Optimizing OD-based up-front discounting strategies for enroute ridepooling services

Siying Wang\textsuperscript{a}, Xiaolei Wang\textsuperscript{a,*}, Chen Yang\textsuperscript{a}, Wei Liu\textsuperscript{b}, and Xiaoning Zhang\textsuperscript{a}

\textsuperscript{a}School of Economics and Management, Tongji University, Shanghai, 200092, P.R. China
\textsuperscript{b}Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong, China

\textbf{Keywords:}
- enroute ridepooling
- up-front pricing strategy
- derivative-free optimization

The technological progress in the recent decade has greatly facilitated the large-scale implementation of dynamic enroute ridepooling services, such as Uber Pool and DiDi Pinche. To sustain a profitable dynamic ridepooling service, a well-designed pricing scheme is crucial. This paper focuses on the optimization of up-front discounting strategies for dynamic enroute ridepooling service, under which passengers are notified of origin-destination(OD)-based discount ratios together with estimated ride time before the start of their trips and enjoy the discounted prices no matter if they succeed or fail to get matched afterward. Assuming that ridepooling demand of each OD pair decreases with its price and the estimated waiting and ride time, we propose to optimize the discounting strategy of each OD pair through two methods. In the first method, the ridepooling price of each OD pair is optimized independently and adjusted day-to-day based on historical information; and in the second method, we optimize the prices of all OD pairs simultaneously, with the complex interactions among the expected ride and waiting times and the demand rates of all OD pairs being considered and captured by a system of nonlinear equations. The nonlinear and non-convex optimization problem of the second method is solved by two derivative-free algorithms: Bayesian optimization and classification-based optimization. Based on a 15*15 grid network with 30 OD pairs and the real road network of Haikou (China), we examine the efficiency of the two algorithms and the system performance under different pricing strategies derived from the two methods. It is found that in comparison with a uniform discounting strategy, OD-based discounting strategies generated by both methods can bring about 10% more profit to the platform. In comparison with the independently optimized discounting strategies generated by the first method, the system optimal discounting strategy generated by the second method can further improve the platform profit by 7.89% and 7.67% on average in our grid-network and real road network experiments.

* Corresponding author. E-mail address: xlwangiris0420@gmail.com.
1 Introduction

The rapid development of GPS positioning and wireless communication technologies and the popularity of smartphones in the recent decade have greatly facilitated large-scale implementations of enroute ridepooling services, such as Uber Pool, DiDi Pinche. Unlike in previous ridepooling services where vehicles depart after ridepooling trips are formed, an enroute ridepooling platform responds to on-demand ridepooling requests in real-time, dispatches vacant or partially occupied vehicles to pick up passengers immediately, and keeps searching for ridepooling partners on the fly. The on-demand feature of enroute ridepooling services significantly enhances the appeal of ridepooling.

A well-designed pricing strategy is essential to the vitality and profitability of enroute ridepooling services. There are two primary pricing modes adopted by existing ridepooling platforms in reality: post-trip pricing and up-front pricing. In the post-trip pricing mode, ridepooling platforms calculate the price based on the actual pairing results after the trips have concluded. So the prices that passengers have to pay for their trips are determined at the end of their journey. And in the up-front pricing mode, passengers are informed of prices and estimated ride times before the commencement of their trips, and pay the prices regardless of whether they are successfully matched or not. In this paper, we concentrate on the up-front pricing mode, which aligns with the pricing approach utilized by ridepooling platforms such as UberPool and DiDi Pinche in practice (Shaheen and Cohen, 2019).

The price of ridepooling is typically anchored to that of solo-ride service, with a discount ratio offered to compensate for the detour costs that passengers may incur due to pooling. And as ridepooling orders between different origin-destination (OD) pairs have different possibilities to be successfully paired, it is intuitive to offer disparate discount ratios for orders between different OD pairs. However, determining an appropriate discounted price for each enroute ridepooling order before the trip commences is far from an easy task. On one hand, the discount ratio and the expected ride time offered by the platform govern passengers’ mode choices, which determine the ridepooling demand rate of each OD pair; on the other hand, the expected ride time of each OD pair, and the expected shared distance of each OD pair, which are determinants of the platform’s profit, change with the ridepooling demand distribution over the network. So, how to predict the impacts of a discounting strategy on the expected ride time of each OD pair through its influence on the ridepooling demand distribution over the network serves as a first challenge. Second, the discount ratio of one OD pair has systematic impacts on the demand rates of other OD pairs, so the platform should carefully coordinate the discount ratios of different OD pairs. For example, consider the simple network as in Fig. 1, orders between OD pair (A,C) can form ridepooling trips with both orders between (A,B) and orders between (B,C). And pooling (A,C) trips with (B,C) yields a longer distance saving. Suppose every passenger is willing to share vehicle space with at most one another during the entire trip, then decreasing the discount ratio for (A,B) orders would increase the ridepooling demand between (A,B) and lower the pairing probability between (A,C) and (B,C). Because in this case, orders between (A,C) become more likely to be matched with (A,B) orders rather than (B,C) orders. So to maximize the platform’s profit or social welfare, the platform should optimize the price discounts of all OD pairs simultaneously, considering the complex matching and competing relationships among different OD pairs.

There have been many studies of enroute ridepooling services in the recent decade, but most of them focus on the optimization of dispatching strategies (e.g., Lin et al., 2012; Santos and Xavier, 2013; Hosni et al., 2014; Ma et al., 2014; Asghari et al., 2016; Tong et al., 2018). Existing studies for the pricing strategy design of enroute ridepooling services are quite few. Some aggregate market equilibrium models are proposed in the literature (e.g., Ke et al., 2020; Zhang and Nie, 2021) to characterize the ridepooling market and study the optimal price rate for ridepooling service. Ke et al. (2020) investigated the equilibrium
of ride-sourcing markets with and without on-demand ridepooling services to analyze the relationships between the platform decision variables (i.e., trip fare, vehicle fleet size and allowable detour time) and the system’s endogenous variables (e.g., pick-up time, passenger demand, successful matching rate and actual detour time). And they examined the optimal trip fares in monopoly optimum, social optimum and second-best solution conditions in pooling and non-pooling markets. Zhang and Nie (2021) formulated the spatial matching-based market equilibrium model of a ride-sourcing market with both solo and pooling services while competing with transit. They examined the platform’s optimal pricing decision while maximizing the profit and social welfare under different regulations. However, such aggregate models determine an optimal average (or uniform) price rate for all ridepooling orders considering neither the disparate pairing potential of each OD pair nor the impacts of the detailed pricing mode. To our knowledge, Li et al. (2022) is the only work that investigates upfront discounting strategies of ride-sourcing services. Provided the ridepooling and non-pooling price rates per unit time, they model passengers’ mode choices considering the variability of waiting time, ride time and detour time, and compare the platform’s profit under both up-front pricing and post-trip pricing modes. Based on the model, they pointed out that up-front pricing can produce more profit and induce more ridepooling demand if the matching probability of each OD pair under an up-front pricing strategy can be accurately given. However, in that work, they simply assume the matching probability of each OD pair to be determined by its own demand rate, and the discount ratio for ridepooling service is universal for all OD pairs. So the complex interactions among ridepooling orders between different OD pairs in enroute ridepooling service are completely ignored in Li et al. (2022), and the obtained optimal uniform discount ratio is not as efficient as an OD-based discounting strategy, as will be shown in our numerical experiments in Section 5. Recently, Wang et al. (2021) proposed a system of nonlinear equations to describe the complex interactions among ridepooling orders between different OD pairs, and demonstrates its satisfactory accuracy in predicting the disparate pairing potentials of orders between different OD pairs in a general road network under given ridepooling demand distribution. This shed lights on the design of OD-based upfront discounting strategies considering the systematic impacts of a discounting strategy on the ridepooling demand distribution, the platform profit, and the ride time of each OD pair. However, in Wang et al. (2021), the demand rate between each OD pair is assumed to be given, and the pick-up time is assumed to be constant, so the model can not capture the influence of the platform’s pricing strategies and the total vehicle supply on the system performance.

In view of the above research gap, this paper makes the first attempt to optimize the OD-based up-front discounting strategies for enroute ridepooling service, considering the disparate pairing potentials of different OD pairs and the systematic impacts of each discounting strategy. Assuming that the ridepooling
demand of each OD pair decreases with its price and the expected ride time, we first extend the model proposed in Wang et al. (2021) to depict the complex interactions among the ridepooling demand distribution and vacant vehicle distribution over the network and the expected passenger waiting time, ride time and vehicle sharing time of orders between each OD pair, so as to calculate the market performance and platform profit at equilibrium under given pricing strategies and vehicle fleet size. Then, we propose two methods for upfront pricing strategy design of enroute ridepooling orders between each OD pair. In the first method (Method 1), the ridepooling discount ratio of each OD pair is optimized independently based on historical pairing potential information, and requires a day-to-day adjustment; and in the second model (Method 2), we optimize the prices of all OD pairs simultaneously, considering the complex interactions among the platform’s discounting strategy and the system performance at equilibrium. As the second model is highly nonlinear and non-convex, we propose two derivative-free algorithms (Bayesian optimization (Mockus, 1994) and the Sequential randomized coordinate shrinking classification algorithm (Hu et al., 2017)) to solve it. Through a large number of simulation experiments based on a 15*15 grid network and the real road network of Haikou, China, we show that in comparison with a uniform discounting strategy, OD-based discounting strategies generated by both methods can bring about 10% more profit to the platform. In comparison with the independently optimized discounting strategies generated by Method 1, the system optimal discounting strategy generated by Method 2 can further improve the platform profit by 7.89% and 7.67% on average in our small-network and large-network-based experiments.

The contribution of this work is three-fold:

- First, this paper proposes a model to describe the systematic impacts of any upfront pricing strategy, considering the complex interactions among the ridepooling demand distribution and vacant vehicle distribution over the network, and the expected passenger waiting time, ride time and vehicle sharing time of orders between each OD pair. In comparison with the one proposed in Wang et al. (2021), our model not only captures the ridepooling demand elasticity with respect to ridepooling discount ratio, waiting time and ride time, but also incorporates the impacts of vehicle supply on passenger waiting time and trip demand.

- Second, this paper is among the first for the design of OD-based up-front pricing strategies for enroute ridepooling service. Two methods, with one optimizing the discount ratio of each OD pair independently (Method 1), and the other one optimizing the discount ratios of all OD pairs in a coordinate manner (Method 2), are examined and compared. Furthermore, for the nonlinear and non-convex optimization problem in Method 2, the applicability of two derivative-free solution methods (Bayesian optimization and classification-based optimization) is examined and validated.

- Third, through extensive simulation experiments based on a grid network and a real road network, this paper reveals the significant improvement that can be brought about by well-designed OD-based discounting strategies in comparison with uniform discounting strategies. Furthermore, the discounting strategy generated by Method 2 which coordinates the ridepooling demand distribution over the network is shown to bring more platform profit than that generated by Method 1.

The remainder of the paper is organized as follows. Section 2 models passengers’ mode choices and the expected system performance at equilibrium under any up-front pricing strategy, and establishes the existence of equilibrium solutions. Section 3 proposes two methods for ridepooling pricing strategy design, with the first one optimizing the discount ratio of each OD pair independently, and the second one optimizing the discount ratio of all OD pairs simultaneously. Section 4 introduces two derivative-free algorithms to
solve the model of Method 2. Section 5 examines the efficiency of the two algorithms and the system performance under the optimal pricing strategies of proposed models and benchmark uniform pricing strategy through numerical experiments. Section 6 concludes the paper and discusses possible directions for future.

2 Modelling the expected system performance under any OD-specific ridepooling pricing strategy

2.1 Problem setup

Consider a ridepooling platform that operates a number of n vehicles to provide enroute ridepooling service on a general road network G(N, A), with N being the set of nodes and A being the set of links. The traffic condition is assumed to be steady during the study period, so the travel time \( t_a \) on each link \( a \in A \) is assumed to be constant. Let \( W \) denote the set of OD pairs on the network, and \( \bar{\lambda}_w \) be the maximal ridepooling demand rate for OD pair \( w \in W \).

For each OD pair, the platform sets the price of pooling services by referencing the price of solo-ride services. Let \( p^s_w \) be the price and ride time of solo-ride service between OD pair \( w \in W \), which is assumed to be exogenous and publicly known. Upon receiving the origin and destination information from a potential ridepooling passenger, the platform immediately announces an up-front discounted price \( \theta_w p^s_w \) for ridepooling orders between OD pair \( w \in W \), together with estimated waiting time and ride time, indicated by \( t_w \) and \( T^p_w \) respectively. The expected ridepooling cost between OD pair \( w \in W \), denoted by \( C^p_w \), is thus

\[
C^p_w = \theta_w p^s_w + \beta_w (t_w + T^p_w) + \Delta
\]

where \( \beta_w \) indicates the average value of time (VOT) of passengers between OD pair \( w \in W \), and \( \Delta \) is the perceived safety cost due to ridepooling. E-hailing passengers compare the expected ridepooling cost with solo-ride cost to determine whether or not to adopt ridepooling mode. So in this study, we assume the ridepooling demand rate between each OD pair \( w \in W \), denoted by \( \lambda^p_w \), is dependent on the maximum potential ridepooling demand rate \( \bar{\lambda}_w \) and the cost difference \( C^p_w - C^s_w \):

\[
\lambda^p_w = \bar{\lambda}_w f_w(C^p_w - C^s_w)
\]

where \( C^s_w \) is the generalized solo-ride cost between OD pair \( w \in W \), which is exogenous, \( f_w(x) \) is a continuous, differentiable and decreasing function that satisfies \( f_w(x) \in [\varepsilon, 1] \) for \( x \in \mathbb{R} \) (here \( \varepsilon \) is a positive constant which can be sufficiently small). The value of \( f_w(C^p_w - C^s_w) \) gives the proportion of e-hailing passengers that choose ridepooling service between OD pair \( w \in W \) when the cost difference between pooling and non-pooling service is \( C^p_w - C^s_w \).

The platform responds to ridepooling orders in real time and adopts the first-protocol matching strategy. As in Daganzo and Ouyang (2019) and Wang et al. (2021), we assume ridepooling passengers share vehicle space with at most one another during the entire trip, and every order contains only one passenger, so the words ’passenger’ and ’order’ are used interchangeably throughout the paper. Every ridepooling passenger will be first considered for pick-up by partially occupied vehicles. If there are no matching partially occupied

---

1 In reality, the price of non-pooling service can be easily obtained by querying a non-pooling platform.

2 The constant \( \varepsilon \) is introduced here to ensure the existence of equilibrium state in Subsection 2.3.
vehicles nearby, the platform dispatches the nearest vacant vehicle to pick up the waiting passenger. And if multiple matching partially occupied vehicles are found during one search, we assume the platform has predetermined priority rules (for example, in increasing the order of the pick-up distances or decreasing the order of saved distances) to determine the vehicle with the highest priority. For every waiting passenger, a partially occupied vehicle is said to be a matching vehicle only if the on-board passenger and the waiting passenger can form a ridepooling trip that satisfies the following pairing conditions: 1) the pick-up time for the waiting passenger does not exceed a maximum value $\bar{t}_r$, and 2) the detour time for both passengers does not exceed a maximum value $\bar{t}_d$.

Under the above matching strategy, ridepooling passengers could be picked up by partially occupied vehicles or vacant vehicles; and they may be successfully paired at their origins or during their trips, or arrive at their destinations without getting paired. As pointed out in Wang et al. (2021), the expected pairing probability and ride time of ridepooling orders between each OD pair is dependent on the ridepooling demand distribution over the network, and the expected waiting time of each OD pair is affected by the distribution of vacant vehicles and their matching partially occupied vehicles over the network. So in Eq. (1), both $t_w$ and $T_{wp}$, $w \in W$ are endogenously determined.

In the following subsection, we investigate how the ridepooling demand distribution $\lambda^p = (\lambda_w^p, w \in W)$ over the network and the vehicle fleet size $n$ determine the waiting time $t_w$ and ride time $T_{wp}$ of ridepooling orders between each OD pair $w \in W$. Recall the impacts of the ridepooling waiting time $t_w$, ride time $T_{wp}$ and the platform’s discounting strategy $\theta_w$ on ridepooling demand rate $\lambda_w^p$, $w \in W$ as described in Eq. (1) and (2), we then obtain a system of nonlinear equations to describe the expected system performance (i.e., the distribution of ridepooling demand and vacant vehicles over the network, the passenger waiting time and ride time between each OD pair) under any OD-based discounting strategy $\theta = (\theta_w, w \in W)$, as given in Subsection 2.3. To facilitate reading, a list of notations is provided in Appendix A.

2.2 Modelling the expected waiting time and ride time of each OD pair

In this subsection, we model the expected waiting time and ride time of ridepooling orders between each OD pair provided the ridepooling demand rate of each OD pair $\lambda^p$ and vehicle fleet size $n$. Note that, under the matching strategies mentioned above, the time taken for passenger-vehicle matching is negligible. Therefore, in our study, the passenger waiting time refers exclusively to the time spent waiting for pick-up. Furthermore, ridepooling orders could be matched to either partially occupied vehicles or vacant vehicles, so the expected waiting time is dependent on both the pick-up time by partially occupied vehicles and the pick-up time by vacant vehicles. We thus introduce $\bar{t}_{pk}$, $\bar{t}_{wp}$, $\bar{t}_w$ and $\bar{t}_d$ to respectively denote the pick-up time of ridepooling orders between OD pair $w \in W$ by partially occupied vehicles and by vacant vehicles. In Wang et al. (2021), they assume the pick-up time by vacant vehicles $\bar{t}_{wp} = (\bar{t}_{wp}, w \in W)$ is constant and propose a model that can accurately predict the pairing probability, ride time and vehicle sharing time of ridepooling orders between each OD pair under given pick-up time $\bar{t}_{pk}$ and demand rate $\lambda^p$. So in the following subsection 2.2.1, we first introduce the model proposed in Wang et al. (2021) to describe the expected ride time of ridepooling orders between each OD pair under given pick-up time $\bar{t}_{pk}$ and demand rate $\lambda^p$. In subsection 2.2.2, we investigate how the demand rate of each OD pair, the total vehicle fleet size, the expected matching results and the flow of vacant vehicles affect the distribution of vacant vehicles over the network so as to determine the expected pick-up time by vacant vehicles $\bar{t}_{wp}$. By examining the endogenous determinants of $\bar{t}_{pk}$, we relax the assumption in Wang et al. (2021) that the pick-up time by vacant vehicles is constant.
2.2.1 The mean ride time of ridepooling orders between each OD pair

We now give a brief introduction about how the model proposed in Wang et al. (2021) captures the complex interactions among ridepooling demand between different OD pairs, generates the expected ride time and vehicle sharing time of each OD pair under any given demand rate $\lambda^p$ and pick-up time by vacant vehicles $\pi^p$.

First, under the matching strategies described in Subsection 2.1, passengers, before getting paired with others, can be classified into two types: 1) “seekers” who just submit orders and the platform searches vehicles for them, and 2) “takers” who have been assigned vehicles but haven’t gotten paired with other passengers. For each OD pair $w \in W$, those newly appearing passengers are defined as passengers in the seeker-state $s(w)$, and those on-board but unpaired passengers traveling on link $a \in A_w$ are defined as passengers in the taker-state $t(a, w)$, where $A_w$ is the set of links on the path$^3$ between OD pair $w \in W$. Passengers who are assigned vacant vehicles and waiting at their origins for pick-up are defined as takers in state $t(a^0_w, w)$, where $a^0_w$ represents a dummy link with both of its head and tail nodes corresponding to the origin node $o_w$. Let $\hat{A}_w = A_w \cup \{a^0_w\}$. The set of seeker-states is indicated by $S = \{s(w), w \in W\}$, and the set of taker-states is indicated by $T = \{t(a, w), a \in \hat{A}_w, w \in W\}$. Let $\tilde{t}_{t(a,w)}$ be the maximal passenger dwelling time in taker-state $t(a, w)$, that is, the maximum time that a passenger in taker-state $t(a, w)$ would stay in this state. For $t(a, w)$ with $a \in A_w$, the maximal passenger dwelling time is the link travel time of the link, denoted by $t_a$, and for $t(a, w)$ with $a = a^0$, the maximal passenger dwelling time in this state is the pick-up time of vacant vehicles $\pi^p_w$. So

$$\tilde{t}_{t(a,w)} = \begin{cases} t_a, & \text{if } a \in A_w \\ \pi^p_w, & \text{if } a = a^0_w, w \in W. \end{cases} \tag{3}$$

The passenger waiting time for pick-up thus influences the passenger maximal dwelling time in the first taker-state $t(a^0, w)$ of each OD pair. Passengers in each taker-state may get paired during their stay in the state or otherwise enter the next taker-state after the maximal dwelling time. According to the pairing conditions, we can determine the set of matching taker-states $T_{s(w)}^n$ for each seeker-state $s(w)$ and the set of matching seeker-states $S_{t(a, w)}$ for each taker-state $t(a, w)$. Furthermore, for each seeker-state $s(w)$ and its pairing taker-state $t(a, \omega) \in T_{s(w)}^n$, $T_{s(w)}^{\omega t(a,\omega)}$ is introduced to indicate the set of taker-states that have a higher matching priority with seeker-state $s(w)$ in comparison with the taker-state $t(a, \omega)$.

As passengers between OD pair $w \in W$ may get paired when they are in different states, to calculate the expected ride time and vehicle sharing time of orders between each OD pair $w \in W$, it is necessary to know the matching probabilities of passengers in each seeker-state and taker-state. To do so, the following five types of variables are defined: (1) $p_s(w)$, that is, the probability of a seeker being matched in seeker-state $s(w)$, and $p_s = (p_s(w), s(w) \in S)$; (2) $p_t(a, w)$, that is, the probability of a taker being matched in taker-state $t(a, w) \in T$, and $p_t = (p_t(a, w), t(a, w) \in T)$; (3) $\rho_t(a, w)$, that is, the probability of having at least one taker in state $t(a, w) \in T$ at any moment, and $\rho = (\rho_t(a, w), t(a, w) \in T)$; (4) $\eta_{s(a, w)}$, that is, the aggregate arrival rate of pairing opportunities with seekers in state $s(w)$ for takers in state $t(a, w) \in T$, and $\eta = (\eta_{s(a, w)}, t(a, w) \in T)$; and (5) $\lambda_{t(a, w)}$, that is, the average arrival rate of unmatched passengers for taker-state $t(a, w) \in T$, and $\lambda_t = (\lambda_{t(a, w)}, t(a, w) \in T)$.

A system of nonlinear equations is proposed to describe the complex interactions among these variables: The probability of passenger existence in every taker-state at any time moment $\rho$ governs the aggre-

---

$^3$Passengers between each OD pair $w \in W$ are assumed to use the same minimum-cost path, unless getting paired with others.
gate arrival rate of pairing opportunities for different taker-states $\eta$, which further affects passengers’ pairing probabilities in different states ($p_s, p_t$); the passengers’ pairing probabilities in different states ($p_s, p_t$) determine the diminishing arrival rates $\lambda_t$ of unpaired passengers along the path of each OD pair, which further affect the probability of passenger existence in each taker-state at any time moment $p$. The detailed formulations (38)-(43) are provided in Appendix B. We have intentionally kept all notations and terminologies identical to those in Wang et al. (2021) so that readers can readily refer to Wang et al. (2021) for more detailed information.

By solving the system of nonlinear equations (38)-(43), we can obtain the value of all the above-mentioned variables under any ridepooling demand rate $\lambda^P$ and pick-up time by vacant vehicle $t_{nk}^p$. And provided the matching probability of unmatched passengers in each state, we can calculate the probability that a passenger between OD pair $w \in W$ fails to get matched before turning into taker-state $t(a,w)$, indicated by $G_{t(a,w)}$, as follows:

$$G_{t(a,w)} = \begin{cases} 1 - p_s(w), k = 0 \\ G_{t(a,w)^{k-1}} \left( 1 - p_t(a_w^{k-1}, w) \right), 1 \leq k \leq |A_w| \end{cases}, w \in W. \tag{4}$$

where $t(a_w^k, w)$ indicates the taker-state associated with the $k$th link along the path of OD pair $w$.

Let $P_w$ be the pairing probability of ridepooling orders between OD pair $w \in W$, that is, the probability for orders between OD pair $w \in W$ getting paired before arriving at the destination. Then from Eq. (4), we immediately have

$$P_w = 1 - G_{t(a_w^{A_w}, w)} \left( 1 - p_t(a_w^{A_w}, w) \right) = 1 - \sum_{k=0}^{A_w} \left( 1 - p_t(a_w^k, w) \right), w \in W. \tag{5}$$

Orders not paired during the trip experience a solo ride time denoted by $t_w^p$, and orders paired with different seekers or takers experience different ride times, depending on their pairing partners. Let $R_{s(w)}^{t(a,\omega)}$ ($R_t^{t(a,\omega)}$) be the ride time of passengers in seeker-state $s(w)$ (taker-state $t(a,w)$) getting paired with passengers in taker-state $t(a,\omega)$ (seeker-state $s(\omega)$). Then the expected ride time of passengers between OD pair $w \in W$ can be calculated by

$$T_w^P = \sum_{t(a,\omega) \in T_{t(w)}} \eta_{t(a,\omega)}^{s(w)} \frac{R_{t(a,\omega)}^{t(a,\omega)}}{\lambda_w^P} + \sum_{a \in A_w} \sum_{s(\omega) \in S_{t(a,w)}} \eta_{t(a,w)}^{s(\omega)} \frac{R_{t(a,\omega)}^{s(\omega)}}{\lambda_w^P} + (1 - P_w) t_w^p, w \in W, \tag{6}$$

where $\eta_{t(a,\omega)}^{s(w)} \eta_{t(a,\omega)}^{s(\omega)}$ ($\eta_{t(a,\omega)}^{s(\omega)}$) is the pairing rate between passengers in seeker-state $s(w)$ (taker-state $t(a,w)$) and taker-state $t(a,\omega)$ (seeker-state $s(\omega)$), and $\eta_{t(a,w)}^{s(\omega)} \eta_{t(a,w)}^{s(\omega)}$ ($\eta_{t(a,w)}^{s(\omega)}$) gives the proportion of ridepooling trips between OD pair $w \in W$ that are paired in seeker-state $s(w)$ (taker-state $t(a,w)$) with takers in taker-state $t(a,\omega)$ (seekers in seeker-state $s(\omega)$).

Let $E_w^P$ be the expected vehicle sharing time of each OD pair $w \in W$, that is, the expected time that a passenger between OD pair $w \in W$ share vehicle space with another passenger. Following the similar method as in Eq. (6), we have

$$E_w^P = \sum_{t(a,\omega) \in T_{t(w)}} \eta_{t(a,\omega)}^{s(w)} \frac{t(a,\omega)}{\lambda_w^P} \eta_{t(a,\omega)}^{s(\omega)} + \sum_{a \in A_w} \sum_{s(\omega) \in S_{t(a,w)}} \eta_{t(a,w)}^{s(\omega)} \frac{t(a,\omega)}{\lambda_w^P} \eta_{t(a,\omega)}^{s(\omega)} + (1 - P_w) t_w^p, w \in W, \tag{7}$$
where \( e_{s(w)}^{t(a, \omega)} \) represents the vehicle sharing time in a ridepooling trip formed by passengers in seeker-state \( s(w) \) and taker-state \( t(a, \omega) \).

### 2.2.2 The mean waiting time of ridepooling orders between each OD pair

Following the notations and terminologies in Wang et al. (2021), we now proceed to investigate the mean waiting time \( t_w \) of ridepooling orders between each OD pair \( w \in W \) under given demand rate \( \lambda^p \), vehicle fleet size \( n \), and expected pairing results \( p_s, p_t, \rho, \eta, \lambda_t \). As mentioned at the beginning of Subsection 2.2, seekers can be assigned to either partially occupied vehicles or vacant vehicles. So the waiting time \( t_w \) is the expectation of the passenger waiting time for pick-up by vacant vehicles \( \bar{t}_w^{pk} \) and the waiting time for pick-up by partially occupied vehicles \( \bar{t}_w^{pk} \).

We first discuss the waiting time for pick-up by vacant vehicles. Seekers who fail to get paired would be assigned to their nearest vacant vehicles. The waiting time \( \bar{t}_w^{pk} \) depends on the number of vacant vehicles nearby. And the distribution of vacant vehicles over the network is determined by the completion rate of ridepooling trips at each node and the vehicle relocation strategies of drivers or the platform. We first examine the completion rate of ridepooling trips at each node. A ridepooling trip is said to be completed only if all on-board passengers are dropped off. So every successfully paired ridepooling trip is completed at the destination of the last passenger disembark. Let \( W(i) \) be the set of OD pairs whose destination node is \( i \in N \), and \( M(i) \) be the set of matching seeker-taker pairs that complete the trip at node \( i \). For vehicles serving passengers between OD pair \( w \in W(i) \), a proportion \( (1 - P_w) \) of them fail to be paired with others, thus complete the service at node \( i \); and for those successfully paired orders, only those matching seeker-taker pairs \( < s(w), t(a, \omega) > \) belonging to \( M(i) \) ends at node \( i \). From the definition of \( \eta \) and \( \rho \) in the previous subsection, suppose \( s(w) \) is a matching seeker-state of \( t(a, \omega) \), then the pairing rate of passengers in seeker-state \( s(w) \) and taker-state \( t(a, \omega) \) can be given by \( \eta_{s(w)}^t(a, \omega)p_t(a, \omega) \). The completion rate of trips at node \( i \) thus can be given by

\[
\nu_i^{c} = \sum_{w \in W(i)} (1 - P_w) \lambda_w^p + \sum_{<s(w), t(a, \omega) \in M(i)} \eta_{s(w)}^t(a, \omega) \rho_t(a, \omega), i \in N. \tag{8}
\]

Upon completing a trip, drivers may move to other areas to seek for the next trip. This relocation movement can be determined individually by drivers according to their experience or centrally determined by the platform. The movement of vacant vehicles between zones would apparently affect the spatial distribution of vacant vehicles, which in turn influences the passenger waiting time for pick-ups by vacant vehicles in each zone. Let \( Z \) be the set of zones, and \( z(i) \in Z \) be the zone to which node \( i \in N \) belongs. Let \( P_{i,z} \) be the proportion of vacant vehicles that relocate to zone \( z \in Z \) after dropping off passengers in node \( i \), \( i \in N \). Then the total relocation flow rate of vacant vehicles to zone \( z \in Z \), denoted by \( v_z \), can be given by

\[
v_z = \sum_{i \in N} P_{i,z} \nu_i^{c}, z \in Z. \tag{9}
\]

There have been many studies that investigate drivers’ individual relocation behavior (e.g., Yang et al., 2002; Xu et al., 2021) and platform’s relocation strategies (e.g. Zhang and Pavone, 2016; Valadkhani and Ramezani, 2023) to determine the value of \( P_{i,z}, i \in N, z \in Z \) under different demand and supply distributions over the network. Since the relocation strategy of vacant vehicles is not the focus of this paper, here we simply assume that drivers cruise within zone \( z(i) \) after completing a trip at node \( i \in N \), i.e., \( P_{i,z(i)} = 1 \), and \( P_{i,z} = 0 \) for \( z \neq z(i) \). This assumption is frequently adopted in the literature for ridesourcing vehicle
dispatching strategy design, (e.g. Yu and Shen, 2019; Shah et al., 2020; Ma and Koutsopoulos, 2022). Let $n^v_z$ indicate the number of vacant vehicles in zone $z \in Z$ at each instant. Without loss of generality, we assume $n^v_z$ is proportional to the arrival rate of vacant vehicles in zone $z \in Z$, i.e.,

$$n^v_z = \frac{v^v_z}{\sum_{z'} v^v_{z'}} n^v_z, z \in Z$$

(10)

where $n^v$ is the total number of vacant vehicles at each instant.

We then proceed to determine the total number of vacant vehicles $n^v$ at each instant. The total number of $n$ vehicles can be divided into three states at each instant: vacant, reserved (i.e., on the way to pick up the first passenger), and occupied (either partially occupied or fully occupied). The total occupied vehicle time can be calculated by the total ride time of passengers between each OD pair minus half of the total vehicle sharing time by all passengers. The total reserved time is the total passenger waiting time for pick-up by vacant vehicles. So the total number of vacant vehicles at each instant satisfies

$$n^v = n - \sum_{w \in W} \left[ (T^p_w - E^p_w) \lambda^p_w + \bar{t}^{pk}_w (1 - P_s(w)) \lambda^p_w \right]$$

(11)

Assuming that the pick-up time by vacant vehicles is a function of the number of vacant vehicles in the zone where the trip starts. Then for orders between OD pair $w \in W$, the pick-up time by vacant vehicles, i.e., $\bar{t}^{pk}_w$, can be given by

$$\bar{t}^{pk}_w = g(n^v_{o(w)})$$

(12)

where $o_w$ denote the origin node of OD pair $w$, and $g(x)$ is assumed to be a continuous and decreasing function, and $g(0) = \bar{g}$.

Now we discuss the mean passenger waiting time for pick-up by partially occupied vehicles, i.e., $\bar{t}^{pk}_w$. A waiting passenger between OD pair $w \in W$ is paired in seeker-state $s(w)$ if it is assigned to a partially occupied vehicle, and the on-board passenger of the partially occupied vehicle must be in one of the pairing taker-states $t(a, w') \in T_s(w)$. Every seeker-taker pair $< s(w), t(a, w') >$ is associated with a certain pick-up distance, so for passengers in seeker state $s(w)$, the mean pick-up time by partially occupied vehicles is dependent on their pairing rates with different pairing taker-states. Unlike the pick-up time by vacant vehicles that decreases with the number of the vacant vehicles nearby, $\bar{t}^{pk}_w$ does not necessarily increase or decrease with the number of matching partially occupied vehicles nearby at each instant. And since the pick-up time of every matching seeker-taker pair is within $[0, \bar{t}_r]$, here we simply assume that $\bar{t}^{pk}_w = \frac{\bar{t}_r}{2}$.

Provided that a passenger between OD pair $w \in W$ may be paired and pick-up by partially occupied vehicles with probability $P_s(w)$, the mean passenger waiting time of OD pair $w$ then can be given by

$$t_w = P_s(w) \frac{\bar{t}_r}{2} + (1 - P_s(w)) \bar{t}^{pk}_w, w \in W$$

(13)

### 2.3 Equilibrium and Solution Algorithm

In the previous subsections, Eqs. (38)-(43) in Appendix B describes how the ridepooling demand rate of each OD pair $\lambda$ and the waiting time for pick-up by vacant vehicles $\bar{t}^{pk}$ influence the passenger pairing probability in each state $(P_s, P_t)$, the probability of passenger existence in each taker-state $\rho$, the arrival rate
of pairing opportunities for each taker-state $\eta$, the arrival rate of unpaired passengers in each taker-state $\lambda_t$. These factors further determine the pairing probability $P_w$, ride time $T^p_w$, and vehicle sharing time $E^p_w$ of each OD pair $w \in W$ as described in Eqs. (4)-(7). Let $T^P = (T^p_w, w \in W)$. From Eqs. (8)-(11), the demand rate of each OD pair $\lambda^P$, the pairing probability $P_w$ of each OD pair $w \in W$, the arrival rate of pairing opportunities $\eta$ and the probability of passenger existence $\rho$ at each instant determines the completion rate of ridepooling trips at each node, which further determine the spatial distribution of vacant vehicles $n^v = (n^v_z, z \in Z)$ over the network, and governs the value of $t^p$ and $t^P = (t_w, w \in W)$ as in Eqs. (12)-(13). And the passenger waiting time $t_w$ and ride time $T^p_w$ of each OD pair $w \in W$, together with the platform’s discounting strategy $\theta = (\theta_w, w \in W)$, governs the ridepooling demand $\lambda^P$, as given in Eqs. (1) and (2). The complicated relationships between these variables above are illustrated by Figure 2. The system reaches an equilibrium state when all these relationships (1)-(13) and (38)-(43) hold simultaneously: provided the expected service quality $(t_w, T^p)$, there are $\lambda^P$ passengers who choose ridepooling services; and with $\lambda^P$ passengers choosing ridepooling services, the service quality is $(t_w, T^p)$.

Figure 2: The relationships between the vector of arguments

Let $X = (\lambda^P, t^P, T^p, \bar{t}^p, n^v, p_s, p_t, \eta, \rho, \lambda_t)$ be the vector of arguments, and define

$$F(X, \theta) = (\lambda^P(X, \theta), t^P(X), T^p(X), \bar{t}^p(X), n^v(X), p_s(X), p_t(X), \eta(X), \rho(X), \lambda_t(X))$$

as a vector of functions, where $\lambda^P(X), t^P(X), T^p(X), \bar{t}^p(X), n^v(X), p_s(X), p_t(X), \eta(X), \rho(X), \lambda_t(X)$ are respectively defined by

$$\lambda^P(X, \theta) = \{\lambda^p_w(X, \theta) = \bar{\lambda}_w f_w(C^P_w(X, \theta) - C^v_w), w \in W\}$$ (14)

$$t^P(X) = \left\{ t_w(X) = p_s(w) \frac{\bar{t}_r}{2} + (1 - p_s(w)) \bar{t}^p_w, w \in W \right\}$$ (15)
\( T^p(X) = \left\{ \begin{array}{l} T^p_w(X) = \sum_{t(a,\omega) \in T_{s(w)}} \eta_{t(a,\omega)} \rho_{t(a,\omega)} \frac{R^s_{t(a,\omega)}}{\lambda^p_w} \\
 + \sum_{a \in A_w} \sum_{s(\omega) \in S_{t(a,\omega)}} \eta_{s(\omega)} \rho_{t(a,\omega)} \frac{R^s_{t(a,\omega)}}{\lambda^p_w} (1 - P_w(X)) \tau_w s(w) \end{array} \right\} \) \quad (16)

\( \bar{t}^{pk} = \left\{ \bar{t}^{pk}_w(X) = g(n^v_{z(\omega)}), w \in W \right\} \) \quad (17)

\( n^v(X) = \left\{ n^v_{z}(X) = \frac{v_z(X)}{\sum_{z \in Z} v'_z(X)} n^v(X), z \in Z \right\} \) \quad (18)

\( p_s(X) = \left\{ \begin{array}{l} p_{s(w)}(X) = \left\{ \begin{array}{l} 1 - \prod_{t(a,\omega) \in T_{s(w)}} (1 - \rho_{t(a,\omega)}), \text{if } T_{s(w)} \neq \emptyset \\
0, \text{otherwise} \end{array} \right\}, s(w) \in S \right\} \) \quad (19)

\( p_t(X) = \left\{ \begin{array}{l} p_{t(a,\omega)}(X) = \left\{ \begin{array}{l} 1 - \exp (-\eta_{t(a,\omega)}(X) \tau_{t(a,\omega)}(X)), \text{if } \eta_{t(a,\omega)} > 0 \\
0, \text{if } \eta_{t(a,\omega)} = 0 \end{array} \right\}, t(a,\omega) \in T \right\} \) \quad (20)

\( \eta(X) = \left\{ \begin{array}{l} \eta_{t(a,\omega)}(X) = \left\{ \begin{array}{l} \lambda_{\omega}, \text{if } s(\omega) \in S_{t(a,\omega)} \text{ and } T^s_{t(a,\omega)} = \emptyset \\
\lambda_{\omega} \prod_{t'(a',\omega) \in T^s_{t(a,\omega)}} (1 - \rho_{t'(a',\omega)}), \text{if } s(\omega) \in S_{t(a,\omega)} \text{ and } T^s_{t(a,\omega)} \neq \emptyset \end{array} \right\} \right\} \) \quad (21)

\( \rho(X) = \left\{ \begin{array}{l} \rho_{t(a,\omega)}(X) = \left\{ \begin{array}{l} \lambda_{t(a,\omega)} [1 - \exp (-\eta_{t(a,\omega)}(X) \tau_{t(a,\omega)}(X))], \text{if } \eta_{t(a,\omega)} > 0 \\
\lambda_{t(a,\omega)} \tau_{t(a,\omega)}(X), \text{if } \eta_{t(a,\omega)} = 0 \end{array} \right\}, t(a,\omega) \in T \right\} \) \quad (22)

\( \lambda_t(X) = \left\{ \begin{array}{l} \lambda_{t(a,k,w)}(X) = \left\{ \begin{array}{l} \lambda^p_{t(a,k,w)} (1 - \rho_{s(w)}), k = 0 \\
\lambda_{t(a,k,w)} \left( 1 - \rho_{t(a,k-1,w)} \right), 1 \leq k \leq |A_w| \end{array} \right\}, t(a,k,w) \in T \right\} \) \quad (23)

where \( C^p_w(X, \theta) \) in Eq. (14) is defined by the right-hand side of Eq. (1), \( P_w(X) \) in Eq. (16) is defined by the right-hand side of Eq. (5), \( \eta_{t(a,\omega)}(X) \) in Eq. (20) is defined by the right-hand side of Eq. (40), \( v_z(X) \) and \( n^v(X) \) in Eq. (18) are respectively defined by the right-hand side of Eqs. (9) and (11), and \( \tau_{t(a,\omega)}(X) \) in Eqs. (20) and (22) are defined by the right-hand side of Eq. (3). Let \( \Omega \) be the feasible set of \( X \) given by

\( \Omega = \left\{ \lambda^p, t^p, t^p, i^{pk}, n^v, p_s, p_t, \eta, \rho, \lambda_t \right\} \) \quad (24)
Then Eqs. (1)-(13) and (38)-(43) can be considered as a fixed point problem: Find \( X \in \Omega \) such that \( X = F(X, \theta) \). Based on Brouwer’s fixed point theorem, it is not difficult to establish the existence of solutions to this fixed point problem.

**Proposition 1.** Suppose \( \bar{\lambda}_w \max \{g, \max \{t_a, a \in A\}\} \leq 1 \) and \( n \geq \sum_{w \in W} (t_w^* + \bar{i}_d) \bar{\lambda}_w \), then the existence of solutions to the system of nonlinear equations (1)-(13) and (38)-(43) on the feasible region \( \Omega \) given by Eq. (24) is guaranteed under any feasible discounting strategy \( \theta = (\theta_w, w \in W) \in [0, 1]^{|W|} \).

**Proof.** Please refer to Appendix C.

To solve the system of nonlinear equations (1)-(13) and (38)-(43) to obtain the expected system performance at equilibrium under any discounting strategy \( \theta \), we develop the following solution procedure. The algorithm is heuristic without a theoretical guarantee of convergence, but in our numerical experiments in Section 5, we have consistently achieved convergence in both grid-network and Haikou network experiments. It is also worthwhile to note that due to the complexity of the problem, the equilibrium solutions of the system of nonlinear equations (1)-(13) and (38)-(43) are not necessarily unique. So following the above solution algorithm, different equilibria may reached from different initial points. In our numerical experiments in Section 5, we simply take one equilibrium solution generated from a randomly generated initial point.

**Algorithm 1** Solution Procedure for solving the fixed point problem under any given \( \theta \)

**Require:** Given initial \( \bar{i}^{pk}, \lambda^p, \theta, \) and \( \epsilon \). Let \( \bar{i}^{pkt} = 0, \lambda^{pl} = 0 \).

**Ensure:** \( \lambda^p, t^p, T^p, \bar{i}^{pk}, n^v, p_s, p_t, \eta, \rho, \lambda_t \)

1: while \( \bar{i}^{pk} - \bar{i}^{pkt} > \epsilon \) or \( \lambda^p - \lambda^{pl} > \epsilon \) do
2: Calculate \( \{p_s, p_t, \eta, \rho, \lambda_t\} \) by solving Eqs.(38)-(43) with fixed \( \bar{i}^{pk} \) and \( \lambda^p \).
3: Calculate \( n^v \) with \( \{p_s, p_t, \eta, \rho, \lambda_t\} \) by solving Eqs.(8)-(11).
4: Update \( \bar{i}^{pkt} \) by solving Eq.(12) with fixed \( n^v \).
5: Calculate \( t^p \) by solving Eq.(13) with fixed \( \{\bar{i}^{pkt}, p_s\} \).
6: Calculate \( T^p \) by solving Eqs.(4)-(7) with fixed \( \{p_s, p_t, \eta, \rho, \lambda_t\} \).
7: Update \( \lambda^{pl} \) by solving Eqs.(1) and (2) with fixed \( \{T^p, t^p, \theta\} \).
8: end while

Let \( \lambda^p(\theta) \) be the equilibrium demand rate, and \( T^p_w(\theta) \) and \( E^p_w(\theta) \) respectively be the expected ride time and vehicle sharing time of OD pair \( w \in W \), under discounting strategy \( \theta \). Let \( c' \) be the platform’s payment to drivers for each occupied unit of time. Then, given a discounting strategy \( \theta \), the profit of the ridepooling platform, denoted by \( \pi(\theta) \), can be given by

\[
\pi(\theta) = \sum_{w \in W} \left[ \theta_w p_w^s - c' \left( T^p_w(\theta) - \frac{E^p_w(\theta)}{2} \right) \right] \lambda^p_w(\theta),
\]

(25)

The first term in the bracket represents the expected revenue of each order between OD pair \( w \in W \), and the second term is the platform’s expected payment to the driver for each order between OD pair \( w \in W \).

### 3 Ridepooling pricing strategy design

As we can see from Eq. (25), the ridepooling demand rate, the expected ride time and the expected vehicle sharing time of each OD pair under a discounting strategy determines the profit of the ridepooling platform.
An accurate prediction of $\lambda_w(\theta)$, $T_w^p(\theta)$ and $E_w^p(\theta)$ of each OD pair $w \in W$ under any given $\theta$ plays a key role in the design of discounting strategies for the platform.

In this section, we propose two methods for the design of discounting strategies for the ridepooling platform. In the first method in Subsection 3.1, the platform offers passengers the estimated ride time based on historical data, and optimizes the discount ratio for each OD pair independently based on the weighted average waiting time, ride time and vehicle sharing time of each OD pair estimated from historical data. As the realized ride time and vehicle sharing time would generally deviate from the platform’s estimation based on historical data, it requires a day-to-day adjustment process. This method is applicable when the platform lacks the ability to predict the expected ride time on historical data, it requires a day-to-day adjustment process. This method is applicable when the platform adopts a day-to-day price adjustment: start from an initial presumed waiting, ride and vehicle sharing time of each OD pair, denoted by $\bar{t}_w^p, \bar{T}_w^p, \bar{E}_w^p$. In the first method in Subsection 3.1, the platform offers passengers the estimated ride time based before implementation. In the second method introduced in Subsection 3.2, the discount ratio of orders between all OD pairs over the network are optimized coordinatively, considering the complex interactions among the expected pairing results and ridepooling demand between different OD pairs as described in Section 2.

### 3.1 Method 1: Independent optimization and day-to-day adjustment based on historical information

Let $\bar{t}_w^p$, $\bar{T}_w^p$ and $\bar{E}_w^p$ respectively be the average waiting time, ride time and vehicle sharing time of OD pair $w \in W$ estimated from historical data. In the first method, the platform simply set $\bar{t}_w^p(\lambda^p) = \bar{t}_w^p, w \in W$ in Eq. (13), set $T_w^p(\lambda^p) = \bar{T}_w^p, w \in W$, and $E_w^p(\lambda^p) = \bar{E}_w^p, w \in W$ in Eq. (25), and solve the following optimization problem to achieve the optimal discount ratio:

\[
\text{Method 1: } \max_{\theta} \pi(\theta) = \sum_{w \in W} \left[ \theta_w p_w^x - c^l \left( \bar{T}_w^p - \frac{\bar{E}_w^p}{2} \right) \right] \lambda_w^p \tag{26}
\]

s.t. \[
\lambda_w^p = \bar{\lambda}_w f_w (C_w^p - \bar{C}_w) \tag{27}
\]

\[
C_w^p = \theta_w p_w^r + \beta_w (\bar{E}_w^p + \bar{T}_w^p) + \Delta \tag{28}
\]

\[
\theta_w \in [0, 1], \ w \in W \tag{29}
\]

In the above optimization problem, $\bar{t}_w^p$, $\bar{T}_w^p$, and $\bar{E}_w^p, w \in W$ are all constants, and the objective function and constraints are separable for each OD pair $w \in W$. So the above optimization problem can be solved separately for each OD pair, and this method can be understood as optimizing the discounting strategy for each OD pair separately. Let $\bar{\theta}(\bar{i}^p, \bar{T}^p, \bar{E}^p) := (\bar{\theta}_w(\bar{t}_w^p, \bar{T}_w^p, \bar{E}_w^p), w \in W)$ be the optimal solution to the above optimization problem when the expected waiting time, ride time and vehicle sharing time of each OD pair are presumed to be $\bar{t}_w^p = (\bar{t}_w^p, w \in W), \bar{T}_w^p = (\bar{T}_w^p, w \in W)$ and $\bar{E}_w^p = (\bar{E}_w^p, w \in W)$, and $\theta^p(\bar{\theta}(\bar{i}^p, \bar{T}^p, \bar{E}^p))$ be the ridepooling demand rate under the discounting strategy $\bar{\theta}(\bar{i}^p, \bar{T}^p, \bar{E}^p)$.

Since the historical average waiting time, ride time and vehicle sharing time are determined by the previous ridepooling demand pattern, after implementing the discounting strategy $\bar{\theta}(\bar{i}^p, \bar{T}^p, \bar{E}^p)$, the ridepooling demand rate of each OD pair would differ from the past one, and the observed waiting, ride time and vehicle sharing time of each OD pair $w$ would deviate from the presumed values $\bar{t}_w^p, \bar{T}_w^p$, and $\bar{E}_w^p$. Therefore, the platform adopts a day-to-day price adjustment: start from an initial presumed waiting, ride and vehicle sharing time of each OD pair, i.e., $\bar{i}^p, \bar{T}^p$ and $\bar{E}^p$, solve Eq. (26)-(29) to obtain the optimal pricing strategy $\bar{\theta}(\bar{i}^p, \bar{T}^p, \bar{E}^p)$ of Method 1; implement the pricing strategy $\bar{\theta}(\bar{i}^p, \bar{T}^p, \bar{E}^p)$, and observe the new waiting, ride and vehicle sharing time of each OD pair, denoted by $\bar{i}^{p'}, \bar{T}^{p'}$ and $\bar{E}^{p'}$; update the estimated waiting, ride
and vehicle sharing time by $\tilde{t}^p \leftarrow \alpha \tilde{t}^p + (1 - \alpha) \tilde{t}^p'$, $\tilde{T}^p \leftarrow \alpha \tilde{T}^p + (1 - \alpha) \tilde{T}^p'$ and $\tilde{E}^p \leftarrow \alpha \tilde{E}^p + (1 - \alpha) \tilde{E}^p'$, and then implement the new pricing strategy derived by solving Eq. (26)-(29) based on the new $\tilde{t}^p$, $\tilde{T}^p$ and $\tilde{E}^p$. The price adjustment process is shown in Fig. 3.

The platform obtains the initial presumed waiting, ride and vehicle sharing time $\tilde{t}^p, \tilde{T}^p, \tilde{E}^p$ based on historical data.

The platform solves Eqs. (26)-(29) in Method 1 to obtain the optimal pricing strategy $\hat{\theta}(\tilde{t}^p, \tilde{T}^p, \tilde{E}^p)$.

The platform implements the price strategy $\hat{\theta}(\tilde{t}^p, \tilde{T}^p, \tilde{E}^p)$ and observes the new waiting, ride and vehicle sharing time $\tilde{t}^p', \tilde{T}^p', \tilde{E}^p'$.

The platform updates the estimated waiting, ride and vehicle sharing time by $\hat{\tilde{t}}^p \leftarrow \alpha \tilde{t}^p + (1 - \alpha) \tilde{t}^p'$, $\hat{\tilde{T}}^p \leftarrow \alpha \tilde{T}^p + (1 - \alpha) \tilde{T}^p'$ and $\hat{\tilde{E}}^p \leftarrow \alpha \tilde{E}^p + (1 - \alpha) \tilde{E}^p'$.

Day-to-day price adjustment iterations

Figure 3: Day-to-day price adjustment process in Method 1.

3.2 Method 2: Coordinated optimization considering the systematic impacts of discounting strategies

In the second method, the platform considers the complex impacts of $\lambda^p$ on $T_w^p(\lambda^p)$ and $E_w^p(\lambda^p)$, and solve the following optimization problem to maximize the platform’s profit by Sophistically managing the ridepooling demand pattern over the network. Then, the discounting strategy design problem becomes:

$$\text{Model } - 2: \max_{\theta} \tilde{\pi}(\theta) = \sum_{w \in W} \left[ \theta_w p_w^s - c' \left( T_w^p(\theta) - \frac{E_w^p(\theta)}{2} \right) \right] \lambda_w^p(\theta)$$

s.t. $\theta_w \in [0, 1], w \in W.$ (30)

where $\lambda_w^p(\theta), T_w^p(\theta)$ and $E_w^p(\theta)$ are subject to Eqs. (1)-(13) and (38)-(43).

In contrast with Method 1, which independently optimizes discounting strategies for each OD pair, Method 2 optimizes the discounting strategies for all OD pairs simultaneously. So the optimal discounting strategy of each OD pair generated by Method 2 will coordinate with those of other OD pairs to maximize the total platform profit. Since for any given $\theta$, the values of $\lambda_w^p(\theta), w \in W, T_w^p(\theta), w \in W,$ and $E_w^p(\theta), w \in W$, can only be obtained by solving a complex system of nonlinear equations, Model 2 is a highly nonlinear and non-convex optimization problem whose gradient information cannot be easily obtained. To solve this problem, we explore two derivative-free algorithms in the next section.
4 Solution methods

In this section, for the nonlinear and non-convex optimization problem (30) in Method 2, we propose two derivative-free methods: (1) Bayesian Optimization; and (2) classification-based optimization. Derivative-free optimization (Kolda et al., 2003; Conn et al., 2009), also known as zeroth-order or black-box optimization, refers to a class of optimization algorithms that do not rely on gradient information. Instead, they operate solely based on the objective function values obtained from sampled solutions. By relaxing the need for derivative information, derivative-free algorithms offer simplicity and effectiveness in addressing complex optimization problems characterized by non-convex and non-differentiable objective functions. Moreover, these algorithms commonly employ a sampling-and-updating framework, iteratively improving the quality of solutions and achieving global optimization with a certain probability (Qian and Yu, 2021).

4.1 Bayesian optimization

Bayesian optimization (BO) is a sequential design strategy for global optimization of black-box models based on function-value information (Mockus et al., 1978). It can effectively handle complex functions where obtaining explicit expressions or gradients is difficult or impractical. By sequentially selecting points to evaluate the objective function based on previous observations, BO intelligently explores the search space while balancing the exploration and exploitation. The typical BO algorithm consists of two main steps. First, a surrogate model is learned to approximate the objective function using a Bayesian posterior probability distribution. Second, an acquisition function is maximized to determine the next sampling point based on the current posterior distribution over the objective function. Here, we provide a brief overview of these steps, while comprehensive reviews and additional details can be found in Brochu et al. (2010), Shahriari et al. (2015), and Frazier (2018).

(1) Surrogate model

In this paper, we adopt the Gaussian Process (GP) as the surrogate model to approximate the objective function, which is the most commonly used model in BO (Mockus, 1994). A GP defines a prior distribution over functions, assuming that any finite set of function values follows a multivariate Gaussian distribution. It is completely specified by its mean function \( m(\mathbf{x}) \) and covariance function \( k(\mathbf{x},\mathbf{x}') \):

\[
f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x},\mathbf{x}')).
\]

(32)

For convenience, it is common practice to employ a zero function as the mean function (e.g., Brochu et al., 2010; Rasmussen and Williams, 2006). Therefore, we define the mean function as a zero function, i.e., \( m(\mathbf{x}) = 0 \). As for the covariance function \( k(\mathbf{x},\mathbf{x}') \), we adopt the widely used Matérn kernel (Matérn, 2013) for our GP modeling:

\[
k(\mathbf{x},\mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)}(\sqrt{2\nu} \| \mathbf{x} - \mathbf{x}' \|)^\nu K_\nu(\sqrt{2\nu} \| \mathbf{x} - \mathbf{x}' \|),
\]

(33)

where \( \Gamma(\cdot) \) is the Gamma function and \( K_\nu \) is the modified Bessel function, and \( \nu \) is a positive parameter (Rasmussen and Williams, 2006).

Let \( \mathbf{x}_{1:n} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \) and \( y_{1:n} = [y_1, \ldots, y_n]^T \), where \( \mathbf{x}_n \) is the \( n^{th} \) sample point, and \( y_n \) is the corresponding objective function value. Given a set of observations \( D_{1:n} = (\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_n,y_n) \), GP obtains the predictive distribution over any point \( \mathbf{x} \) in the search domain:

\[
f(\mathbf{x})|D_{1:n} \sim \mathcal{N}(\mu_n(\mathbf{x}), \sigma_n^2(\mathbf{x})),
\]

(34)
where the mean and covariance are given as: \( \mu_\theta(x) = k^T K^{-1} y_{1:n}, \sigma_\theta^2(x) = k(x, x) - k^T K^{-1} k. \) Here \( k \) is a vector that \( k = [k(x_1, x_1), \ldots, k(x_n, x_n)]^T \) and \( K \) is covariance matrix such that \( K_{i,j} = k(x_i, x_j). \)

(2) Acquisition function

The acquisition function plays a crucial role in selecting the next point to evaluate. It considers both the mean \( \mu_\theta(x) \) and covariance \( \sigma_\theta^2(x) \) provided by the surrogate model to balance the exploration (sampling where the objective mean \( \mu_\theta(x) \) is high) and exploitation (sampling where the uncertainty \( \sigma_\theta^2(x) \) is high), and is cheap to compute. The goal is to maximize the utility of the next sampling. Typical choice of acquisition functions includes the probability of improvement (PI) (Kushner, 1964), the expected improvement (EI) (Mockus, 1998), and the GP upper confidence bound (GP-UCB) (Srinivas et al., 2009). Similar to many other studies, we adopt GP-UCB as the acquisition function.

\[
\alpha_{GP-UCB}(x) = \mu_\theta(x) + \phi_n \sigma_\theta(x) \tag{35}
\]

where \( \phi_n \) is used to balance between exploration and exploitation and increases with iteration.

Based on the provided description, Algorithm 2 presents the workflow for solving Method 2 using BO. The algorithm begins by placing a Gaussian process prior to the profit function \( \tilde{\pi} \), denoted by \( \tilde{\pi} \sim \mathcal{GP}(m(\theta), k(\theta, \theta')) \) (line 1). A vector of feasible price discount ratios \( \theta = (\theta_w, w \in W) \) within the solution space \( \Omega_\theta = \{ \theta | \theta_w \in [0, 1], w \in W \} \) is referred to as a solution. The algorithm randomly samples \( n_0 \) solutions from the solution space \( \Omega_\theta \) (line 2). Subsequently, it calculates the platform’s profit \( \tilde{\pi}(\theta) \) using Eq. (30) for each solution and records the initial dataset \( D_{1:n} = \{ (\theta_1, \tilde{\pi}_1), \ldots, (\theta_n, \tilde{\pi}_n) \} \) consisting of solution-value tuples \( (\theta, \tilde{\pi}(\theta)) \) (line 3). In each iteration \( n^4 \), BO employs a Gaussian process (GP) as a surrogate model and updates the posterior distribution \( f(\theta)|D_{1:n} \sim \mathcal{N}((\mu_n(\theta), \sigma_n^2(\theta)) \) based on the existing dataset \( D_{1:n} \) (line 5). Subsequently, it determines the next sampling solution \( \theta_{n+1} \) by optimizing the acquisition function \( \alpha_{GP-UCB}(\theta) \) (line 6). The algorithm calculates the profit \( \tilde{\pi}_{n+1} \) and updates the dataset \( D_{1:n+1} \) (line 7). It then updates the best solution \( \theta^* \) (line 8). Finally, the optimal discount ratios are returned.

**Algorithm 2** Optimize Method 2 with BO

**Require:**
- Function of platform’s ridepooling profit: \( \tilde{\pi}(\theta) \);
- Solution space: \( \Omega_\theta = \{ \theta | \theta_w \in [0, 1], w \in W \} \);
- Number of initial sampled solutions: \( n_0 \);
- Maximal number of iterations: \( N \);
- GP mean and covariance function: \( m(\theta) = 0, k(\theta, \theta') \) as in eq.(33)
- Acquisition function: \( \alpha_{GP-UCB}(x) = \mu_\theta(x) + \kappa_n \sigma_\theta(x) \)

**Ensure:**
- The optimal discount ratio: \( \theta^* \).
1: Place a Gaussian process prior on \( \tilde{\pi} \): \( \tilde{\pi} \sim \mathcal{GP}(m(\theta), k(\theta, \theta')) \)
2: Randomly sample \( n_0 \) solutions from \( \Omega_\theta \) and calculate \( \tilde{\pi}_i \) of each solution \( i \in [1, n_0] \).
3: Set \( n = n_0, D_{1:n} = \{ (\theta_1, \tilde{\pi}_1), \ldots, (\theta_n, \tilde{\pi}_n) \} \)
4: while \( n < N \) do
5: Update the posterior probability distribution on \( \tilde{\pi} \) with \( D_{1:n}; f(\theta)|D_{1:n} \sim \mathcal{N}((\mu_n(\theta), \sigma_n^2(\theta)) \)
6: Find the next solution to maximize the acquisition function:
\[ \theta_{n+1} = \arg \max_{\theta \in \Omega_\theta} \alpha_{GP-UCB}(\theta) \]
7: \( \tilde{\pi}_{n+1} = \tilde{\pi}(\theta_{n+1}), D_{1:n+1} = D_n \cup (\theta_{n+1}, \tilde{\pi}_{n+1}) \)
8: Update the best solution: \( \theta^* \)
9: \( n = n+1 \)
10: end while

\(^4\)In this section, \( n \) and \( N \) are a little abused to denote the number of iterations and the maximum number of iterations.
4.2 Classification-based optimization

Classification-based optimization (CBO) is an emerging theoretical framework in the field of model-based derivative-free optimization methods (Lozano, 2006; Hashimoto et al., 2018; Qian and Yu, 2021). In the context of classification-based optimization, a classification model is trained to categorize solutions into two distinct categories: good or bad. This trained model effectively divides the solution space into regions corresponding to these categories. Then the next sampling point has a higher likelihood of being chosen from the identified good region. In this study, we employ the Sequential randomized coordinate shrinking classification algorithm (SRACOS) (Yu et al., 2016; Hu et al., 2017), which is an implementation of classification-based optimization that has demonstrated empirical effectiveness on various tasks. The distributed version of SRACOS, known as ZOOpt (Liu et al., 2017), is an open-source toolkit readily accessible for use.

Now we introduce the process of solving for the optimal platform’s ridepooling profit and the optimal price ratio for each OD using the SRACOS algorithm, as depicted in Algorithm 3. In this algorithm, a vector \( \theta = (\theta_w, w \in W) \) of feasible price discount ratios on the solution space \( \Omega_\theta \) is referred to as a solution, where \( \Omega_\theta = \{ \theta | \theta_w \in [0,1], w \in W \} \). To begin, the algorithm uniformly samples a solution set \( S \) of size \( n_0 \) from the solution space \( \Omega_\theta \) and calculates the platform’s profit \( \tilde{\pi}(\theta) \) using Eq. (30) for each solution in \( S \) (line 1). Each solution, along with its corresponding profit, forms a solution-value tuple \( (\theta, \tilde{\pi}(\theta)) \), and all these tuples are stored in a tuple set denoted by \( B \) (line 2). Next, the algorithm partitions the solution-value tuples in \( B \) into a positive set \( B^+ \), which contains the \( k \) best tuples, and a negative set \( B^- \), which contains the remaining tuples (line 3). It keeps track of the current best solution \( \theta^* \) (line 4). During the \( n^{th} \) iteration, the algorithm employs a binary classification algorithm \( C \) to learn a hypothesis \( h \) (line 6). The hypothesis \( h \) represents an axis-parallel box that distinguishes a randomly chosen positive solution from all the negative solutions. The details of the learning algorithm can be found in (Yu et al., 2016). Let \( D_h \) denote the region of positive solutions defined by hypothesis \( h \). The algorithm samples a new solution \( \theta_{n+1} \) from the uniform distribution over \( D_h \) with a probability of \( \gamma \), or from the solution space \( \Omega_\theta \) with a probability of \( 1 - \gamma \) (line 7). Once the profit \( \tilde{\pi}_{n+1} \) is calculated and a new solution-value tuple is obtained (line 8), the algorithm updates the sets \( B^+ \) and \( B^- \) using specific strategies to ensure that \( B^+ \) consistently retains the best solutions found so far (line 9). Then it updates the best solution \( \theta^* \) (line 10). In the end, the best solution \( \theta^* \) is returned as the output.

5 Numerical experiments

In this section, we conduct extensive numerical experiments to assess the computational performance of the two solution algorithms for Model 2 and compare the platform’s profits and user experience under the up-front discount pricing strategies obtained by the two different methods in Section 3 through simulation.

5.1 Experiments on a 15*15 grid network

5.1.1 Experiments settings

We first conducted numerous experiments based on a 15*15 small grid network with 30 randomly generated OD pairs. Figure 4 illustrates one such distribution pattern of these pairs. The red and blue dots, respectively, represent
**Algorithm 3** Optimize Method 2 with SRACOS

**Require:**
- Function of platform’s ridepooling profit: $\bar{\pi}(\theta)$;
- Solution space: $\Omega_\theta = \{ \theta_\theta \in [0, 1], w \in W \}$;
- Number of initial sampled solutions: $n_0$;
- A binary classification algorithm: $C$;

**Ensure:**
- The best solution: $\theta^*$

1: Uniformly sample a set of solutions consisting $n_0$ solutions from $\Omega_\theta$ and calculate $\bar{\pi}_i$ of each solution $i \in [1, n_0]$
2: Set $n = n_0$, record the tuple of solution values $B = \{(\theta_1, \bar{\pi}_1), \ldots, (\theta_n, \bar{\pi}_n)\}$
3: Partition $B$ into a positive set containing the $k$ best tuples and a negative set consisting of the rest of tuples: $(B^+, B^-)$
4: Record the best solution: $\theta^*$
5: while $n < N$ do
6: Learn a hypothesis $h$ from a binary classification algorithm: $h = C(B^+, B^-)$
7: Sample the next solution $\theta_{n+1}$ from the uniform distribution over $D_h$, the positive region represented by $h$, with probability $\gamma$ or over solution space $\Omega_\theta$ with $1 - \gamma$
8: $\bar{\pi}_{n+1} = \bar{\pi}(\theta_{n+1})$, $B = B \cup (\theta_{n+1}, \bar{\pi}_{n+1})$
9: Update the positive $B^+$ and negative set $B^-$
10: Update the best solution: $\theta^*$
11: $n = n + 1$
12: end while

indicate the origin and destination of each OD pair, and the blue lines connecting each OD pair are the service route for ridepooling passengers before getting matched. The maximal ridepooling demand rate of each OD pair $w \in W$ is assumed to be 0.1 orders/min. A scalar $\kappa$ is introduced to simultaneously change the demand rates of all OD pairs.

![Figure 4: The 15*15 grid network](image)

The function $f_w(x)$ in Eq.(2) is assumed to following the logistic function

$$f_w(x) = \frac{1}{1 + e^{\varsigma(x)}}, w \in W,$$  \hspace{1cm} (36)

where $\varsigma$ is a constant that calibrates the sensitivity of the demand to the relative cost difference and is set to
be 1.8 in the grid-network example. The pick-up time function $g(n^z_w)$ in Eq.(12) is assumed to follow

$$g(n^z_w) = \frac{\zeta}{2\theta \sqrt{n^z_w/\Lambda}}, z \in Z,$$

(37)

where $\theta$ is vehicle speed, $\Lambda$ is zone area, $\zeta$ is the detour ratio that is estimated to be 2.29 for the grid-network example by data fitting based on the simulation experiments. The parameter $\Lambda$ denotes the searching area for vacant vehicles and is set to be $10^2$ square grids. The rest parameter settings are summarized in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default value</th>
<th>grid network</th>
<th>Haikou network</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_w$</td>
<td>The maximal ridepooling demand rate of each OD pair $w, w \in W$</td>
<td>0.1 orders/min</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta$</td>
<td>The inconvenience cost of ridepooling passengers</td>
<td>0.5 CNY</td>
<td>1 CNY</td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td>Passenger’s average value of time (VOT)</td>
<td>2 CNY/min</td>
<td>2 CNY/min</td>
<td></td>
</tr>
<tr>
<td>$c$</td>
<td>A unit time-based rate for pre-discount price $p^*_w$ of each order between OD pair $w$</td>
<td>3 CNY/min</td>
<td>1.25 CNY/min</td>
<td></td>
</tr>
<tr>
<td>$c'$</td>
<td>The platform’s payment to the driver for each occupied unit time</td>
<td>2 CNY/min</td>
<td>1 CNY/min</td>
<td></td>
</tr>
<tr>
<td>$\bar{t}_p$</td>
<td>The maximally allowed pick-up time</td>
<td>5 min</td>
<td>6 min</td>
<td></td>
</tr>
<tr>
<td>$\bar{t}_d$</td>
<td>The maximally allowed detour time</td>
<td>5 min</td>
<td>6 min</td>
<td></td>
</tr>
<tr>
<td>$\bar{v}$</td>
<td>The vehicle speed</td>
<td>1 grid/min</td>
<td>30km/hr</td>
<td></td>
</tr>
<tr>
<td>$\zeta$</td>
<td>A constant in specified ridepooling demand function $f_w(\cdot)$</td>
<td>1.8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$\zeta$</td>
<td>The detour ratio in specified pick-up time function $p^w_{pl}$</td>
<td>2.29</td>
<td>2.5</td>
<td></td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>Zone area</td>
<td>100 square grids</td>
<td>3 $km^2$</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Default value and ranges of the main parameters used in numerical experiments

In our experiment, we compare the optimal pricing strategies derived by Method 1 and Method 2 with different uniform discounting strategies. This comparison was carried out across 100 distinct scenarios, which incorporated varying combinations of the demand scaler ($\kappa$) selected from the set $\{1, 2.5, 5, 7.5, 10\}$, a range of vehicle fleet size ($n$) from $\{175, 200, 250, 300, 400\}$, and 4 randomly generated distribution patterns of the 30 OD pairs. For each scenario, we first utilize Method 1 and Method 2, as described in Section 3, to derive the optimal pricing strategies aimed at maximizing platform profit. Subsequently, these pricing strategies are implemented in simulations to assess their impact on system performance.

Specifically, for Method 1, we start from an estimated waiting time, ride time, and vehicle sharing time for each OD pair derived from warm-up simulations, and then put them into Eqs. (26)-(29) to generate the initial pricing strategy. The optimization problem (26)-(29) is solved with a common solver from Scipy, utilizing the L-BFGS-B method. The resulting pricing strategy is then implemented in the simulation environment. As outlined in Subsection 3.1, the observed the realized ride time, shared time and pick-up time of each OD pair are utilized to update the platform’s estimation of waiting time, ride time and vehicle sharing time for each OD pair, which lead to the pricing strategy for the next day. For all the 100 scenarios in our experiments, we observe the platform profit stabilizes within 10 day-to-day iterations, in the sense that the relative difference in platform profit between two consecutive days consistently remains less than 1% after a number of day-to-day iterations. Typical curves for platform profits during the day-to-day evolution are presented in Appendix D. The simulated platform profit for Method 1, as presented in the following sections, always represents the average of the last five iterations (Day 6 to Day 10).
For Method 2, as introduced in Subsection 3.2, we employ two derivative-free optimization algorithms, outlined in Section 4, to solve the profit optimization problem. We obtain the optimal pricing strategies and the expected waiting and ride time of each OD pair under the optimal pricing strategies, as presented in Section 2.2, and then apply these theoretical results to simulations.

For uniform discounting strategies, it is also subject to a day-to-day evolution process. Similar to Method 1, we observe that the simulated platform profit stabilizes within 10-day iterations across all scenarios. Typical curves of the day-to-day platform profits are also presented in Appendix D. And the same as for Method 1, the average profit generated by the uniform discounting strategy is calculated as the average of the profits from day 6 to day 10. For each scenario, we calculate the platform profit and system performance under 5 uniform discounting strategies $\tilde{\theta}$ from $\{0.8, 0.85, 0.9, 0.95, 1\}$, and choose the one that leads to the highest profit for comparison with Method 1 and Method 2.

In the simulation of each scenario, we run discrete-event simulations to model the arrival of e-hailing orders, passengers’ mode choices, and passenger-vehicle matches over 15,000 simulation time units. A set of e-hailing orders is initially generated according to a Poisson Process based on a given maximal ridepooling demand rate of each OD pair. Passengers make their choice to opt for ridepooling or not based on the discounted prices and the expected waiting and ride times offered by the platform to them. For the real-time matching process, we assume the platform adopts the same matching strategy as described in Subsection 2.1.

The solution algorithms for the optimization problems and simulation experiments are coded in Python and tested on a 1.8 GHz Intel i7 laptop with 8GB RAM.

5.1.2 Performance of optimal pricing strategies

In this subsection, we first evaluate the computational efficiency of the two derivative-free algorithms proposed in Section 4 for calculating the optimal discounting strategies of Method 2. As the day-to-day adjustment of Model 1 can also be perceived as a heuristic solution algorithm for generating optimal discounting strategies, we also present the total computing time\(^6\) required by Method 1 to reach a stable profit. Table 2 presents the comparison of computational performance for five example scenarios. As we can see from the table, Method 2 generally leads to higher profit than Method 1, and SRACOS is substantially faster than BO in our problem. So, for the rest of the scenarios, we always use SRACOS for Method 2.

<table>
<thead>
<tr>
<th>No.</th>
<th>Example scenarios</th>
<th>Platform profit(CNY/h)</th>
<th>Computing time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\kappa$</td>
<td>$\eta$</td>
<td>$\kappa$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>400</td>
<td>2073.53</td>
</tr>
<tr>
<td>2</td>
<td>2.5</td>
<td>400</td>
<td>6331.04</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>400</td>
<td>13939.18</td>
</tr>
<tr>
<td>4</td>
<td>7.5</td>
<td>400</td>
<td>22182.50</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>400</td>
<td>29449.02</td>
</tr>
</tbody>
</table>

For the 100 scenarios in the grid network, we compare the platform profit under the optimal pricing strategies generated by Method 1, Method 2 and the best uniform strategies through simulation. The relative

\(^6\)The computing time for Method 1 includes not only the time spent on solving (26) 10 times, but also the simulation time for updating the estimated ride time, waiting time, and vehicle sharing time for 10 day-to-day evolution.
improvements across all scenarios are depicted in the boxplot in Fig. 5. The boxplot on the left shows that in comparison with the best uniform strategy, Method 1 consistently enhances platform profits compared to the uniform strategy, with an average improvement of 10.91%. This result emphasizes that implementing differential pricing based on OD pairs has a beneficial effect on the platform’s profit. Moreover, the boxplot on the right demonstrates the relative improvement of Method 2 over Method 1. On average, the optimal pricing strategies generated by Method 2 can increase platform profits by 7.89% in comparison to the strategies generated by Method 1.

![Figure 5: Relative improvement of the platform profit (M1 v.s Uni and M2 v.s. M1) in 100 scenarios for the grid-network experiments](image)

Next, we present the platform profit and service performance under a sample scenario where the demand scale $\kappa = 5$ and vehicle fleet size $n = 200$. We vary the benchmark uniform discounting strategies at levels of discount ratio $\bar{\theta} = \{0.8, 0.85, 0.9, 0.95, 1.0\}$. From Fig. 6, if the platform adopts a uniform discounting strategy for all ridepooling orders, the platform’s profit first increases and then decreases with $\bar{\theta}$, and the highest platform profit (13810.89 CNY/h) is achieved when $\bar{\theta} = 0.95$. However, if the platform adopts an OD-specific discount ratio according to Method 1, then the platform’s profit can increase to 14179.18 CNY/h, which is 2.67% higher than the maximal profit under the optimal uniform discounting strategy $\bar{\theta} = 0.95$. If the platform further implements the OD-specific discounting strategies according to Method 2, then its profit can be further improved to 15287.88 CNY/h, which is 7.82% higher than that under the pricing strategies derived in Method 1. This highlights the benefits of coordinated optimization of the discount ratios for different OD pairs considering the interactions among different OD pairs.

Table 3 further presents the ridepooling service performance under these pricing strategies reported by simulation. As shown in Table 3, in our grid-network experiments, all three pricing strategies achieved very high successfully pairing probability, with Method 2 (98.66%) slightly outperforming Uni (98.07%) and Method 1 (98.28%). In terms of attracting passengers to choose ridepooling, Method 1 leads by drawing the most passengers per hour on average (819 orders/hr), followed by Method 2 with 760 orders/hr, and the uniform strategy (Uni) with the least (720 orders/hr). These results are influenced by factors such as pricing, waiting time, and ride time. Under the pricing strategy derived from Method 2, the average discount ratio (0.9437) is the lowest, and passengers experience the shortest average detour time and longest vehicle
We further depict the violin plots for the distribution of discount ratios, waiting times, ride times, and detour times of each order under those three pricing strategies in Fig. 7. As can be seen from Fig. 7(a), there are obvious differences between the optimal discounting strategies generated by Method 1 and Method 2. The distributions of other metrics in Fig. 7(b)-(d) show much less significant differences.

### 5.2 Experiment on the Haikou network

To further validate the computational feasibility and performance of our proposed methods, we conducted experiments based on the real road network of Haikou, China, using an open dataset of ride-hailing orders in Haikou provided by DiDi Chuxing. In this experiment, we employed the same ridepooling demand function and pick-up time functions as those used in the grid-network experiments. The relevant parameters for these functions are detailed in Table 1 in Subsection 5.1.1.

The optimal pricing strategies generated by Method 1 and Method 2 follow exactly the same method.
as described in Subsection 5.1.1. And the obtained pricing strategies are implemented in the simulation environment based on the Haikou real road network to examine the resulting system performance. The experiment simulates the occurrence, matching, and movements of drivers and passengers with a first-come, first-served strategy in real-world scenarios, assuming all orders are willing to choose ridepooling or solo ride service, and each passenger decides whether to participate in the ridepooling service according to the probability function $f_w(x)$ defined in Section 5.1.1.

The road network of Haikou is constructed as a directed graph using OpenStreetMap data (as shown in Figure 8). The demand distribution in the experiment is generated by loading the real-world order file from May 2017 in Haikou. The total number of drivers is set to be 25% of the hourly order number. And the initial locations of drivers on the road network are randomly generated.

In the simulation experiment, all drivers are assumed to choose the shortest path between any two nodes on the network. After dropping off all onboard passengers, a fully or partially occupied vehicle becomes vacant and randomly travels through the road network until it is assigned to a passenger. Unlike in the grid-network experiments passengers are willing to wait for a vacant vehicle regardless of the waiting time for pickup, thus ensuring all orders are responded to by the platform, in Haikou network experiments, we consider the patience of passengers. Passengers will exit the matching system if their waiting time exceeds the maximum waiting time for a response, which is set to be 300 seconds in the experiment. Consequently,
the platform’s order response rate will no longer be 100%.

![Figure 8: The road network of Haikou](image)

Under the aforementioned setup, we simulated the system performance under different uniform discounting strategies and the optimal pricing strategies derived by Method 1 and Method 2. The profits of these strategies are illustrated in Figure 9. The same as in the grid network experiments, the platform profit first increases and then decreases with the uniform discount ratio; the optimal discounting strategy generated by Method 1 outperforms the best uniform discounting ratio (0.8), and the optimal strategy generated by Method 2 brings about the highest profit.

Table 4 presents the ridepooling service performance and Fig. 10 depicts the distribution of discount ratio, waiting time, ride time and detour time of each OD pair under the optimal uniform discounting strategy (Uni.: 0.8) and the optimal pricing strategies derived from Method 1 (M1) and Method 2 (M2). As shown in the table, Method 1 leads to 1.2% more profit than the uniform discounting strategy (Uni.: 0.8), but responses to 7% less ridepooling requests. This is because Method 1 converges to the highest average discount ratio in this example (as can be seen from Fig. 10). The average discount ratio under the optimal pricing strategy generated by Method 2 is similar to the optimal uniform discount ratio 0.8, and is generally lower than the optimal discount ratios generated by Method 1 in this example. Under this discounting strategy, the platform achieves the highest platform profit with the highest pairing ratio and longest total distance saving.

To assess the robustness of the proposed Method 2, we further implement the optimal pricing strategy derived under the order:driver ratio 100:25 to scenarios with different number of drivers (and the same number of orders), and compare the system performance with those under the uniform discount ratio 0.8 and the optimal pricing strategies generated by Method 1. Note that for Method 1, the pricing strategies are updated day to day according to the performance under different order-driver ratios. The results are presented in Table 5. As we can see from the table, despite the significant variations in the order-driver ratios, the pricing strategy derived by Method 2 consistently outperforms other strategies in terms of platform profit, response rates, and pairing ratio.
6 Conclusion

This paper studies the upfront price design problem for an enroute ridepooling platform. Assuming that ridepooling demand of each OD pair decreases with its price and the estimated waiting and ride time, a system of nonlinear equations is proposed to describe the complex interactions among the ridepooling demand, the expected pairing results and service quality between different OD pairs under any pricing strategy. Based on the model, we then propose and optimize an OD-based up-front pricing strategy through two methods. In the first method, the ridepooling price of each OD pair is optimized independently with a day-to-day adjustment, and in the second method, we optimize the prices of all OD pairs simultaneously, with the complex interactions among the expected detour distances and demand rates of all OD pairs being considered and captured by the system of nonlinear equations. The nonlinear and non-convex optimization

Table 4: Performance of pricing strategies in system metrics in Haikou network experiment

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Uni.: 0.8</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform profit (CNY)</td>
<td>8665</td>
<td>8770</td>
<td>9443</td>
</tr>
<tr>
<td>Ridepooling orders</td>
<td>265</td>
<td>250</td>
<td>269</td>
</tr>
<tr>
<td>Fraction of total orders that choose ridepooling</td>
<td>71.04%</td>
<td>67.02%</td>
<td>72.12%</td>
</tr>
<tr>
<td>Served orders</td>
<td>180</td>
<td>152</td>
<td>180</td>
</tr>
<tr>
<td>Pairing ratio</td>
<td>9.24%</td>
<td>11.35%</td>
<td>13.62%</td>
</tr>
<tr>
<td>Average waiting time (min)</td>
<td>3.19</td>
<td>3.13</td>
<td>3.16</td>
</tr>
<tr>
<td>Average ride time (min)</td>
<td>27.03</td>
<td>27.73</td>
<td>27.32</td>
</tr>
<tr>
<td>Average detour time (min)</td>
<td>0.65</td>
<td>0.69</td>
<td>0.74</td>
</tr>
<tr>
<td>Average vehicle sharing time (min)</td>
<td>13.96</td>
<td>13.21</td>
<td>12.45</td>
</tr>
<tr>
<td>Total distance saving due to ridepooling (km)</td>
<td>156</td>
<td>147</td>
<td>165</td>
</tr>
</tbody>
</table>
problem in the second method is solved by two optimization methods respectively: Bayesian optimization and classification-based optimization. Based on a 15*15 grid network with 30 OD pairs and the real road network of Haikou (China), we conduct numerical experiments to examine the efficiency of the two algorithms and the system performance under the optimal pricing strategies generated by the two methods. It is found that in comparison with a uniform discounting strategy, OD-based discounting strategies generated by both methods can bring about 10% more profit to the platform. And in comparison with the independently optimized discounting strategies generated by the first method, the system optimal discounting strategy generated by the second method can further improve the platform profit by 7.89% and 7.67% on average in our grid-network and real road network experiments.

This paper made the first attempt to optimize OD-based ex-ante pricing strategies for an e-hailing platform that operates enroute ridepooling service only. However, in reality, platforms like DiDi in China, operate both solo-ride and ridepooling services. In this case, the passenger waiting time for solo-ride services may be positively or negatively related to the passenger waiting time for ridepooling services, depending on the platform’s vacant vehicle allocation strategy between the two types of services. How to model the passenger waiting times for solo-ride and ridepooling services when the platform operates the same vehicle fleets for both services, and how to design the OD-based pricing strategy for both types of services are interesting and important topics that we leave for future study. And as we point out, the relocation movement of vacant vehicles would apparently affect passenger waiting time, therefore influence the platform’s pricing
Table 5: Performance of Uni.: 0.8 v.s. M1 v.s. M2 under different order-driver ratios

<table>
<thead>
<tr>
<th>Order-driver ratio</th>
<th>Pricing Strategy</th>
<th>Platform profit</th>
<th>Response rate</th>
<th>Pairing ratio</th>
<th>Average detour time (min)</th>
<th>Average waiting time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100:10</td>
<td>Uni.: 0.8</td>
<td>5615</td>
<td>0.42</td>
<td>0.11</td>
<td>1.16</td>
<td>3.52</td>
</tr>
<tr>
<td></td>
<td>M1</td>
<td>6165</td>
<td>0.43</td>
<td>0.13</td>
<td>1.06</td>
<td>3.67</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>6378</td>
<td>0.45</td>
<td>0.21</td>
<td>1.23</td>
<td>3.6</td>
</tr>
<tr>
<td>100:50</td>
<td>Uni.: 0.8</td>
<td>7808</td>
<td>0.61</td>
<td>0.09</td>
<td>0.69</td>
<td>3.35</td>
</tr>
<tr>
<td></td>
<td>M1</td>
<td>8105</td>
<td>0.57</td>
<td>0.09</td>
<td>0.69</td>
<td>3.28</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>8655</td>
<td>0.62</td>
<td>0.12</td>
<td>0.7</td>
<td>3.37</td>
</tr>
<tr>
<td>100:75</td>
<td>Uni.: 0.8</td>
<td>8458</td>
<td>0.66</td>
<td>0.09</td>
<td>0.47</td>
<td>3.18</td>
</tr>
<tr>
<td></td>
<td>M1</td>
<td>8835</td>
<td>0.62</td>
<td>0.09</td>
<td>0.47</td>
<td>3.12</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>9319</td>
<td>0.67</td>
<td>0.13</td>
<td>0.58</td>
<td>3.07</td>
</tr>
</tbody>
</table>

strategy design. For simplicity, this paper only considers scenarios where drivers cruise within the zones where they complete their previous services. To further enhance service quality and platform profit, the platform could optimize its relocation and pricing strategies coordinately. Furthermore, in our study, we assume the total vehicle fleet size is constant, so the impacts of the platform’s payment strategy on drivers are completely ignored. The payment strategy are pivotal as they significantly influence drivers’ willingness to accept ridepooling orders, creating a direct link to the overall passenger experience. So devising suitable payment strategies for drivers that harmonize their incentives with passenger satisfaction is another vital aspect for future exploration.

CRediT authorship contribution statement

Siying Wang: Conceptualization, Methodology, Investigation, Software, Validation, Writing – original draft. Xiaolei Wang: Conceptualization, Investigation, Methodology, Writing – original draft, Writing – review & editing. Chen Yang: Software, Validation, Writing – review. Wei Liu: Conceptualization, Writing – review & editing. Xiaoning Zhang: Conceptualization, Writing – review & editing.

Acknowledgments

Funding sources: The work described in this study was supported by grants from the National Natural Science Foundation of China under Project No. 72021002, No. 72022013 and No. 71974146. Xiaolei Wang was also supported by the Fundamental Research Funds for the Central Universities, China. Declaration of Interest statement: None.

References


A List of notations

<table>
<thead>
<tr>
<th>Sets</th>
<th>Interpretations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Set of nodes on the road network</td>
</tr>
<tr>
<td>$A$</td>
<td>Set of links on the road network</td>
</tr>
<tr>
<td>$W$</td>
<td>Set of OD pairs</td>
</tr>
<tr>
<td>$A_w$</td>
<td>Set of links traversed by the solo-ride routes between OD pair $w \in W$</td>
</tr>
<tr>
<td>$\hat{A}_w = A_w \cup {a_0^w}$</td>
<td>The union set of $A_w$ and ${a_0^w}$, where $a_0^w$ is a dummy link representing the passenger waiting state at the origin of each OD pair</td>
</tr>
<tr>
<td>$S$</td>
<td>Set of seeker-states $s(w), w \in W$</td>
</tr>
<tr>
<td>$T$</td>
<td>Set of taker-states $t(a, w), a \in \hat{A}_w, w \in W$</td>
</tr>
<tr>
<td>$S_{t(a,w)}$</td>
<td>Set of matching seeker-states for taker-state $t(a, w) \in T$</td>
</tr>
<tr>
<td>$T_{s(w)}$</td>
<td>Set of matching taker-states for seeker-state $s(w) \in S$</td>
</tr>
<tr>
<td>$T_{s(w)}^s(a, w)$</td>
<td>Set of taker-states that have a higher matching priority with seeker-state $s(w)$ compared to the taker-state $t(a, w)$</td>
</tr>
<tr>
<td>$W(i)$</td>
<td>Set of OD pairs whose destination node is $i \in N$</td>
</tr>
<tr>
<td>$M(i)$</td>
<td>Set of matching seeker-taker pairs whose ridepooling trips end at node $i \in N$</td>
</tr>
<tr>
<td>$Z$</td>
<td>Set of zones on the road network. For node $i \in N$, $z(i) \in Z$ is the zone to which node $i$ belongs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exogenous variables</th>
<th>Interpretations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Lambda$</td>
<td>The maximal ridepooling demand rate for each OD pair $w, w \in W$</td>
</tr>
<tr>
<td>$p^s_w$</td>
<td>The solo-ride price of OD pair $w, w \in W$</td>
</tr>
<tr>
<td>$\bar{\beta}$</td>
<td>Passengers’ value of time</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>The perceived safety cost of ridepooling</td>
</tr>
<tr>
<td>$C^s_w$</td>
<td>The generalized solo-ride cost between OD pair $w \in W$</td>
</tr>
<tr>
<td>$\vartheta$</td>
<td>Average vehicle speed</td>
</tr>
<tr>
<td>$n$</td>
<td>Total vehicle fleet size</td>
</tr>
<tr>
<td>$c'$</td>
<td>The platform’s payment to drivers for each occupied unit time</td>
</tr>
<tr>
<td>$t(a)$</td>
<td>Link travel time on link $a \in A$</td>
</tr>
<tr>
<td>$\bar{t}$</td>
<td>The maximal pick-up time by partially occupied vehicles allowed by the pairing condition</td>
</tr>
<tr>
<td>$\bar{t}_{pk}$</td>
<td>The maximal detour time for both passengers in a ridepooling trip</td>
</tr>
<tr>
<td>$\bar{g}$</td>
<td>The maximal pick-up time by vacant vehicles</td>
</tr>
<tr>
<td>$t^s_w(a, w)$</td>
<td>The solo ride time between OD pair $w \in W$</td>
</tr>
<tr>
<td>$R^s(a, w)$</td>
<td>The ride time of passengers in taker-state $t(a, w)$ getting paired with passengers in seeker-state $s(\omega) \in S_{t(a,w)}$</td>
</tr>
<tr>
<td>$R^s_{t(a,w)}(a, w)$</td>
<td>The ride time of passengers in seeker-state $s(\omega) \in S$ getting paired with passengers in taker-state $t(a, w) \in T_{s(\omega)}$</td>
</tr>
<tr>
<td>$\epsilon_{s(a, w)}$</td>
<td>The vehicle sharing time in a ridepooling trip formed by passengers in seeker-state $s(w) \in S$ and taker-state $t(a, \omega) \in T_{s(a)}$</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>Area of each zone</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision variables</th>
<th>Interpretations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_w$</td>
<td>The discount ratio for ridepooling orders between OD pair $w, w \in W$</td>
</tr>
<tr>
<td>$\Theta$</td>
<td>The vector of discount ratios</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Endogenous variables</th>
<th>Interpretations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda^p_w$</td>
<td>The ridepooling demand rate for passengers between OD pair $w \in W$, and $\lambda^p = (\lambda^p_w, w \in W)$</td>
</tr>
<tr>
<td>$t_w$</td>
<td>The average waiting time of ridepooling trips between OD pair $w \in W$, and $t^p = (t_w, w \in W)$</td>
</tr>
<tr>
<td>$\bar{t}_{pk}$</td>
<td>The average pick-time of vacant vehicles for OD pair $w \in W$, and $\bar{t}<em>{pk} = (\bar{t}</em>{pk}, w \in W)$</td>
</tr>
</tbody>
</table>
The average ride time of ridepooling trips between OD pair \( w \in W \), and \( T^w_p \) = \( (T^w_p, w \in W) \)

Pairing probability of passengers in seeker-state \( s(w) \in S \), and \( p_s = (p_s^{(w)}, s(w) \in S) \)

Pairing probability of passengers in taker-state \( p_{t(a,w)} \), \( t(a,w) \in T \), and \( p_t = (p_{t(a,w)}, t(a,w) \in T) \)

The probability of having at least one passenger in taker-state \( t(a,w) \in T \), and \( \rho = \{ p_{t(a,w)}, t(a,w) \in T \} \)

The arrival rate of pairing opportunities for taker-state \( t(a,w) \) with seeker-state \( s(\omega) \), and \( \eta = (\eta_t(a,w), s(\omega) \in S_t(a,w), t(a,w) \in T) \)

The arrival rate of unpaired passengers in taker-state \( t(a,w) \in T \), and \( \lambda_t = (\lambda_t(a,w), t(a,w) \in T) \)

The aggregate arrival rate of pairing opportunities for taker-state \( t(a,w) \in T \)

The time span of each taker-state \( t(a,w) \in T \)

The pairing probability of orders between OD pair \( w, w \in W \)

The expected vehicle sharing time of ridepooling passengers between OD pair \( w, w \in W \)

The generalized ridepooling cost between OD pair \( w, w \in W \)

The trip completion rate at node \( i \in N \)

The relocation flow rate of vacant vehicles to zone \( z \in Z \)

The expected number of vacant vehicles at zone \( z \in Z \) at each instant, and \( n^v = (n^v_z, z \in Z) \)

The expected number of vacant vehicles at each instant

### B The system of nonlinear equations in Wang et al. (2021)

In this appendix, we briefly introduce how Wang et al. (2021) models the interactions between the variables mentioned in Subsection 2.2.1: (1) \( p_s(w) \), that is, the probability of a seeker being matched in seeker-state \( s(w) \in S \); (2) \( p_{t(a,w)} \), that is, the probability of a taker being matched at taker-state \( t(a,w) \in T \); (3) \( \rho_t(a,w) \), that is, the probability of having at least one taker in state \( t(a,w) \in T \) at any moment; (4) \( \eta_t(a,w) \), that is, the aggregate arrival rate of pairing opportunities for takers in state \( t(a,w) \in T \); (5) \( \lambda_t(a,w) \), that is, the average arrival rate of unmatched passengers for taker-state \( t(a,w) \in T \).

A seeker would be immediately matched if takers appear in its matching taker-state \( t(a,w) \in T_s(w) \), so the matching probability \( P_s(w) \) of a seeker in state \( s(w) \) depends on the existent probability \( \rho_t(a,w) \) of takers in state \( t(a,w) \in T_s(w) \):

\[
 p_s(w) = \begin{cases} 
 1 - \prod_{t(a,w) \in T_s(w)} (1 - \rho_t(a,w)), & \text{if } T_s(w) \neq \emptyset, \\
 0, & \text{otherwise}
 \end{cases}, s(w) \in S; 
\]

(38)

A taker in state \( t(a,w) \) may be matched if pairing opportunities arrive (i.e. seekers that satisfy the matching priority rule newly appear in its matching seeker-state \( s(w) \in S_t(a,w) \)) during the time \( \tau_t(a,w) \) that the taker stays at state \( t(a,w) \), so the matching probability \( p_{t(a,w)} \) of takers in state \( t(a,w) \) depends on the arrival rate
of pairing opportunities $\eta_{t(a,w)}$ in all its matching seeker-states $s(w) \in S_{t(a,w)}$:

$$
\eta_{t(a,w)}^{s(\omega)} = \begin{cases} 
\lambda_\omega, & \text{if } s(\omega) \in S_{t(a,w)} \text{ and } T_{s(\omega)}^{\sim t(a,w)} = \emptyset \\
\lambda_\omega \prod_{t(a',\omega) \in T_{s(\omega)}^{\sim t(a,w)}} (1 - p_{t(a',\omega)}), & \text{if } s(\omega) \in S_{t(a,w)} \text{ and } T_{s(\omega)}^{\sim t(a,w)} \neq \emptyset
\end{cases}, \quad (39)
$$

$$
\eta_{t(a,w)} = \begin{cases} 
\sum_{s(\omega) \in S_{t(a,w)}} \eta_{t(a,w)}^{s(\omega)}, & \text{if } S_{t(a,w)} \neq \emptyset \\
0, & \text{otherwise}
\end{cases}, \quad t(a,w) \in T, \quad (40)
$$

$$
p_{t(a,w)} = \begin{cases} 
P \left( n(\tau_{t(a,w)}) \geq 1 \big| \eta_{t(a,w)} \right) = 1 - \exp \left( -\eta_{t(a,w)} \tau_{t(a,w)} \right), & \text{if } \eta_{t(a,w)} > 0 \\
0, & \text{if } \eta_{t(a,w)} = 0
\end{cases}, \quad t(a,w) \in T; \quad (41)
$$

where $\tau_{t(a,w)}$ is the maximal passenger dwelling time in taker-state $t(a,w)$ defined in Eq. (3).

The existent probability $\rho_{t(a,w)}$ of takers in state $t(a,w)$ at any moment depends on the average arrival rate $\lambda_{t(a,w)}$ of unmatched passengers for taker-state $t(a,w)$ and the expected time that every taker is available for matching in its taker-state:

$$
\rho_{t(a,w)} = \lambda_{t(a,w)} \tau_{t(a,w)} = \begin{cases} 
\lambda_{t(a,w)} [1 - \exp \left( -\eta_{t(a,w)} \tau_{t(a,w)} \right)], & \text{if } \eta_{t(a,w)} > 0 \\
\lambda_{t(a,w)} \tau_{t(a,w)}, & \text{if } \eta_{t(a,w)} = 0
\end{cases}, \quad t(a,w) \in T. \quad (42)
$$

Since takers appear in state $t(a,w)$ only if she/he is not matched in previous states, the average arrival rate $\lambda_{t(a,w)}$ of unmatched taker for state $t(a,w)$ depends on matching probability of a seeker $p_{\omega(w)}$ and takers $P_{t(a,w)}$:

$$
\lambda_{t(a,w)} = \begin{cases} 
\lambda_{a,w} [1 - p_{\omega(w)}], & n = 0 \\
\lambda_{t(a,w)} \left( 1 - p_{t(a,w)} \right), & 1 \leq n \leq |A_w|, \quad t(a,w) \in T.
\end{cases} \quad (43)
$$

## C Proof of Proposition 1

**Proof.** As discussed in Subsection 2.3, the system of nonlinear equations can be perceived as a fixed point problem. According to Brouwer’s fixed point theorem (Fuente, 2000), a fixed point problem $X = F(X)$ admits at least one solution on $\Omega$, if and only if $\Omega$ is compact and non-empty, $F(X)$ is continuous on $\Omega$, and $F(X) \in \Omega$ for any $X \in \Omega$.

From Eq. (24), it is apparent that $\Omega$ is compact and non-empty. So it suffices to show the continuity of $F(X)$ on $\Omega$ and that $F(X) \in \Omega$.

- **Continuity of $F(X)$ on $\Omega$.** From Eqs. (1), (5), (8), (9), (10), (40) and (3), $C_{\omega}^{\rho}(\mathbf{X}, \theta)$, $P_{\omega}(\mathbf{X})$, $v_{\omega}(\mathbf{X})$, $v_{\omega}(\mathbf{X})$, $n_{\omega}(\mathbf{X})$ and $\tau_{\omega}(\mathbf{X})$ are all continuous functions of $\mathbf{X}$ on $\Omega$. So the continuity of $\lambda_{\omega}(\mathbf{X}, \theta)$, $t_{\omega}(\mathbf{X})$, $T_{\omega}(\mathbf{X})$, $i_{\omega}(\mathbf{X})$, $n_{\omega}(\mathbf{X})$, $p_{\omega}(\mathbf{X})$, $p_{\omega}(\mathbf{X})$, $\eta(\mathbf{X})$, and $\lambda_{\omega}(\mathbf{X})$ on $\Omega$ given in Eqs.
(14)-(21) and (23) can be readily concluded. To validate the continuity of \( F(X) \), we thus only need to prove the continuity of \( \rho(X) \). From Eq. (22), \( \rho_{t(a,w)}(X) \) is continuous at any \( X \in \Omega \) with \( \eta_{t(a,w)} > 0 \). For \( X^* \in \Omega \) with \( \eta_{t(a,w)}^* = 0 \), by L’Hôpital’s role, we have

\[
\lim_{X \to X^*} \rho_{t(a,w)}(X) = \rho_{t(a,w)}(X^*) \quad \text{for} \quad X^* \in \Omega \quad \text{with} \quad \eta_{t(a,w)}^* = 0, \quad \rho_{t(a,w)}(X) \text{ is thus continuous at any} \quad X \in \Omega.
\]

\( \cdot \) \( F(X) \in \Omega \). Since \( f_w(x) \in [\varepsilon, \tilde{\lambda}_w] \) for any \( x \in R \), we have \( \lambda_w^*(X) \in [\varepsilon, \tilde{\lambda}_w] \) for any \( X \in \Omega \). Since \( g(x) \) is decreasing with \( x \) and \( g(0) = \tilde{g} \), we have \( t_w^*_a(X) \in [0, \tilde{g}] \) and consequently \( t_w(X) \in [0, \max \{ \tilde{g}, \tilde{i}_r/2 \} ] \) for any \( X \in \Omega \). Since the maximum detour time of ridepooling trip is \( \tilde{i}_d \), we must have \( T_w^p(X) \geq (t_w^* + \tilde{i}_d) \tilde{n}_w \), and consequently \( T_w^p(X) \in [0, t_w^* + \tilde{i}_d] \) for any \( X \in \Omega \). Since \( n \geq \sum_{w \in W} (t_w^* + \tilde{i}_d) \tilde{\lambda}_w \), we have \( n(X) \in [0, n] \) for any \( X \in \Omega \), and consequently \( n_t(X) \in [0, n] \). And from Eqs. (19)-(21) and (23), it is not difficult to verify that \( p_{t(a,w)}(X), \rho_{t(a,w)}(X) \in [0, 1] \) and \( \lambda_{t(a,w)}(X), \eta_{t(a,w)}(X) \in [0, \tilde{\lambda}_w] \) holds for any \( X \in \Omega \). For \( \rho_{t(a,w)}(X) \), at \( X \in \Omega \) with \( \eta_{t(a,w)} = 0 \), we must have \( \rho_{t(a,w)}(X) = \lambda_{t(a,w)}(X^*) \leq \tilde{\lambda}_w \max \{ \tilde{g}, \max \{ t_d, a \in A \} \} \). At \( X \in \Omega \) with \( \eta_{t(a,w)} > 0 \), we have

\[
\rho_{t(a,w)}(X) = \frac{\lambda_{t(a,w)}[1 - \exp(-\eta_{t(a,w)} \tilde{t}_{t(a,w)})]}{\eta_{t(a,w)}} 
\]

\[
\leq \lambda_{t(a,w)} \tilde{\lambda}_{t(a,w)} \left[ \int_0^{\tilde{t}_{t(a,w)}} \eta_{t(a,w)} \exp(-\eta_{t(a,w)} t) dt + \tilde{\lambda}_{t(a,w)} \exp(-\eta_{t(a,w)} \tilde{t}_{t(a,w)}) \right] 
\]

\[
= \lambda_{t(a,w)} \tilde{\lambda}_{t(a,w)} \left[ -\exp(-\eta_{t(a,w)} \tilde{t}_{t(a,w)}) + 1 + \exp(-\eta_{t(a,w)} \tilde{t}_{t(a,w)}) \right] 
\]

\[
= \lambda_{t(a,w)} \tilde{\lambda}_{t(a,w)} \max \{ \tilde{g}, \max \{ t_d, a \in A \} \} 
\]

So \( \rho_{t(a,w)}(X) \leq \tilde{\lambda}_w \max \{ \tilde{g}, \max \{ t_d, a \in A \} \} \) holds for all \( X \in \Omega \). Recall the assumption that \( \tilde{\lambda}_w \max \{ \tilde{g}, \max \{ t_d, a \in A \} \} \leq 1 \), we readily have \( \rho_{t(a,w)}(X) \in [0, 1] \) for any \( X \in \Omega \).

Based on the Brouwer’s fixed point theorem, we then can conclude that the existence of solutions is guaranteed for the system of nonlinear equations (1)-(13) and (38)-(43) under any discounting strategy \( \theta \) when \( \tilde{\lambda}_w \max \{ \tilde{g}, \max \{ t_d, a \in A \} \} \leq 1 \) and \( n \geq \sum_{w \in W} (t_w^* + \tilde{i}_d) \tilde{\lambda}_w \).

\[ \square \]

D Day-to-day system evolution

In this appendix, we present the results of numerical experiments to illustrate the convergence of the platform’s ridepooling profit when applying the day-to-day adjustment method of Method 1, as described in
Subsection 3.1 and benchmark uniform discounting strategy, as introduced in Subsection 5.1 under grid network. The experiments consider different scenarios, with over 100 scenarios investigated in the grid-network experiments and 3 scenarios examined in the Haikou real road network experiments. In each scenario, we apply 10 iterations of day-to-day adjustment. For the uniform discounting strategy, during each iteration, the historical average waiting and ride time data are updated through simulation experiments to obtain the expected waiting and ride time for the next day. As for Method 1, in each iteration, historical waiting and ride time data are similarly updated to calculate the expected waiting and ride time, and in accordance with the pricing optimization method described in Section 3.1, we obtain a new round of optimal discounting strategy. We conclude the iterative process when the relative difference in platform profit between two iterations consistently remains less than 1%.

From the experimental results across all scenarios, including both on the grid network and the Haikou network, it was observed that after 5 iterations of day-to-day evolution, platform profit comes to converge, and the relative difference in platform profit between two iterations for both uniform discounting strategy and optimal strategy derived by Method 1 consistently remain below 1%. We present the evolution of the platform profit for selected scenarios, as shown in Fig. 11. Fig. 11 (a) display the results for three example scenarios in the grid-network experiments, namely, demand scale $\kappa = 1$, vehicle fleet size $n = 175$; $\kappa = 5$, $n = 200$; and $\kappa = 10$, $n = 400$. Fig. 11 (b) illustrates the results for the Haikou network with the order-driver ratio of 100:75. As we can see from the figures, the initial iterations exhibit distinct trends influenced by the initial expected ride/vehicle sharing time derived from historical distance information. Nevertheless, across all example scenarios, the uniform discounting strategy and optimal pricing strategies derived by Method 1 converge rapidly, typically within around 5 iterations for both grid-network and the Haikou network experiments.

![Figure 11: Evolution of the platform’s profit within the day-to-day adjustment iterations under the uniform pricing strategy and Method 1](image-url)