore, the predetermined preferences $L(i)$ of driver i over available orders J are given by the descending order of the utility value, as formulated in Eq. (4).

$$U_{i,j} = \beta_0 + \beta_1 D_{i,j}^d + \beta_2 s'_{i,j} = \beta_0 + \beta_1 (D_{i,j} + D_{d_j,d_i} - D_i) + \beta_2 [C_1 + \alpha_1 (D_{i,j} + D_{d_j,d_i} - D_i)] \tag{4}$$

In this equation, β_0, β_1 and β_2 are the estimated parameters, where β_0 is the alternative specific constant (known as ASC or intercept), and β_1 and β_2 are the weights of detour distance $D_{i,j}^d$ and compensation $s'_{i,j}$, respectively.

Similarly, each order j has a preference list $L(j)$ over the driver set I. We assume that each order j's preference is arranged in ascending sort of the estimated travel time $D_{i,j}/v_i$, where $D_{i,j}$ denotes the travel distance of driver i picks up and drops off a given order j, while v_i be the travel speed of driver i, given her transportation mode.

In addition, let $i \succ_j i'$ be the fact that order j prefers i to i', and similarly, we write $j \succ_i j'$ for driver i. And if an individual i is not indifferent between any two alternatives j and j', denoted by $j \sim_i j'$, she has unstrict preferences. For simplification, in this study, we assume that all preferences are strict. Specifically, when utility values are equivalent, we arrange them according to the sequence of agent IDs. Formally, we define a CDS system as a tuple $\langle I, J, \mathcal{L} \rangle$, where $\mathcal{L}$ be the set of all preference lists of both drivers and orders, and the outcome of this CDS system is defined as:

Definition 1. A matching function μ is a one-to-one correspondence from the set $I \cup J$ onto the set $I \cup J$, that is, $\mu^2(i) = i$ such that if $\mu(i) \neq i$, then $\mu(i)$ is in J and if $\mu(j) \neq j$, then $\mu(j)$ is in I. We refer to $\mu(i)$ as the counterpart of i.

Note that $\mu^2(i) = i$ means that if order $\mu(i)$ is matched to driver i, then the driver i is matched to order $\mu(i)$. And $\mu(i) = i$ means the driver i is not matched, while $\mu(j) = j$ denotes the unmatched order j. It is also noteworthy to mention that the GS marriage market (Gale and Shapley, 1962) and related work in the transportation domain, such as ride-sharing market (Wang et al., 2018), taxi sharing market (Peng et al., 2020), EV charging sharing (Shurrab et al., 2021), there is a prevalent assumption of individual rationality. That is, the preference list of an agent is presumed to be truthful, and every element within this list is deemed acceptable to the agent. In this paper, we consider the inherent uncertainty in a driver's order acceptance behavior and potential irrational decision-making. Consequently, building upon the groundwork of previous research, we introduce a concept of reinforced stable matching that incorporates these considerations.

Definition 2. A matching μ between occasional drivers and orders is reinforced stable if and only if there is no blocking compensation s and blocking pair (i, j) given compensation s satisfying the following condition

- If a driver k prefers being unmatched to being matched with $\mu(k)$ with given compensation s, it can be represented as follows:

$$(\mu(k), s') \succ_k (\mu(k), s) \tag{5}$$

- There exists a pair (i, j), where driver $i \in I$ and order $j \in J$. If driver i prefers being matched with j to $\mu(i)$ given compensation s, while order j prefers being matched with i to being matched with $\mu(j)$, the relationships are represented as two in-equations as below, the case is called blocked by pair (i, j) given compensation s

$$(j, s') \succ_i (\mu(i), s) \tag{6}$$

$$(i, s') \succ_j (\mu(j), s) \tag{7}$$

Therefore, Definition 2 essentially suggests that at reinforced stable, no driver can accept an order with an inappropriate compensation, and no driver can further increase her utility by unilaterally switching to another order. Given these notations and definitions, in the following section, we will introduce the designed reinforced stable matching mechanism.

4. Reinforced stable matching mechanism

In this section, we present a two-stage mechanism to achieve reinforced stable matching within the CDS. The first stage aims to generate an initial matching solution ensuring there are no blocking pairs, while the second stage is centered on establishing a compensation scheme for drivers. This scheme considers the acceptance probabilities of drivers to reinforce the matching. Detailed explorations of each stage are provided in Sections 4.1 and 4.2, respectively. Additionally, in Theorem 1, we offer a formal proof confirming the reinforced stability of the matching outcomes produced by this mechanism.

4.1. First stage: Gale–Shapley matching algorithm

Given a set of available drivers I, and a set of available orders J, with their preference lists $\mathcal{L}$, the goal of the first stage is to generate a matching solution that there is no blocking pairs, refer to Algorithm 1. The proof of no blocking pairs is given in Section 4.3.

The Algorithm 1 starts by initializing the sets of matched drivers $\hat{I}' = \emptyset$, matched orders $\hat{J}' = \emptyset$, unmatched drivers $\tilde{I} = I$, and unmatched orders $\tilde{J} = J$ (Line 2 and line 3). And we initialize the matching correspondence by $\mu : \emptyset \longrightarrow \emptyset$ (Line 4). The loops work while there exists an order $j \in \tilde{J}$ that its preference list $L(j)$ is not empty (Line 5). In each loop, each order $j \in \tilde{J}$ propose to its preferred driver i in its preference list $L(j)$ and remove i from $L(j)$ (Line 6). Then if the driver i has not been matched $i \in \tilde{I}$, matching builds $\mu(i) = j$, and add i and j into matched sets I and J, respectively, meanwhile, remove i and j from unmatched sets $\tilde{I}$ and $\tilde{J}$ as well (Line 8). If driver i is matched with an order $j' = \mu(i)$ before, and order j is preferred to j based on driver i's preference list $L(i)$, the driver i will accept the preferred order j, and rejects the order j'. The rejected order j' is put back to unmatched order set $\tilde{J}$, and remove from matched order set $\hat{J}'$. The accepted order j is added into $\hat{J}'$, and remove from $\tilde{J}$ (Line 13). The output of the algorithm is a matching function μ, and the sets of matched drivers $\hat{I}'$, matched orders $\hat{J}'$, unmatched drivers $\tilde{I}$, and unmatched orders $\tilde{J}$.

4.2. Second stage: Compensation model

Given matched drivers, represented as $\hat{I}'$, matched orders, denoted by $\hat{J}'$, and a function μ that maps matched drivers to their respective matched orders, our aim is to determine the best compensation scheme for these matched drivers using a Stochastic Linear Programming model (SLP). For every matched driver $i \in \hat{I}'$, there is a probability that they will either accept or decline their matched order $\mu(i)$. The cost of delivering this order can be visualized as a binary stochastic variable. This variable follows a

Algorithm 1 Gale-Shapley stable matching algorithm (GS)

procedure GS(I, J, $\mathcal{L}$)
  Initialize the matched drivers and orders $\hat{I}' = \hat{J}' = \emptyset$
  Initialize the unmatched drivers and orders $\tilde{I} = I$ and $\tilde{J} = J$
  Initialize the matching $\mu : \emptyset \longrightarrow \emptyset$
  while $\exists j \in \tilde{J}$ & $L(j) \neq \emptyset$ **do**
    $i :=$ highest-ranked driver in $L(j)$, j is assigned to i, remove i from $L(j)$
    if $i \notin \hat{I}$ **then**
      $\mu(i) = j$, add i and j into $\hat{I}'$, and $\hat{J}'$, and remove i and j from $\tilde{I}$ and $\tilde{J}$
    end if
    if $i \in \hat{I}$ **then**
      $j' = \mu(i)$
      if $j \succ_i j'$ **then**
        $\mu(i) = j$, remove j' from $\hat{J}'$, add j' into $\tilde{J}$, remove j from $\tilde{J}$, and add j into $\hat{J}'$
      end if
    end if
  end while
  $\tilde{I}$, and $\tilde{J}$ are assigned to PF
  return μ, $\hat{I}'$, $\hat{J}'$,
end procedure

Bernoulli distribution $\sim \mathcal{B}(p^a_{i,\mu(i)})$. If a driver declines an order, which happens with a probability of $p^n_{i,j}$, the delivery cost becomes $C^o_{\mu(i)}$. However, if a driver accepts the order (which happens with a probability $p^a_{i,\mu(i)}$), the delivery cost is denoted by a decision variable s_i. This s_i is essentially the compensation given to driver i for the delivery. The objective function Eq. (8) aims to minimize the expected delivery cost of orders that have been matched.

$$\min_{s} \sum_{\forall i \in \hat{I}'} (s_i p^a_{i,\mu(i)} + C^o_{\mu(i)} p^n_{i,\mu(i)}) \tag{8}$$

subject to:

$$C^o_{\mu(i)} = C_0 + \alpha_0 D_{\mu(i)}, \quad \forall i \in \hat{I}' \tag{9}$$

$$\sum_{\forall i \in \hat{I}} s_i \leq \sum_{\forall i \in \hat{I}} \omega C^o_{\mu(i)}, \quad \forall i \in \hat{I}' \tag{10}$$

$$s_i \geq 0, \quad \forall i \in \hat{I}' \tag{11}$$

Eq. (9) define the compensation paid to PFs to deliver rejected orders. Eq. (10) ensure the compensation budget, which must lower than a rate w of original delivery costs of PFs. Eq. (11) define the decision variable. Solving the optimization problem determines the optimal compensation scheme for each matched driver, reported by $s_i \in S$, where $i \in \hat{I}'$.

4.3. Reinforced stable matching mechanism: algorithm and proof

Algorithm 2 Reinforced Gale-Shapley stable matching mechanism (R-GS)

procedure R-GS(I, J, $\mathcal{L}$)
  Initialize the set of drivers $\hat{I} = \emptyset$ who accept to deliver, and accepted orders $\hat{J} = \emptyset$
  $\mu, \hat{I}', \hat{J}' \longleftarrow GS(I, J, \mathcal{L})$
  $S \longleftarrow SLP(\mu, \hat{I}', \hat{J}')$
  for $j \in \hat{J}'$ **do**
    Propose $(j, s_{\mu(j)})$ to driver $\mu(j)$
    if $s_{\mu(j)} \succ_{\mu(j)} s'_{\mu(j),j}$ **then**
      Add driver $\mu(j)$ to set $\hat{I}$, add order j to $\hat{J}$.
    else
      Assign order j to PF, $\mu(j) = j$, $\mu(i) = i$
    end if
  end for
  return μ, $\hat{I}$, and $\hat{J}$
end procedure

The reinforced stable matching mechanism, through integrating the above two stages, is given in Algorithm 2. Given available drivers I, available orders J, and preference lists $\mathcal{L}$ as input (Line 1). We first initialize the set of drivers who accept to deliver assigned orders as empty ($\hat{I} = \emptyset$), and initialize the set of accepted orders as empty as well ($\hat{J} = \emptyset$) (Line 2). Then we operate the first stage GS algorithm (Algorithm 1) to compute a initial matching solution, consisting of a matching function μ, matched drivers $\hat{I}'$ and orders $\hat{J}'$, as well as the unmatched drivers $\tilde{I}$ and unmatched orders $\tilde{J}$ (Line 3). Given this matching solution as input, SLP model computes an optimal compensation scheme for each matched drivers $\hat{I}'$ (Line 4). Each order $j \in \hat{J}'$ with computed compensation $s_{\mu(j)}$ is proposed to the matched driver $\mu(j) \in \hat{I}'$, if the driver accepts the order with compensation, we add driver $\mu(j)$ into set $\hat{I}$, and add order j into set $\hat{J}$ (Line 7 and line 8). If the driver $\mu(j)$ rejects the delivery request $(j, s_{\mu(j)})$, the order j is assigned to PFs (Line 9 and line 10). The output of the algorithm is the set of drivers who accept the delivery requests, $\hat{I}$, and the set of accepted orders, $\hat{J}$.

Based on our above definition of reinforced matching stability, in Theorem 1, we prove that the order-driver matching outcome produced by the R-GS mechanism is reinforced stable.

Theorem 1. *The matching result of R-GS is reinforced stable.*

Proof. To demonstrate the reinforced stability of our final assignment, we need to prove the absence of blocking compensation and blocking pairs given compensation.

- No blocking compensation
  Assuming there exists a driver k who prefers being unmatched rather than matched with assigned order $\mu(k)$ given compensation s. Then according to line 9 of Algorithm 2, she has already rejected the order, which contradicts the assumption. As such, we do not encounter a blocking compensation s.
- No blocking pairs
  Assuming there exists a pair (i, j), where $i \in \hat{I}$, and $j \in \hat{J}$, both i and j prefer each other to the system assigned mate $\mu(i)$, and $\mu(j)$, the relationship can be represented as follows:

$$(j, s'_{i,j}) \succ_i (\mu(i), s_i) \tag{12}$$

  According to the line 13 of Algorithm 2, we have

$$(\mu(i), s_i) \succ_i (\mu(i), s'_{i,\mu(i)}) \tag{13}$$

  Based on preference transitivity, it has:

$$(j, s'_{i,j}) \succ_i (\mu(i), s'_{i,\mu(i)}) \tag{14}$$

  However, referring to the line 5 to line 16 of Algorithm 1, order j must have already proposed to i, and been rejected, which constracts the assumption.
  Similarly, for order j:

$$(i, s'_{i,j}) \succ_j (\mu(j), s_{\mu(j)}) \tag{15}$$

  Considering that the preference of orders is unaffected by the compensation price, we can conclude that if driver i's position in the preference list of j precedes that of driver $\mu(j)$, but according to lines 5 to 16 of Algorithm 1, j must have already been proposed to driver i and subsequently rejected by her. This contradicts the assumption.

Therefore, we have no blocking compensation and pairs. The proof is complete, and the matching result generated by Algorithm 2 is stable. □

5. Questionnaire design and data analysis

To obtain the driver acceptance probability mentioned in the R-GS mechanism in Section 4, we conduct a series of surveys to gather data on different types of drivers' acceptance behaviors towards various orders, aiming to identify the optimal prediction model. In this section, we discuss how we design a questionnaire to gather relevant data on the order acceptance behavior of occasional drivers. Through comparative analysis, we identify the optimal model for predicting the probability of drivers accepting orders. In addition, validation and estimation of ML algorithms are carried out using Python's Scikit-Learn library, while hyperparameter selection for each ML algorithm is achieved via Python's Scikit-Learn library RandomizedSearchCV package.

5.1. Questionnaire design

Building upon our preliminary work (Hou et al., 2022), we further explore the influence of additional external factors and personal attributes on drivers' acceptance behavior. From our preliminary and intuitive observations, the main factors influencing drivers' order acceptance can be the distance of the order from their current location (presented by OA), the travel distance of the

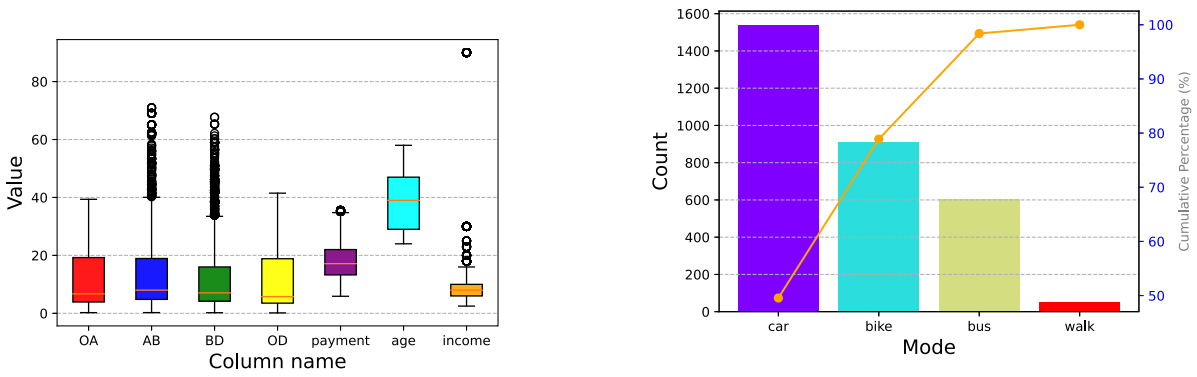
Fig. 4. Questionnaire sample.

order (AB), the detour distance for the driver (BD), the original trip distance of driver (OD), and the compensation amount provided by the system (payment). Other factors might include the size (three levels: small, medium, and large) of the package, the season (two levels: winter and summer) and weather conditions (two levels: sunny and rainy) of the day. Additionally, personal attributes of the driver such as their income, gender, and mode of transportation might also influence their decision to accept or decline an order. Based on these insights, we design a questionnaire with 5000 different sample scenarios, as illustrated in the Fig. 4, and distribute them among various individuals through multiple channels.

5.2. Data description and analysis

After discarding invalid or incomplete surveys, we obtained 308 valid questionnaires, including 3080 order samples from Shanghai, China. Figs. 5(a) and 5(b) summarizes the descriptive statistics of collected data in China between **December 2022 and April 2023**. In the survey, 47.7% of the respondents were female, while 52.6% were male, and 55.94% are rejected by respondents, and 44.06% are accepted.

To predict whether drivers would be willing to accept assigned orders, we systematically compare several commonly used ML algorithms, consisting of Random Forests (RFC), K-neighbor (KNN), Support Vector Machine (SVM), Extreme Gradient Boosting (XGB), Artificial Neural Network (ANN) and the classic binomial logit model (DCM). To facilitate the replication of our work by readers, the optimal hyperparameters for different ML methods are provided. Please refer to Table 4.. We employ the Receiver Operating Characteristic (ROC) curve, to measure the predictive performance and capability of the various behavioral models discussed in this study in forecasting the order acceptance choices of crowd-shippers. The ROC curve of our model, as illustrated in Fig. 6, lies proximate to the upper-left corner, underscoring the model's robust predictive capacity.



(a) Numerical data statistics (b) Categorical data statistics

Fig. 5. Descriptive statistics.

Table 4
ML classifiers parameters tuning (individual attributes and alternative attributes).

Model	Parameter description	Parameter in Sklearn	Value range	Optimal value
Random forest	Measurement of split quality	criterion	$\{gini, entropy\}$	gini
	Maximum depth	max_depth	$\{1, 2, \ldots, 30\}$	10
	The number of trees	n_estimators	$\{20, 21, \ldots, 200\}$	159
	The number of features to consider when looking for the best split	max_features	$\{auto, sqrt, log2\}$	auto
	The minimum number of samples for splitting an internal node	min_samples_split	$\{1, 2, \ldots, 10\}$	4
	The minimum number of samples for being at a leaf node	min_samples_leaf	$\{1, 2, \ldots, 10\}$	2
K-Neighbors	Number of neighbors to use	n_neighbors	$\{1, 2, \ldots, 100\}$	25
Support vector machine	Regularization parameter	C	$\{0, 1, 2, \ldots, 20\}$	3
	Kernal function	kernel	$\{linear, poly, rbf, sigmoid\}$	rbf
	Kernel coefficient	gamma	$\{scale, auto\}$	scale
Extreme Gradient Boosting	Learning rate	learning_rate	$\{0.001, 0.01, 0.1, 1\}$	0.1
	Maximum depth of trees	max_depth	$\{1, 2, \ldots, 20\}$	5
	Number of boosting rounds	n_estimators	$\{1, 2, \ldots, 100\}$	57
	Subsample ratio of the training data	subsample	$\{0.1, 0.2, \ldots, 1\}$	0.8
Artificial neuron network	Batch size	batch_size	$\{32, 64, 128, 256\}$	256
	Learning rate	lr	$\{0.001, 0.01, 0.1\}$	0.01
	Dropout rate	drop_out	$\{0.1, 0.3, 0.5\}$	0.3
	Activation function	activation	$\{relu, sigmoid\}$	relu
	Number of hidden layer	–	$\{1, 2, 3, 4\}$	2
	Number of neurons in each layer	–	$\{32, 64, 128, 256\}$	32, and 64

Furthermore, to prevent overfitting, we randomly split 80% of the survey results as training data and the rest of the observations are regarded as testing data. The training data is utilized to estimate parameters, while the testing data serves to evaluate the model's predictive capacity. To enhance the reliability of the model assessment, we repeated this procedure 200 times and discovered that the results are robust. The F1-score comparative results are given in Table 5

Upon comparison, we find that the XGB algorithm outperforms the other algorithms in terms of prediction accuracy. Therefore, in the computational study, we adopt the XGB mechanism as the behavior model for predicting drivers' order acceptance probability. Since the output of XGB method is a weighted score, we assume that the probability follows a logistic function.

6. Computational study

In this section, we compare the performance of the proposed reinforced stable matching mechanism (R-GS) with two benchmark mechanisms (GS and OPT, presented in Appendix). The GS method serves as a benchmark for deterministic preferences and rational

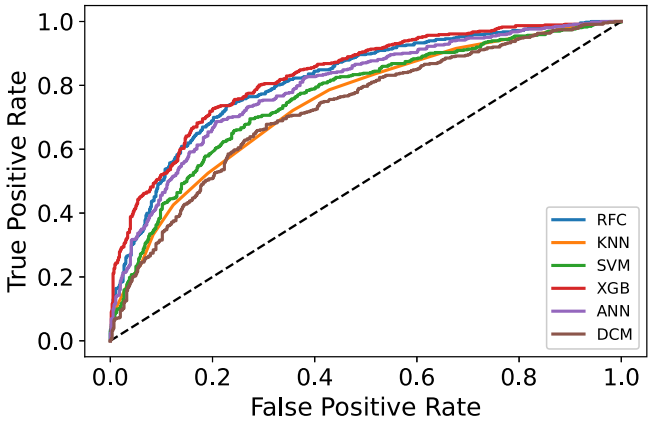


Fig. 6. The ROC curve.

Table 5
F1 score comparison of different ML algorithms.

	mean	std
RF	0.65	0.018
ANN	0.621	0.026
KNN	0.626	0.016
XGB	0.693	0.015
SVM	0.628	0.015
DCM	0.608	0.019

Table 6
Parameter settings.

Parameter	Description	Value
λ	Order arrival rate	20
ι	Average waiting time of each driver	30 min
σ^2	Variance waiting time of each driver	5
C_0	Base delivery cost of professional fleet	¥10
C_1	Base compensation of drivers	¥6
α_0	Delivery cost of professional fleet per kilometer	¥1
α_1	Compensation of drivers per extra kilometer	¥1.1
$\mathcal{V}$	Velocity of professional fleet	40 km/h
v_{car}	Velocity of private car	40 km/h
v_{bus}	Velocity of public bus	20 km/h
v_{bike}	Velocity of bicycle	10 km/h
v_{walk}	Velocity of pedestrian	5 km/h
β_c	Estimated parameter for compensation	0.73
β_d	Estimated parameter for detour distance	0.85
β_0	Intercept value	−4.29
C^l	Penalty for late order	¥3
ω	Highest compensation rate	0.9

decision-making, while OPT represents a classical optimization approach, solving a stochastic integer programming model using the Gurobi commercial solver. This section provides some insights for research in the crowd-sourced delivery and even more in urban mobility field.

6.1. Experimental setup

The experiments are executed on a computer with an Intel Core i7 6-core CPU with 16 GB of RAM, running at 2.6 GHz, using Mac OS X version 11.0.1. The GS algorithm is coded in Python version 3.8.5. The optimization models, including the SLP model and OPT model, were implemented in Gurobi 10.1.0. And the aforementioned parameters, including their descriptions and values, are illustrated in Table 6.

For the sake of generalizability and authenticity, within a circular area centered on the downtown coordinates (31.208366, 121.468460) of **Shanghai city** and with a radius of 40 km, we uniformly generate 20 location coordinates. The creation of map data

Fig. 7. Locations distribution: Driver origins: blue points, driver destination: green points, order pick-up location: red points, order drop-off location: orange points.

utilize the "geopy" module in Python. To simplify the computation of distances between locations, Euclidean distance is employed. Additionally, we uniformly categorize the 20 locations into four equal parts, each marked with a distinct color, to represent drivers' origins, destinations, and the pickup and drop-off locations for orders. This categorization is depicted in Fig. 7.

Experiments are conducted on randomly generated instances. For the drivers, their personal details, including age (years old), following a normal distribution $\mathcal{N}(39, 10)$, transportation mode (car, bus, bike, and walk), drawn from uniform distribution, and gender (male and female), drawn from uniform distribution. Also, income (thousands Chinese Yuan ¥) is drawn from a normal distribution $\mathcal{N}(40, 10)$. Their origins and destinations are uniformly spread across the blue and green points, respectively, as marked on the map. As for the orders, their parcel sizes are uniformly chosen from three distinct levels: large, medium, and small. Their pickup and drop-off locations are uniformly distributed across the red and orange points on the map. Pertaining to weather and seasonal information, each instance is associated with a distinct combination of weather and season. These combinations are uniformly generated from pairs of (summer, winter) and (rainy, sunny) conditions. In addition, we assume order arrivals follows Poisson process with rate λ.

Subsequently, to understand the influence of both symmetrical and asymmetrical order and driver quantities on the metrics, we established sets for the number of drivers ($|I| = \{10, 20, 30, 40, 50\}$) and the number of orders ($|J| = \{20, 40, 60, 80, 100\}$). During the subsequent experimental phase, we combine these data scales at random. For every such combination, we randomly generate 10 instances. The results presented are the averaged values from these 10 instances.

6.2. Evaluation metrics definitions

To compare the performance in system benefits of the proposed mechanism and two benchmark mechanisms, we define the following metrics:

- Order rejection rate:
  As the motivation for this paper, matching stability is the primary subject of investigation. In this regard, we employ the order rejection rate (RR) as the evaluation metric to quantify stability of different matching mechanisms. It is defined as the total number of orders rejected by drivers to the number of orders proposed to drivers.

$$RR = \frac{|\dot{J}|}{|\hat{J}'|} * 100\% \tag{16}$$

- Cost reduction rate:
  This paper adopts the cost reduction rate (CR) as the evaluation metric to manifest cost-effectiveness, which is defined as the ratio of the difference in system costs between adopting CDS and not adopting CDS to the cost of the system without employing CDS.

$$CR = \frac{\sum_{\forall j \in J}(C_j^o + C^l T D_j^o) - [\sum_{\forall j \in \hat{J}}(s_{\mu(j)} + C^l T D_{\mu(j),j}^d) + \sum_{\forall j \in \bar{J}}(C_j^o + C^l T D_j^o)]}{\sum_{\forall j \in J}(C_j^o + C^l T D_j^o)} * 100\% \tag{17}$$

- Order delay rate:
  We employ the order delay rate (DR) as the evaluation metric to gauge the delivery service quality of the matching mechanisms. It is defined as the total number of late orders, delivered by occasional drivers, to the total number of matched orders.

$$DR = \frac{\sum_{\forall j \in \hat{J}} TD_{\mu(j),j}}{|\hat{J}'|} * 100\% \tag{18}$$

6.3. Performance evaluation

Since we do not find an existing approach in the literature that integrates stable matching theory with the consideration of drivers' acceptance uncertainty, we show the benefits of the proposed reinforced stable matching mechanism (R-GS) by comparing its results with those generated by traditional GS algorithm without considering drivers' uncertainty and stochastic mixed integer programming model without considering stable matching. We call these two benchmark approaches as GS and OPT approaches, where GS focus on the matching stability, and OPT approach reflects the traditional view of emphasizing on cost reduction in operation.

Fig. 8 shows the comparison results in terms of the average value of order rejection rate (RR) over the proposed R-GS, and two benchmark approaches GS, and OPT, which is regarded as a primary numerical indicator for measuring stable matching performance. Taking $|I| = 30$, and $|J| = 100$ as an example, the proposed R-GS mechanism can achieve 3.33% average RR while the benchmark approaches OPT and GS achieve 44.67% and 40% average RR respectively. Especially, when $|J| = 40$ and $|I| = 30$, the three approaches show the greatest difference in RR. Compared to the benchmark approaches OPT and GS, R-GS reduce the RR by 58% and 50.66% respectively, showing a significant advantage. In general, our proposed R-GS mechanism significantly outperforms the other two benchmark approaches across various combinations of driver order quantities. The primary reason is that we not only take into account the individual preferences of the drivers but also introduce a tailored compensation mechanism that reinforces the driver's acceptance of orders, which is directly reflected in the RR rate.

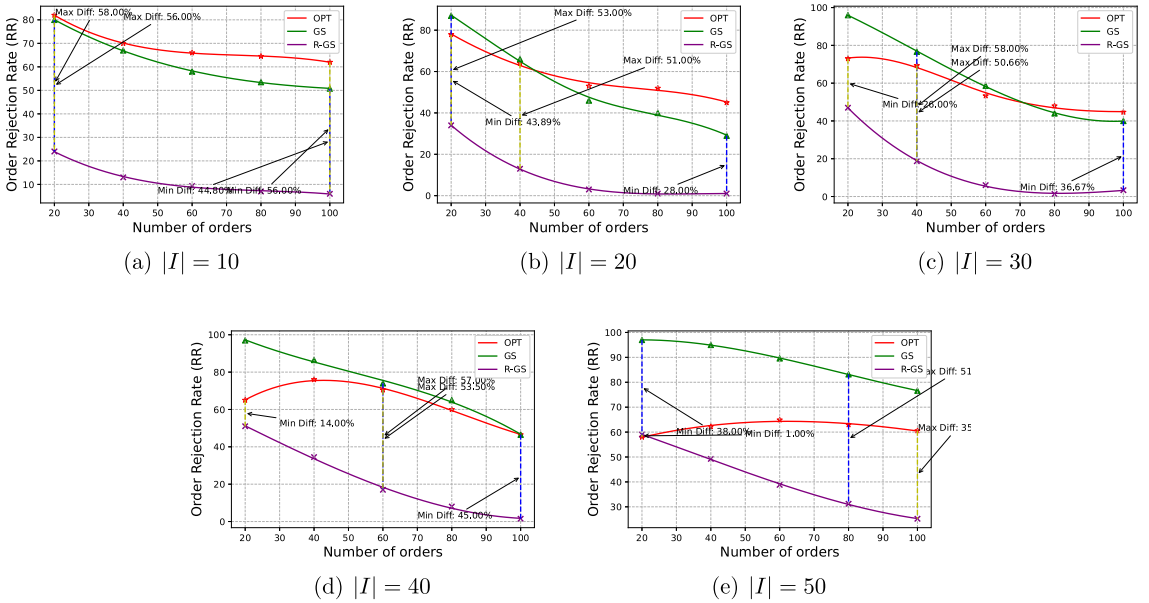
Furthermore, we observe that, while keeping the number of drivers constant and increasing the number of orders, the RR rates of all three approaches exhibit a noticeable declining trend. The rationale behind this is that, whether it is the GS algorithm that considers drivers' order acceptance preferences, the OPT approach that only considers the probability of drivers accepting orders, or the hybrid R-GS mechanism, expanding the pool of available orders for matching makes it easier to find orders that are more suitable for each driver. Additionally, we observe that when the order quantity is approximately over twice the number of drivers ($|J| > 2|I|$), the RR curves of the GS and OPT approaches intersect. Subsequently, the performance of the GS method surpasses that of the OPT approach. The underlying reason for this phenomenon is that when the number of orders is relatively small, the OPT approach tends to find the system's optimal match. Some drivers with longer delivery distances and higher costs might not receive matching requests, leading to a lower RR rate. However, as the number of orders increases, the GS algorithm allows drivers to match with their most preferred orders, thereby achieving a better RR performance.

Fig. 9 shows the comparison results in terms of the average value of cost reduction rate (CR) over the proposed R-GS, and two benchmark approaches GS, and OPT. Specifically, when the number of available drivers is $|I| = 30$ and the number of available orders is $|J| = 40$, the cost reduction rate of the R-GS mechanism reaches nearly 18%, while the GS algorithm and OPT approach have cost reduction rates of approximately 8% and 12%, respectively. In general, the proposed R-GS mechanism significantly outperforms the two benchmark approaches on CR. Specifically, we observe that R-GS exhibits the best CR when the number of orders is slightly greater than the number of drivers.

Moreover, we observe that the OPT approach outperforms the GS algorithm when the number of orders is fewer, and the cost-effectiveness of the three approaches tends to converge as the number of orders continues to grow, keeping the number of drivers constant. The primary reason is that when the order quantity is less than the number of drivers $|J| \leq |I|$, the R-GS mechanism, to ensure that drivers accept the matching results calculated by the GS algorithm, will increase the compensation amount for the drivers, thereby elevating the costs. Moreover, when the order quantity is sufficiently large, even if all drivers agree to deliver an order, the CR rate will not exhibit significant variations.

Fig. 10 shows the comparison results in terms of the average value of order delayed rate (DR) over the proposed R-GS, and two benchmark approaches GS, and OPT. Taking $|I| = 40$, and $|J| = 20$ as an example, R-GS has an average DR of up to 37%, while the other two approaches, OPT and GS have RR of only 1% and 3% respectively. The main reason for this phenomenon is that when the number of drivers exceeds the number of orders $|I| > |J|$, due to our R-GS mechanism prioritizing the acceptance situation of drivers, more compensation is paid to ensure drivers accept the assigned orders (this is also reflected in the comparison of CR). Furthermore, because drivers have varying speeds and need additional distance to pick up orders, compared to the OPT and GS approaches, which do not prioritize increasing the likelihood of drivers accepting orders, most of these assigned orders get rejected and are quickly and directly delivered by the backup PF, resulting in better performance in terms of DR. It is worth noting that when the number of orders exceeds the number of drivers $|I| \leq |J|$, R-GS shows a clear downward trend in terms of DR. When the number of orders is sufficiently large, for instance, $|I| = 20$, and $|J| = 100$, the DR of R-GS drops to the same level as GS at 3%, and the difference with OPT is only 2%.

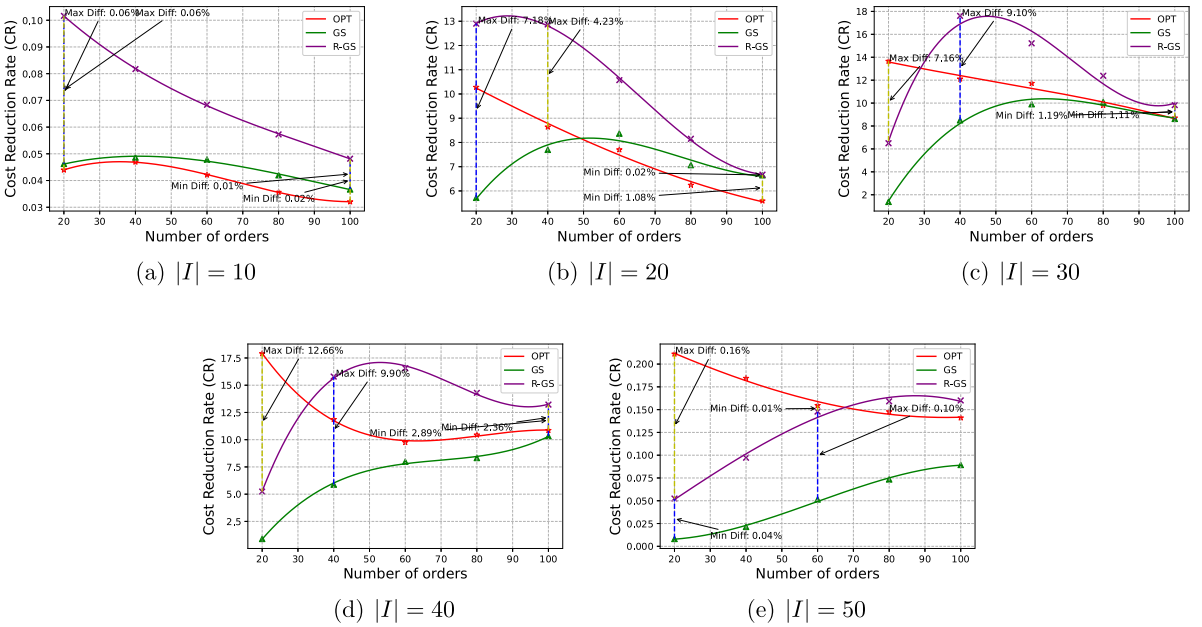
These phenomenons aligns with our original intentions, assumptions, and considerations when designing the R-GS mechanism. Because during the design of the mechanism, by integrating the driver behavior prediction model with the optimization model, we generate a tailored compensation scheme, maximizing the probability of drivers accepting orders. Achieving the lowest order rejection rate is thus unsurprising. At the same time, due to the use of cost effective occasional drivers instead of PFs for deliveries, the overall system cost is also reduced.



(a) $|I| = 10$ (b) $|I| = 20$ (c) $|I| = 30$

(d) $|I| = 40$ (e) $|I| = 50$

Fig. 8. Comparative analysis of Order Rejection Rate (RR) by OPT, GS, and R-GS mechanisms as a function of the number of orders (from 20 to 100) for different numbers of drivers ($|I| = 10, 20, 30, 40, 50$).



(a) $|I| = 10$ (b) $|I| = 20$ (c) $|I| = 30$

(d) $|I| = 40$ (e) $|I| = 50$

Fig. 9. Comparative analysis of Cost Reduction Rate (CR) by OPT, GS, and R-GS mechanisms as a function of the number of orders (from 20 to 100) for different numbers of drivers ($|I| = 10, 20, 30, 40, 50$).

6.4. Extreme value comparison

To verify the stability of the algorithm, we specifically consider a scenario with 10 drivers and 20 orders and observe the extreme value distribution of RR, CR, and DR across three different approaches. Refer to Fig. 11

As observed, the R-GS mechanism demonstrates superior performance in CR and RR compared to the other two methods, though it exhibits greater variance in DR. This variance is attributed to the mechanism's focus on core aspects of crowd-sourced delivery, such as driver experience and overall system efficiency. Overall, this trade-off is considered acceptable

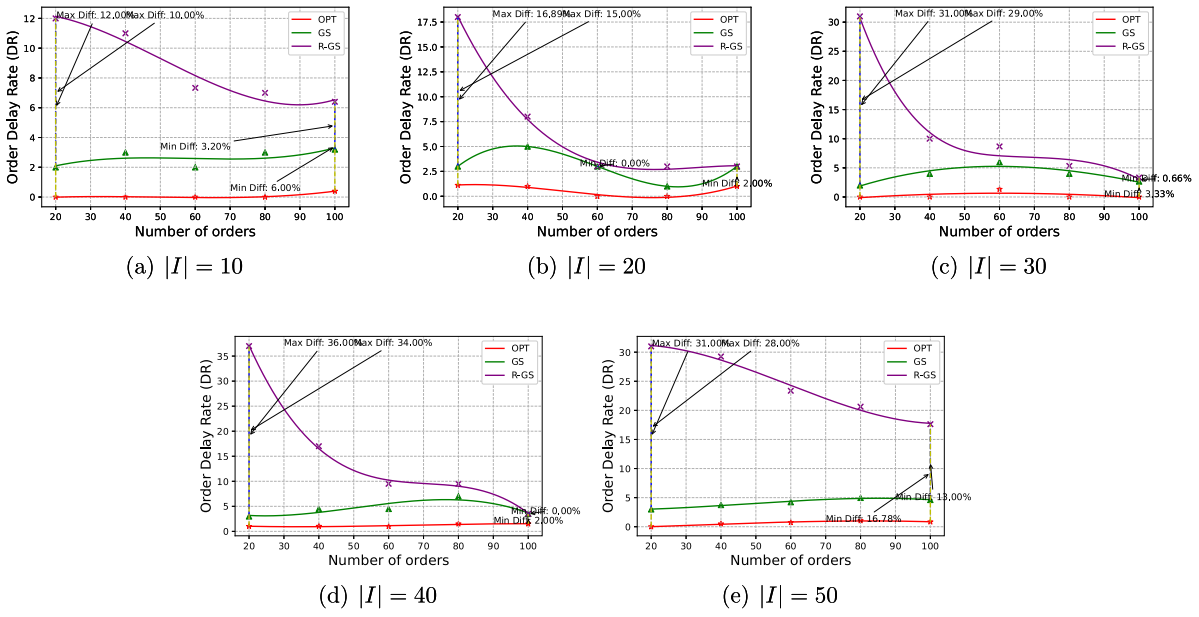


Fig. 10. Comparative analysis of Order Delay Rate (DR) by OPT, GS, and R-GS mechanisms as a function of the number of orders (from 20 to 100) for varying numbers of drivers $|I| = 10, 20, 30, 40, 50$.

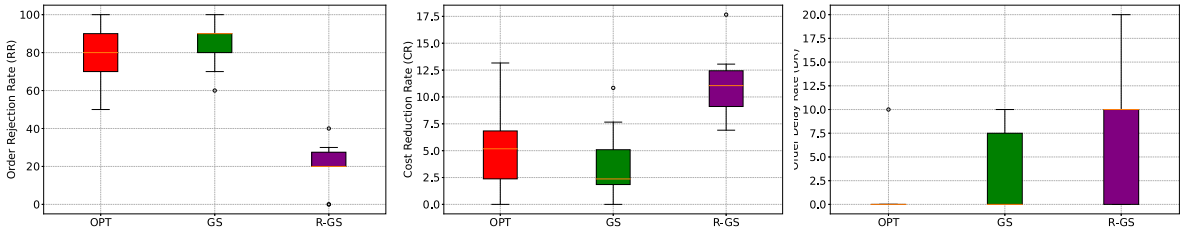


Fig. 11. Comparative analysis of RR, CR, and DR for 10 instances across three different approaches. The central line indicates the median, the box represents the interquartile range (IQR), the whiskers extend to 1.5 times the IQR, and the dots denote outliers.

6.5. Computational efficiency

To test the computational efficiency (reflected by $\Delta\tau$, matching processing time, refer to Section 3) of the proposed mechanisms, we randomly generate ten groups of orders and drivers with different quantities, record and compare the running time of the proposed R-GS mechanism and two benchmark approaches, GS and OPT, and the results are shown in the Fig. 12.

We found that the OPT algorithm with time complexity $O(n)$ is more scalable than the other two algorithms. Both the GS and R-GS algorithms have the same time complexity $O(n^2)$ since they go through a round of deferred acceptance processes. The R-GS mechanism adds an additional optimization process to calculate the new compensation mechanism, resulting in slightly higher running time. Overall, in large-scale computational processes, this running time is acceptable.

7. Conclusion and future work

In this study, we propose an reinforced stable matching mechanism which integrates stable matching theory with a driver behavior model to generate an efficient and stable driver-order matching outcome in the CDS system. This two-phase mechanism first employs the Gale–Shapley (GS) algorithm to achieve an initial stable match. Then, it introduces a stochastic linear programming model in the second phase to create an optimal compensation strategy that reduces expected delivery costs and increases drivers' acceptance rates, leading to an enhanced stable matching outcome we call "Reinforced Stability".

The limitations of this paper lie in the computational efficiency of the proposed reinforced stable matching mechanism. Our future work includes balancing computational efficiency and match stability by developing approximate stable matching mechanisms to adapt to large-scale data environments. Additionally, the generation of preferences for the delivery order requesters is somewhat arbitrary. Generating preferences based on relevant surveys and statistics is also a necessary aspect of our future efforts. In this study, we have primarily considered the impact of numerical values, such as detour distance and compensation price, on driver

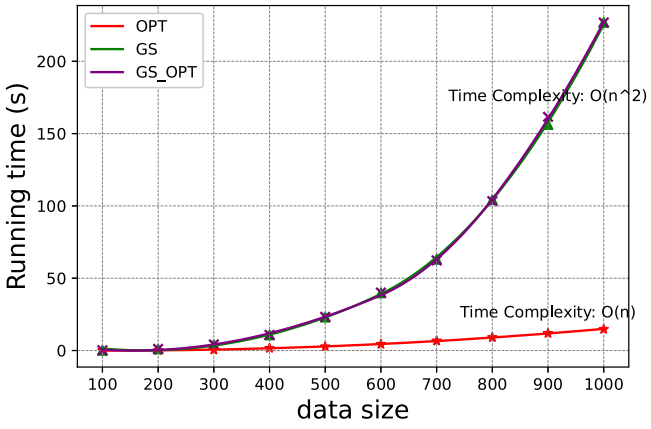


Fig. 12. Computational efficiency, running time.

acceptance behavior. However, in reality, factors such as the geographical characteristics of the orders and the travel routes of the drivers can also significantly influence their decisions. Therefore, our subsequent survey analysis will take these additional factors into account.

We will extend the reinforced stable matching mechanism to accommodate scenarios where drivers may accept multiple orders, effectively exploring a one-to-many delivery model within the CDS framework. Different from the classic one-to-many stable matching, in CDS, subsequent order acceptance behaviors are influenced by the characteristics of the orders already accepted. Moreover, given that a driver's order acceptance behavior could be influenced by certain temporal factors, the associated data collection, analysis, and the consequent design of the matching mechanism also stand out as interesting research subjects. The design and visualization of a simulation model incorporating uncertain driver acceptance behavior is also a key focus of our subsequent work.

CRediT authorship contribution statement

Shixuan Hou: Writing – original draft, Methodology, Conceptualization. **Chun Wang:** Supervision. **Jie Gao:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

We propose a mixed-integer linear programming model employed for computing the optimal order-driver matching outcomes. This model also takes into account the uncertainty associated with drivers' order acceptance behavior, which serves as a benchmark mechanism and is compared against the R-GS mechanism mentioned in the main text. Regarding the notations and their description mentioned in the appendix, readers are referred to Table 3.

The objective function seeks to find the optimal driver-request matching results that minimize the overall expected cost for the system. Let G_1 represent the expected cost of deliveries performed by occasional drivers, consisting of the compensation paid to each driver and the penalty costs, with probability $p^a_{i,j}$, G_2 represent the expected cost of deliveries performed by backup PF, consisting of the compensation paid to each PF and the penalty costs, while the orders are rejected by the occasional driver with probability $p^n_{i,j} = 1 - p^a_{i,j}$, and G_3 represent the cost of unmatched orders delivered by PF, also consisting of the compensation paid to each PF and penalty costs. The decision variable $x_{i,j}$, within the matching model, represents the matching results between drivers and requests.

$$\min_{\boldsymbol{x}} G_1(x) + G_2(x) + G_3(x) \tag{19}$$

$$G_1(x) = \sum_{\forall i \in I} \sum_{\forall j \in J} (s'_{i,j} + C^l T D^d_{i,j}) p^a_{i,j} x_{i,j} \tag{20}$$

$$G_2(x) = \sum_{\forall i \in I} \sum_{\forall j \in J} (C^o_j + C^l T D^o_j) p^n_{i,j} x_{i,j} \tag{21}$$

$$G_3(x) = \sum_{\forall j \in J} (C^o_j + C^l T D^o_j)(1 - \sum_{\forall i \in I} x_{i,j}) \tag{22}$$

subject to:

$$\sum_{\forall i \in I} x_{i,j} \leq 1, \quad \forall j \in J \tag{23}$$

$$\sum_{\forall j \in J} x_{i,j} \leq 1, \quad \forall i \in I \tag{24}$$

$$x_{i,j} = \{0,1\}, \quad \begin{cases} \forall i \in I \\ \forall j \in J \end{cases} \tag{25}$$

Eq. (23) ensure each order can only be assigned to as much as one driver, while Eq. (24) ensure each driver can only deliver no more than one order. And Eq. (25) define the decision variable.

Solving the optimization problem determines the optimal driver-order matchings, reported by a mapping $\mu : I \cup J \longrightarrow I \cup J$, if an order j is assigned to a driver $x_{i,j} = 1$ let i be in set $\hat{I}'$, and let j be in set $\hat{J}'$, $\mu(i) = j$, and $\mu(j) = i$. The orders, not assigned to drivers, are proceed by PF. Let the orders be in set $\tilde{J}$, if $\sum_{\forall i \in I} x_{i,j} = 0, \ \forall j \in J$.

References

Al-Saudi, A., Himpel, F., 2020. Crowd logistics delivery determinants: A stated-preference survey.

Allahviranloo, M., Baghestani, A., 2019. A dynamic crowdshipping model and daily travel behavior. Transp. Res. E 128, 175–190.

Alnaggar, A., Gzara, F., Bookbinder, J.H., 2021. Crowdsourced delivery: A review of platforms and academic literature. Omega 98, 102139.

Archetti, C., Savelsbergh, M., Speranza, M.G., 2016. The vehicle routing problem with occasional drivers. European J. Oper. Res. 254 (2), 472–480.

Arslan, A.M., Agatz, N., Kroon, L., Zuidwijk, R., 2019. Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. Transp. Sci. 53 (1), 222–235.

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. Res. Transp. Bus. Manage. 37, 100516.

Aziz, H., Biró, P., Gaspers, S., de Haan, R., Mattei, N., Rastegari, B., 2020. Stable matching with uncertain linear preferences. Algorithmica 82, 1410–1433.

Balinski, M., Sönmez, T., 1999. A tale of two mechanisms: student placement. J. Econ. Theory 84 (1), 73–94.

Behrend, M., Meisel, F., 2018. The integration of item-sharing and crowdshipping: Can collaborative consumption be pushed by delivering through the crowd? Transp. Res. B 111, 227–243.

Behrend, M., Meisel, F., Fagerholt, K., Andersson, H., 2019. An exact solution method for the capacitated item-sharing and crowdshipping problem. European J. Oper. Res. 279 (2), 589–604.

Biró, P., Fleiner, T., Irving, R.W., Manlove, D.F., 2010. The college admissions problem with lower and common quotas. Theoret. Comput. Sci. 411 (34–36), 3136–3153.

Boysen, N., Emde, S., Schwerdfeger, S., 2022. Crowdshipping by employees of distribution centers: Optimization approaches for matching supply and demand. European J. Oper. Res. 296 (2), 539–556.

Chen, W., Mes, M., Schutten, M., 2018. Multi-hop driver-parcel matching problem with time windows. Flex. Serv. Manuf. J. 30, 517–553.

Dayarian, I., Savelsbergh, M., 2020. Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. Prod. Oper. Manage. 29 (9), 2153–2174.

Gale, D., Shapley, L.S., 1962. College admissions and the stability of marriage. Amer. Math. Monthly 69 (1), 9–15.

Gdowska, K., Viana, A., Pedroso, J.P., 2018. Stochastic last-mile delivery with crowdshipping. Transp. Res. Proc. 30, 90–100.

Hou, S., 2019. A Two-sided Matching System Design for Dynamic Labor Markets (Ph.D. thesis). Concordia Institution for information systems engineering.

Hou, S., Gao, J., Wang, C., 2022. Optimization framework for crowd-sourced delivery services with the consideration of shippers' acceptance uncertainties. IEEE Trans. Intell. Transp. Syst. 24 (1), 684–693.

Le, T.V., Stathopoulos, A., Van Woensel, T., Ukkusuri, S.V., 2019. Supply, demand, operations, and management of crowd-shipping services: A review and empirical evidence. Transp. Res. C 103, 83–103.

Le, T.V., Ukkusuri, S.V., 2019. Influencing factors that determine the usage of the crowd-shipping services. Transp. Res. Rec. 2673 (7), 550–566.

Le, T.V., Ukkusuri, S.V., Xue, J., Van Woensel, T., 2021. Designing pricing and compensation schemes by integrating matching and routing models for crowd-shipping systems. Transp. Res. E 149, 102209.

Liu, Q., Mailath, G.J., Postlewaite, A., Samuelson, L., 2014. Stable matching with incomplete information. Econometrica 82 (2), 541–587.

Macrina, G., Pugliese, L.D., Guerriero, F., 2020. Crowd-shipping: a new efficient and eco-friendly delivery strategy. Procedia Manuf. 42, 483–487.

Manlove, D., 2013. Algorithmics of Matching Under Preferences, vol. 2, World Scientific.

Mousavi, K., Bodur, M., Roorda, M.J., 2022. Stochastic last-mile delivery with crowd-shipping and mobile depots. Transp. Sci. 56 (3), 612–630.

Peng, Z., Shan, W., Jia, P., Yu, B., Jiang, Y., Yao, B., 2020. Stable ride-sharing matching for the commuters with payment design. Transportation 47, 1–21.

Pourrahmani, E., Jaller, M., 2021. Crowdshipping in last mile deliveries: Operational challenges and research opportunities. Soc.-Econ. Plan. Sci. 78, 101063.

Punel, A., Ermagun, A., Stathopoulos, A., 2018. Studying determinants of crowd-shipping use. Travel Behav. Soc. 12, 30–40.

Rasulkhani, S., Chow, J.Y., 2019. Route-cost-assignment with joint user and operator behavior as a many-to-one stable matching assignment game. Transp. Res. B 124, 60–81.

Roth, A.E., Peranson, E., 1999. The redesign of the matching market for American physicians: Some engineering aspects of economic design. Am. Econ. Rev. 89 (4), 748–780.

Roth, A.E., Sönmez, T., Ünver, M.U., 2005a. A kidney exchange clearinghouse in new England. Amer. Econ. Rev. 95 (2), 376–380.

Roth, A.E., Sönmez, T., Ünver, M.U., 2005b. Pairwise kidney exchange. J. Econ. Theory 125 (2), 151–188.

Said, C., 2021. Uber May Stop Letting Drivers See Destinations and Name Prices. San Francisco Chronicle, https://www.sfchronicle.com/business/article/Uber-may-stop-letting-drivers-see-destinations-16078491.php.

Sampaio, A., Savelsbergh, M., Veelenturf, L.P., Van Woensel, T., 2020. Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers. Networks 76 (2), 232–255.

Shurrab, M., Singh, S., Otrok, H., Mizouni, R., Khadkikar, V., Zeineldin, H., 2021. A stable matching game for V2V energy sharing–a user satisfaction framework. IEEE Trans. Intell. Transp. Syst. 23 (7), 7601–7613.

Sühr, T., Biega, A.J., Zehlike, M., Gummadi, K.P., Chakraborty, A., 2019. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 3082–3092.

Torres, F., Gendreau, M., Rei, W., 2022. Vehicle routing with stochastic supply of crowd vehicles and time windows. Transp. Sci. 56 (3), 631–653.

Tu, W., Zhao, T., Zhou, B., Jiang, J., Xia, J., Li, Q., 2019. OCD: Online crowdsourced delivery for on-demand food. IEEE Internet Things J. 7 (8), 6842–6854.

Wang, X., Agatz, N., Erera, A., 2018. Stable matching for dynamic ride-sharing systems. Transp. Sci. 52 (4), 850–867.

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.
Yan, P., Lee, C.-Y., Chu, C., Chen, C., Luo, Z., 2021. Matching and pricing in ride-sharing: Optimality, stability, and financial sustainability. Omega 102, 102351.
Zhang, N., Liu, Z., Li, F., Xu, Z., Chen, Z., 2023. Stable matching for crowdsourcing last-mile delivery. IEEE Trans. Intell. Transp. Syst..
Zhang, H., Zhao, J., 2018. Mobility sharing as a preference matching problem. IEEE Trans. Intell. Transp. Syst. 20 (7), 2584–2592.
Zhao, B., Xu, P., Shi, Y., Tong, Y., Zhou, Z., Zeng, Y., 2019. Preference-aware task assignment in on-demand taxi dispatching: An online stable matching approach. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33, pp. 2245–2252.
Zhou, Z., Liu, P., Feng, J., Zhang, Y., Mumtaz, S., Rodriguez, J., 2019. Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach. IEEE Trans. Veh. Technol. 68 (4), 3113–3125.