8-2024

Optimization of customer service and driver dispatch areas for on-demand food delivery

Jingfeng YANG
Singapore Management University, jfyang.2018@phdcs.smu.edu.sg

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

Hai WANG
Singapore Management University, haiwang@smu.edu.sg

Optimization of customer service and driver dispatch areas for on-demand food delivery

Jingfeng Yang, Hoong Chuin Lau, Hai Wang *

School of Computing and Information Systems, Singapore Management University, 80 Stamford Rd, Singapore

*Corresponding author. Contact: haiwang@smu.edu.sg

Published in Transportation Research Part C (2024) 165, 104653. DOI: 10.1016/j.trc.2024.104653

Abstract: With the rapid development and popularization of mobile and wireless communication technologies, on-demand food delivery (OFD) platforms have been able to connect restaurants, customers, and drivers in real time, drastically changing dining and food delivery services. Motivated by the critical need for supply and demand management in the on-demand food delivery market, we focus on the optimization of customer service area and driver dispatch area for on-demand food delivery services. Specifically, for each restaurant, the platform needs to decide the (1) customer service area (CSA), i.e., the surrounding area within which customers can see the restaurant's information and order food from it; and (2) driver dispatch area (DDA), i.e., the surrounding area within which drivers can see the restaurant's information and deliver orders from it. Hence, our focus is on the area sizing optimization problem that enables the platform to dynamically balance supply and demand by adjusting the radii of its customer service and driver dispatch areas. Leveraging a real dataset from a food delivery platform, we propose a data-driven optimization framework that combines discrete choice models for demand estimation, machine learning methods for order delivery time prediction, and mathematical programming for the optimization of CSA and DDA areas. The objective is to maximize the total number of orders served with a service level requirement on order delivery time. We integrate the model tree prediction model for delivery time prediction into our optimization model, resulting in a Mixed Integer Quadratically Constrained Program (MIQCP), that can be solved efficiently. Extensive experiments using real-world data demonstrate that the proposed framework outperforms several benchmarks in practice.

Keywords: On-demand food delivery, Customer service area, Driver dispatch area, Data-driven optimization

1. Introduction

Advanced technologies such as smartphones and wireless communications are transforming transportation-enabled urban services in many ways at a rapid pace. The emergence and success of on-demand passenger and logistics service platforms is one of the most notable innovations (Agatz et al., 2024). As one of the key innovations, on-demand food delivery platforms such as Uber Eats, DoorDash, Grab Food, and Meituan that provide door-to-door food delivery services have achieved great success in the past few years, especially accelerated by the COVID-19 pandemic. For example, Grab Food, Southeast Asia's largest food delivery service provider in 480 cities in 8 countries, has a GMV of $7.6 billion with a 29% annual increase in 2021. Meituan, the largest on-demand food delivery platform in China, serves more than 30 million orders daily and generates a profit of 4.71 billion RMB in 2020. According to a report by McKinsey & Company (2021), the food delivery market has doubled during the COVID-19 pandemic with a market value of over $150 billion in the US.

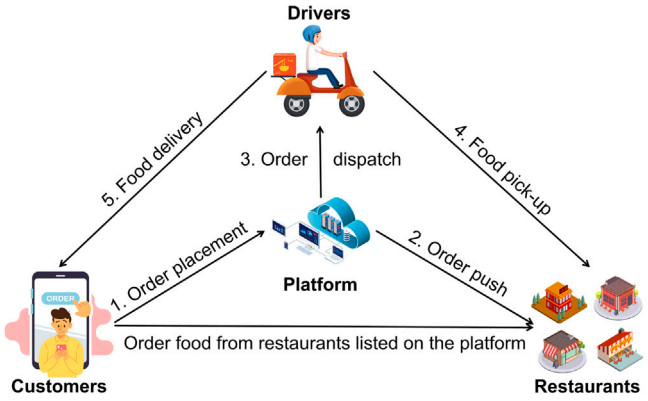

Fig. 1. Illustration of the on-demand food delivery service.

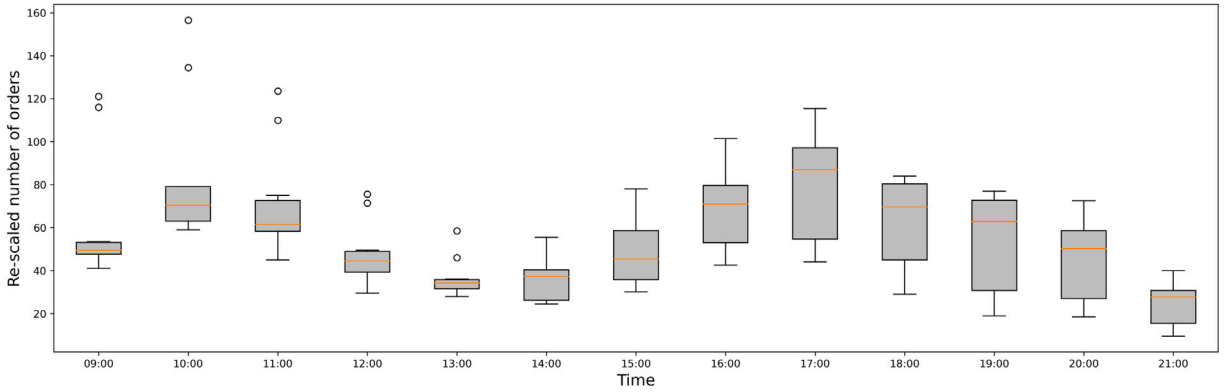

Fig. 2. The distribution of (re-scaled) order numbers across hours.

As shown in Fig. 1, there are three parties other than the platform in the on-demand food delivery market: customers, restaurants, and drivers. Customers choose and order food from a restaurant nearby listed on the platform. Once an order is placed, the platform notifies the restaurant to prepare food and broadcasts/dispatches the order and delivery task information to drivers waiting nearby. A driver then picks up food from the restaurant and delivers it to the customer.

In a typical food delivery service, demand and supply are both time-dependent, erratic, and uncertain. For example, using the real data from a crowd-sourcing food delivery platform, Fig. 2 demonstrates the distributions of hourly orders from 09:00 to 21:00. We can see that the order volume increases greatly during the peak period before noon and evening, resulting in a lack of drivers and longer delivery time. Customers still have high expectations for delivery time and may abandon the platform and seek alternatives if the delivery time is too long. For such platforms, it is very challenging and requires great efforts to balance time-dependent supply and demand.

To coordinate the balance between supply and demand in on-demand platforms, a common approach is dynamic pricing (Taylor, 2018; Feldman et al., 2018; Yang et al., 2020b; Bahrami et al., 2023). When a restaurant is busy with a large number of orders, the platform could raise the delivery fee to discourage customers from ordering from that restaurant and also encourage drivers to deliver orders for that restaurant. Another approach is to adjust the restaurant's customer service area (Yildiz and Savelsbergh, 2019b; Ulmer et al., 2022; Ding et al., 2020). If the number of orders is too high and the delivery time is too long (i.e., delivery supply is less than demand), the platform could decrease the restaurant's service area to reduce demand; in contrast, if the number of orders is too low and the delivery time is short, (i.e., delivery supply is more than demand), then the service area can be enlarged to serve more customers.

Both of the aforementioned approaches only concentrate on the demand side—the management of customer orders, and ignore the supply side (the drivers). In this paper, we propose and focus on a new approach to balance supply and demand — by adjusting the customer service area and driver dispatch area simultaneously. Specifically, for each restaurant in a specified operating time horizon, the platform decides the (1) customer service area (CSA), i.e., the radius of the surrounding area within which customers can see the restaurant's information and order food from it; and (2) driver dispatch area (DDA), i.e., the radius of the surrounding area within which drivers can see the restaurant's information and deliver orders from it. Dynamic modifications to the driver dispatch area (DDA) of a restaurant hold significant business importance. Long pickup distances adversely impact both the order delivery time and driver utilization. In practical scenarios, the restaurant's order demand fluctuates dynamically, particularly during midday

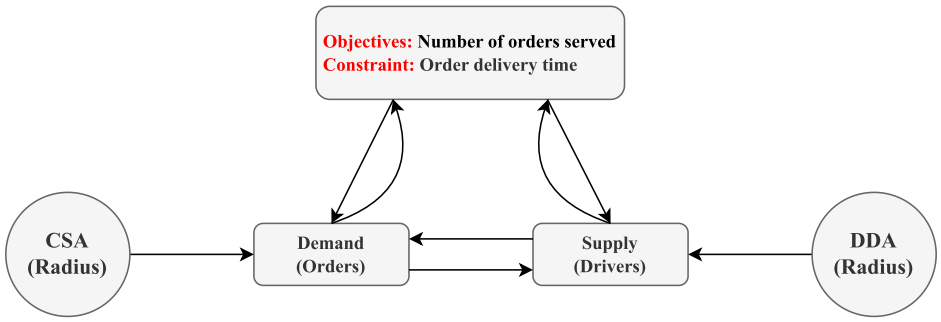

Fig. 3. Restaurant area sizing optimization problem in the on-demand food delivery service.

and evening peak hours. Ensuring timely delivery necessitates the potential expansion of the DDA, which could attract additional drivers to prevent delays and ensure timely deliveries. On the contrary, during off-peak hours, the adjustment of a restaurant's DDA allows for effective regulation of the number of drivers allocated to serve that restaurant. Assigning orders to drivers within a limited DDA radius aids in controlling the distance covered by drivers. When a driver is assigned a delivery significantly distant from their current location, it may result in extended delivery times and heighten the prospect of order delays, ultimately casting a negative effect on customer experience.

This consideration motivates us to consider an optimization problem to coordinate the supply and demand simultaneously, which we refer to in this study as the Restaurant Area Sizing Optimization (RASO) problem. We propose a data-driven optimization framework that determines the optimal radius of the customer service area and driver dispatch area. The objective is to maximize the total number of orders served while ensuring a service level requirement on order delivery time.

Finding the optimal radii of CSA and DDA is challenging. As shown in Fig. 3, while the radius of CSA affects the number of orders (i.e., demand side) and the radius of DDA affects the number of potential delivery drivers (i.e., supply side) for each restaurant, the interactions of demand and supply will collectively affect the order delivery time in a complicated way, owing to the high complexity in dynamic order-dispatching and driver-routing. To address these challenges, we first study the relationship between the radius of CSA and the number of orders for each restaurant. Second, we examine the number of drivers, which depends on the radius of the DDA. Third and more importantly, we explore how various factors related to demand, supply, and others, affect the order delivery time. Moreover, in practical scenarios, due to the flexibility of drivers in delivering orders from various restaurants, those operating within the driver dispatch area (DDA) of one restaurant may not exclusively serve that restaurant, especially if they are also within the DDA for other restaurants. This introduces additional complexity to the problem. In the context of this research, our focus is on treating restaurants within a geographical neighborhood as a cluster, where we assume each restaurant within the cluster share a common CSA and DDA, and that there is no dependency across clusters.

In summary, the main contributions of this paper are summarized as follows:

- We propose and solve an innovative operational problem, i.e., the Restaurant Area Sizing Optimization (RASO) problem for on-demand food delivery services, which determines the optimal radii of CSA and DDA for restaurants simultaneously to balance supply and demand, with the objective of maximizing the number of orders served.
- Specifically, we propose a data-driven optimization framework to solve the RASO problem. On the demand side, a discrete choice model is developed to characterize the relationship between customer order behavior and customer-restaurant distance, and then estimate the number of orders for the restaurants. On the supply side, several machine learning models are proposed to predict the order delivery time with varying sizes of CSA and DDA.
- We integrate a model tree prediction of order delivery time and formulate the RASO problem as a Mixed Integer Quadratically Constrained Program (MIQCP), which can be solved efficiently.
- We perform a set of extensive numerical experiments using a real-world dataset. The computational study demonstrates that the proposed framework can significantly improve the number of orders served and outperform benchmark methods.

The reminder of the paper is organized as follows. Section 2 reviews related literature. Section 3 describes the research problem. Section 4 presents the data-driven framework for joint optimization of CSA and DDA. Section 5 presents the real data, simulator, and the performance of the customer demand prediction and order delivery time prediction. Section 6 presents experimental results of the proposed model and other benchmark methods. Finally, we conclude and discuss future research in Section 7.

2. Literature review

Our work is closely related to three streams of literature: (1) supply and demand management for on-demand platforms, (2) order-dispatching and driver-routing, and (3) delivery time prediction.

Supply and demand management for on-demand platforms. Customer demand and driver supply are two sides of on-demand platforms, which are usually unbalanced. Many scholars paid attention to supply and demand management in food delivery (Wang,

2022; Liang et al., 2023) and ride-sourcing markets (Wang and Yang, 2019), including service area management and dynamic pricing.

In the area of service area management, to the best of our knowledge, Yildiz and Savelsbergh (2019b) are the first to analyze how service area size impacts the profit of a delivery platform. The authors derive a functional dependency between revenue and service area size and other parameters, such as customer arrival rate, revenue per customer, compensation per delivery and miles traveled, and customer satisfaction. Mankad et al. (2019) also investigates how the service area, which is called the "service outlet", affects a store's supply and demand. Ulmer et al. (2022) study the dynamic service area sizing problem in an urban delivery and model it as a Markov Decision Process, and propose a value function approximation method to decide the radius of the customer service area. Auad et al. (2020) study the customer service area problem and focus on matching the levels of supply and demand. A Mixed Integer Programming (MIP) model is proposed to determine the optimal service radius that maximizes the number of orders served. Ding et al. (2020) study restaurant delivery scope problem for restaurants using machine learning algorithms to rank potential delivery scopes and then combinatorial optimization to select the delivery scope. In the on-demand ride-sourcing market, Yang et al. (2020a) investigate the optimal matching radius with an objective of enhancing system efficiency in terms of passenger waiting time, vehicle utilization, and matching rate.

In the area of surge pricing, Tong et al. (2020) study the dynamic pricing strategy for Online-to-Offline (O2O) on-demand food service in China from both theoretical and empirical perspectives. They demonstrate that platforms that employ dynamic pricing strategies have much more demand than platforms that use static pricing systems. MacKay et al. (2022) provide an empirical investigation of the adoption of a dynamic pricing algorithm in an environment with time-varying demand and firm capacity restrictions in restaurants. They find that dynamic pricing can reduce demand volatility, which results in an increase in the proportion of transactions during periods of low demand. Bai et al. (2019) propose a pricing framework for an on-demand service platform, and examine how various factors affect the optimal price, wage, and commission with an objective of maximizing the platform's profit or social welfare. Numerous studies have investigated the implementation of pricing strategies to regulate supply and demand in ride-sourcing markets. Bimpikis et al. (2019) examine spatial pricing discrimination in a ride-sharing platform, emphasizing the influence of demand patterns on pricing, profits, and consumer surplus. Zhu et al. (2021) propose a Mean-field Markov Decision Process to depict the dynamics in ride-sourcing systems with mixed agents for spatial–temporal subsidies to solve the supply–demand imbalance issue. Liu et al. (2023) investigates the impacts of the prevailing threshold-based driver incentives on ride-sourcing drivers' labor supply with an extensive ride-sourcing dataset. Extensive relevant research exists on the pricing and incentive challenges within the ride-sourcing market, spanning various perspectives, such as pricing for pooling services (Ke et al., 2020; Zhang and Nie, 2021; Bahrami et al., 2022; Ke et al., 2022; Liu and Ouyang, 2023); pricing for platform's regulations (Li et al., 2022; Vignon et al., 2023); driver incentives and multi-homing (Sun et al., 2019; Angrist et al., 2021; Guo et al., 2023); and third-party platform-integration (Zhou et al., 2022).

Order dispatching and driver routing. Order-dispatching is important for on-demand transportation services such as ride-hailing, ride-sharing, and food delivery. For ride-hailing, the task is dispatching vehicles to serve passengers; for food delivery, the task is dispatching drivers with recommendations of delivery routes to deliver food within a promised time period.

For recent studies on ride-sharing, the reader may refer to Agussurja et al. (2019), Qin et al. (2020), Lyu et al. (2024), Luo et al. (2023). For food delivery, the order-dispatching and driver-routing problem is formalized as meal delivery routing problem (MDRP) in Reyes et al. (2018), which is a variant of dynamic pickup and delivery problem (DPDP) that received much attention for the past decades (for example, Psaraftis et al. (2016), Du et al. (2023)). In a recent study on MDRP, Yildiz and Savelsbergh (2019a) formulate the problem assuming that the platform has perfect information about order arrivals and solve it using a combined column and row generation approach. Ulmer et al. (2021) formulate the problem as a stochastic dynamic pickup and delivery problem using a route-based Markov Decision Process (MDP). An anticipatory customer assignment policy is proposed for order-dispatching and vehicle routing. Weng and Yu (2021) aim to improve the working conditions of drivers by order-dispatching algorithm design. A queuing-model-based algorithm is proposed with the objective of minimizing the waiting time of drivers while ensuring a good user experience. In addition, machine learning methods have been employed in food delivery order-dispatching and driver-routing problems, such as (Bozanta et al., 2022; Gao et al., 2021; Chen et al., 2022).

Delivery time prediction. Delivery time prediction is a variant of the ETA (Estimated Time of Arrival) problem (Wang et al., 2014, 2018, 2019). It estimates travel time between multiple points, which is often approximated by analytical functions (such as the ideas proposed by Beardwood et al. (1959), Wang and Odoni (2016), Chen and Wang (2018)), or predicted using machine learning and deep learning models. Compared with the traditional ETA problem, delivery time prediction in on-demand food delivery services is more challenging since the delivery time is endogenously affected by the demand and supply in the market.

Only limited work focuses on order delivery time prediction in food delivery. Liu et al. (2021) develop a data-driven framework that integrates travel time prediction and order-dispatching for a single restaurant. Hildebrandt and Ulmer (2021) propose two methods: an offline method that predicts order arrival time based on state features by means of gradient-boosted decision trees (GBDTs), and an offline-online method that exploits an offline supervised learning approximation with a deep neural network to perform detailed online simulations in real-time. Gao et al. (2021) discuss a time prediction module that simultaneously predicts order pickup time from the restaurants, driving time on the road, and delivery time to the customer's location. In the context of online retailing, Salari et al. (2022) develop a data-driven framework to predict the distribution of order delivery time and set the delivery time promised to customers using tree-based machine learning models.

In Table 1, we present an overview of the relevant literature that focuses on the area sizing optimization problem for same-day delivery services (SDD) and food delivery services (FD). The table delineates the specific characteristics and features explored in the papers: "Objective" specifies the objective function employed in the optimization model; "Decision" indicates the decision variables

Table 1
Literature on area sizing optimization problem in food and same-day delivery.

	Paper	Objective	Decision	Methodology	Approach	Instance
SDD	Ulmer et al. (2022).	Served orders	Service Radius	MDP	CA + VFA	×
	Banerjee et al. (2023).	Served orders	Service Radius	IP	Rolling horizon + CA	✓
FD	Yildiz and Savelsbergh (2019b).	Total profit	Service Radius	IP	AfQ + FOA	×
	Auad et al. (2020).	Served orders	Service Radius	IP	Solver	×
	Ding et al. (2020).	GMV	Candidate Service Scopes	IP	ML + Heuristics	✓
	Proposed **RASO-MT** Model.	Served orders	Service & Dispatch Radius	MIQCP	MNL + ML + Solver	✓

Note. RASO-MT: Restaurant Area Sizing Optimization with Model Tree; GMV: Global Merchandise Volume; MDP: Markov Decision Process; IP: Integer Programming; CA: Continuous Approximation; VFA: Value Function Approximate; AfQ: Approximation for Queueing; FOA: First-Order Algorithm; MNL: Multinomial Logit Model; ML: Machine Learning.

to be optimized; "Methodology" identifies the modeling method used; "Approach" identifies the primary approaches employed to solve the model; "Instance" denotes whether the experiments conducted in the respective papers employed real-world datasets as instances for evaluation.

In this paper, we present the first work to jointly optimize the customer service area and driver dispatch area, which requires demand estimation and order delivery time prediction. This is a new lever for balancing supply and demand in food delivery services.

3. Problem description

In this section, we present a formal description of the RASO problem for on-demand food delivery service, then present a simple example for illustration.

3.1. Problem statement

In the context of food delivery services, each restaurant is associated with a customer service area (CSA), denoted as A_c, which represents the geographical region within which customers can view the restaurant's information and place food orders. Each restaurant is also associated with a driver dispatch area (DDA), denoted as A_d, wherein drivers can access the restaurant's information and fulfill delivery orders. Both the CSA and DDA are determined by the platform and can be adjusted dynamically. To begin, we consider a food delivery system consisting of a cluster of restaurants $\mathcal{R}$, wherein a fleet of capacitated and homogeneous drivers $\mathcal{D} = \{d_1, d_2, \dots, d_m, \dots\}$ delivers a set of orders $\mathcal{O} = \{o_1, o_2, \dots, o_n, \dots\}$ that arrive starting from the initiation of the operating horizon $\mathcal{T}$.

Customer Orders: Each order $o \in \mathcal{O}$ can be described as a tuple (o^+, o^-, L^o), where o^+ denotes the pickup location (restaurant), o^- represents the delivery location (customer), and L^o indicates the promised delivery time (all orders have a guaranteed delivery time, e.g., 45 min). We assume that the order preparation time varies based on the restaurant and follows a Gamma distribution, a topic further discussed in Section 5.2. Customers have the option to cancel their orders if they fail to secure a driver within a specified time frame.

Drivers: The system comprises a total number of $\mathcal{D}$ homogeneous drivers. Each individual driver $d \in \mathcal{D}$ possesses a service capacity denoted as p (representing the maximum number of orders they can carry). Drivers' initial locations are influenced by multiple factors, which are beyond the scope of this model. In practice, areas with a high volume of orders tend to attract more drivers. Therefore, in the simulation and evaluation presented in Sections 5 and 6, we assume that drivers' initial locations correspond to the distribution of customer order locations.

Operating Horizon, CSA and DDA: The initiation of the operating horizon $\mathcal{T}$ is denoted by $t = 1$, and its conclusion, signifying the end of the operating, is indicated as $t = |T|$. At the commencement of each time period $t \in \mathcal{T}$, the platform is required to make three decisions: (1) determining the radius of the CSA, denoted as ρ_c, which directly influences the demand for each restaurant; (2) specifying the radius of the DDA, denoted as ρ_d, which significantly impacts the delivery supply for each restaurant; and (3) establishing the method for dispatching orders to available drivers for delivery. For each time period $t \in \mathcal{T}$, the platform has the option to establish the radii ρ_c and ρ_d and subsequently perform specific multiple order-dispatching in batches.

The entire decision-making process in the Restaurant Area Sizing Optimization (RASO) can be summarized as follows: At the initiation of each RASO decision time t, the platform first determines the radius ρ_c for the CSA, represented as A_c, and the radius ρ_d for the DDA, denoted as A_d. During time period t, at each order-dispatching decision time, the platform receives $\mathcal{O}_t$ orders within A_c for each restaurant and observes a set of drivers $\mathcal{D}_t$ within A_d for each restaurant. Subsequently, the platform makes decisions regarding the dispatch of orders to drivers. These dispatched decisions lead each driver $d \in \mathcal{D}$ to implement a delivery route RT_d, defined as a sequence of visiting locations, based on their ongoing carrying orders Ω_d and newly dispatched orders O_d. In the event that an order $o \in \mathcal{O}$ cannot find an available driver, it will be held until the next dispatching time unless it is cancelled by the customer. These decisions are made iteratively until the end time $|T|$ of the operating horizon $\mathcal{T}$. In order to describe the problem more clearly, all variables and parameters related to the restaurant will be added with the index r in Section 4 and Section 5.

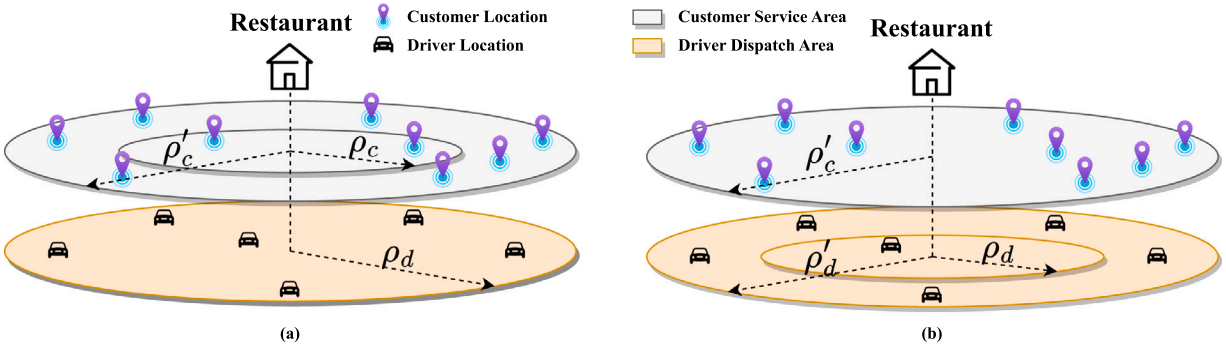

Fig. 4. Example of a restaurant's (a) customer service area and (b) driver dispatch area.

Table 2
Notation for the restaurant area sizing optimization problem.

Sets:	
$\mathcal{R}$	Set of Restaurants
$\mathcal{O}$	Set of orders
$\mathcal{D}$	Set of drivers
$\mathcal{T}$	Planning horizon; $\mathcal{T} = \{1, 2, \ldots, \lvert T\rvert\}$
Input parameters:	
L^o	Promised delivery time for order o
$\epsilon_{\max}$	A threshold predetermined as upper bound for average order delay
$\rho_{c,r}^{\min}, \rho_{c,r}^{\max}$	Minimum and maximum allowed radius of restaurant r's customer service area (CSA)
$\rho_{d,r}^{\min}, \rho_{d,r}^{\max}$	Minimum and maximum allowed radius of restaurant r's driver dispatch area (DDA)
Decision variables:	
$\rho_{c,r}$	Radius of customer service area (CSA) of restaurant r
$\rho_{d,r}$	Radius of driver dispatch area (DDA) of restaurant r
Intermediate variables:	
$O_r(\rho_{c,r})$	Function, the output is the number of prospective orders of restaurant r
$O_r(\rho_{c,r}, \rho_{d,r})$	Function, the output is the total number of served orders of restaurant r
$L_r(\rho_{c,r}, \rho_{d,r})$	Function, the output is the average delivery time for served orders of restaurant r

3.2. Example

The impact of the radius ρ_c of the CSA on demand, exemplified by the number of orders received by a restaurant, and the influence of the radius ρ_d of the DDA on supply, demonstrated by the number of available drivers for the restaurant, is illustrated in Fig. 4. In Fig. 4(a), when the CSA radius is ρ_c, only 2 orders are expected to arrive within the CSA (colored in gray), while 6 drivers are available within the DDA (colored in orange). The estimated demand is considerably lower than the available service capacity, suggesting a potential benefit in increasing the CSA radius to attract more orders for that restaurant, for instance, expanding it to ρ'_c to accommodate 9 expected orders. In Fig. 4(b), when the DDA radius is ρ_d, 9 orders are expected within the CSA, but only 1 driver is available within the DDA. The service capacity is significantly lower than the estimated demand, indicating the platform's advantage in increasing the DDA radius to attract more available drivers for that restaurant, for example, extending it to ρ'_d to include 6 drivers.

4. Solution method

This section presents the optimization framework for the RASO problem. We begin by introducing the main steps and the master optimization model in Section 4.1. Subsequently, we delve into customer order estimation on the demand side in Section 4.2, order-dispatching algorithms, and service operation in Section 4.3, and order delivery time prediction in Section 4.4. We present a specific Mixed Integer Quadratically Constrained Programming (MIQCP) model in Section 4.5, incorporating a model tree prediction for order delivery time. Lastly, we discuss a possible method to consider overlapping service areas for multiple restaurants in Section 4.6. Important notation is summarized in Table 2.

4.1. Framework and optimization model

We present a data-driven optimization framework encompassing customer demand estimation, order-dispatching, order delivery time prediction, and optimization of customer service and driver dispatch areas. The framework is depicted in Fig. 5.

First, delivery time prediction is depicted on the right. Based on real-world data concerning historical orders and deliveries, the key steps for delivery time prediction are as follows: (1) generating instances with various CSA and DDA radii, (2) extracting

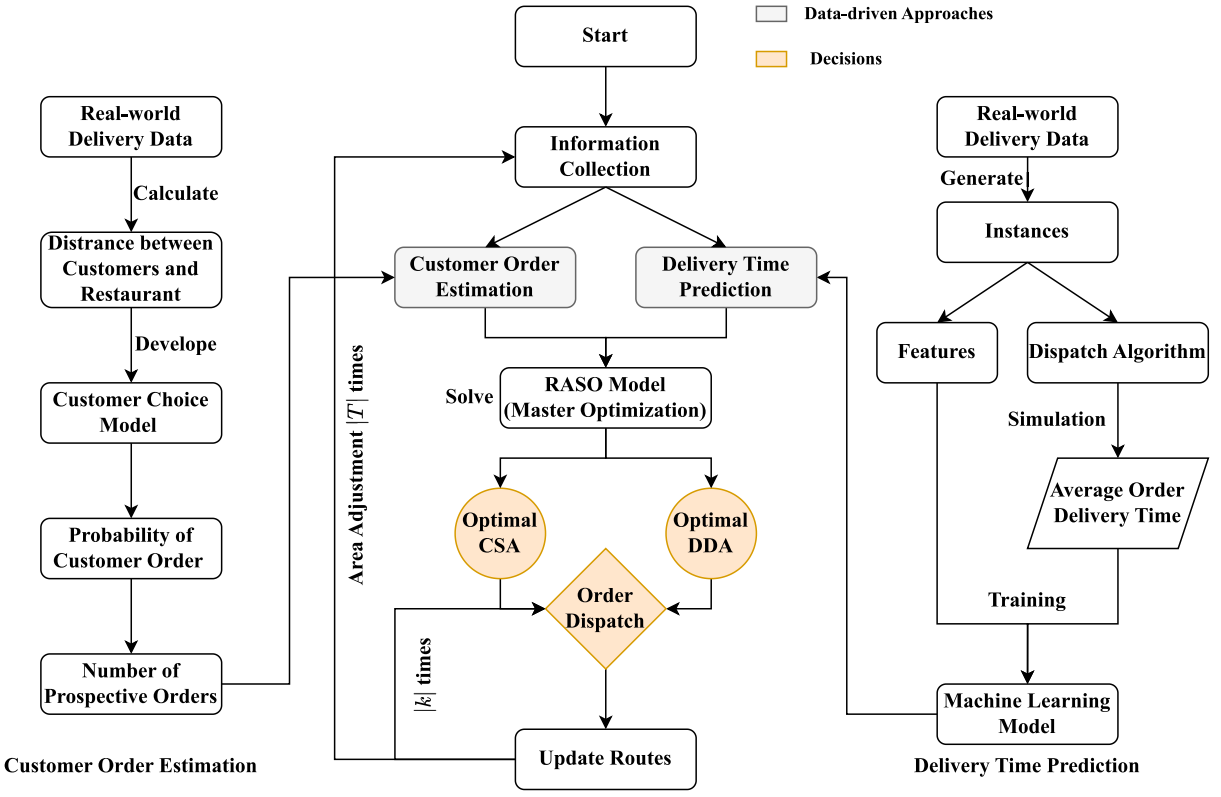

Fig. 5. Optimization framework for restaurant area sizing optimization problem.

relevant features from instances to predict delivery times, (3) simulating order deliveries using a predefined dispatch algorithm to obtain order delivery times as label data, and (4) training supervised machine learning models for delivery time prediction.

Next, customer order estimation is presented on the left. Initially, we calculate the distance between customers and restaurants, using it as an input feature to develop a customer choice model. This model allows us to investigate the relationship between customer order behavior and the distance to the restaurant. Subsequently, we compute the probability of a customer placing an order and estimate the total number of prospective orders for each restaurant.

The central illustration of the main optimization framework follows. At the beginning of time period t, we initiate the process with the collection of essential information, such as the locations of customers and drivers. Subsequently, we utilize our proposed data-driven approaches to develop two models: one for estimating customer order demand and the other for predicting delivery times. Through the integration of the estimation and prediction models with our proposed optimization model, we can effectively solve the RASO problem. Consequently, we can determine the optimal radii of the CSA and DDA for restaurants, thereby achieving a balance between supply and demand. The dispatch decisions result in updated delivery routes for each driver, with each driver having their designated route.

Specifically, we propose a master optimization formulation for the RASO problem. Our formulation focuses on a single time period, and it can be easily extended to multiple time periods using the rolling horizon methodology. The primary goal from the platform's perspective, as expressed in objective (1), is to maximize the number of served orders while guaranteeing a satisfactory level of order delivery time. More precisely, the platform aims to ensure that the order delivery delay for each restaurant $r \in \mathcal{R}$ in any time period $t \in \mathcal{T}$ remains below a predetermined threshold ϵ_{max} (which could be 0 by default), as stipulated in constraint (2). Here, we assume that all orders have the same promised delivery time. The selection of the CSA and DDA radii occurs within practically permissible ranges, as indicated by constraints (3)–(5).

$$[\textbf{RASO}] \quad \max_{\rho_{c,r},\rho_{d,r}} \quad O_r(\rho_{c,r},\rho_{d,r}), \tag{1}$$

$$\text{s.t.} \quad L_r(\rho_{c,r},\rho_{d,r}) - L^o \le \epsilon_{\max}, \tag{2}$$

$$\rho_{c,r}^{\min} \le \rho_{c,r} \le \rho_{c,r}^{\max}, \tag{3}$$

$$\rho_{d,r}^{\min} \le \rho_{d,r} \le \rho_{d,r}^{\max}, \tag{4}$$

$$\rho_{c,r},\rho_{d,r} \in \mathbb{R}^+. \tag{5}$$

In each time period $t \in \mathcal{T}$, the decision variables $\rho_{c,r}$ and $\rho_{d,r}$ represent the radii of the restaurant's CSA and DDA, respectively. The function $O_r(\rho_{c,r}, \rho_{d,r})$ quantifies the total number of successfully delivered orders from restaurant r and represents the count of served orders. The function $L_r(\rho_{c,r}, \rho_{d,r})$ calculates the average delivery time for the served orders from restaurant r. The primary challenges in solving the RASO problem are twofold: firstly, achieving accurate approximation of the objective function $O_r(\rho_{c,r}, \rho_{d,r})$ through a closed-form formula, and secondly, accurately approximating the delivery time $L_r(\rho_{c,r}, \rho_{d,r})$ and effectively integrating it into the optimization formulation. In reality, both $O_r(\rho_{c,r}, \rho_{d,r})$ and $L_r(\rho_{c,r}, \rho_{d,r})$ are challenging to estimate or express mathematically, as they depend on the complex interplay between supply and demand, and are undoubtedly influenced by decisions $\rho_{c,r}$ and $\rho_{d,r}$ simultaneously. To address these challenges, we introduce a data-driven approach to approximate function $O_r(\rho_{c,r}, \rho_{d,r})$ based on real-world data. Additionally, we utilize machine learning models to derive the optimal representation of function $L_r(\rho_{c,r}, \rho_{d,r})$. The details of our approach are elaborated in the subsequent subsections.

4.2. Customer order demand estimation

We begin by discussing the estimation of customer order demand. Customers are notably sensitive to the restaurant's proximity on the demand side. Longer distances from the restaurant tend to lead to extended delivery times, thereby reducing customers' willingness to place orders. The objective is to understand the relationship between the number of prospective orders $O_r(\rho_{c,r})$ and the radius of CSA $\rho_{c,r}$ while taking into account customers' order behavior. In particular, we employ a multinomial logit (MNL) model to investigate how the distance between a customer's location and the restaurant influences their order behavior.

Customer choice set: From a customer's perspective, they have three choices: to order from restaurant r, to order from any other restaurant, or not to order on the platform at all. To represent this customer choice, we introduce a binary decision variable $z_{i,r}$, which signifies the order behavior for each restaurant:

$$z_{i,r} = \begin{cases} 1, & \text{if customer } i \text{ place an order at restaurant } r, \\ 0, & \text{if customer } i \text{ does not place an order at restaurant } r. \end{cases} \tag{6}$$

The variable $U_{i,r}$ denotes the utility experienced by customer i when placing an order at restaurant r. In real-world scenarios, $U_{i,r}$ relies on several factors, including the restaurant's cuisine style, meal price, delivery fee, rating, customer preferences, and the distance between the customer and the restaurant. This paper assumes the following expression for $U_{i,r}$:

$$U_{i,r} = V_{i,r} + \boldsymbol{\beta}_r \cdot \mathbf{x}_{i,r} + \epsilon_{i,r}, \tag{7}$$

where $\mathbf{x}_{i,r} = (x_{i,r,1}, x_{i,r,2}, \dots, x_{i,r,|M|})$ represents the factors that relative to the customer restaurant selection, $\boldsymbol{\beta}_r = (\beta_{r,1}, \beta_{r,2}, \dots, \beta_{r,|M|})^T$ are the restaurant choice parameters, the dimension is $|M|$. The value $V_{i,r}$ is a constant representing the location utility[1] when $\mathbf{x}_{i,r} = 0$, while $\epsilon_{i,r}$ captures numerous unknown or unobservable factors treated as random variables following the Gumbel distribution. Based on the latest industry study conducted by Nextbite (2021), 77% of surveyed customers identified delivery time as the most crucial factor when selecting an online restaurant for delivery. The primary objective of this paper is to examine how a restaurant's CSA (radius) affects the volume of restaurant orders (demand). Generally, the distance between a customer and a restaurant plays a significant role in food delivery time and subsequently influences customer preferences for restaurant selection. To simplify the representation, we substitute the vector of factors $\boldsymbol{x}_{i,r}$ and $\boldsymbol{\beta}_r$ with variable $x_{i,r}$ representing the distance between the restaurant and the customer, along with a corresponding constant β_r, to model this influence. Intuitively, the farther the customer is from the restaurant and the longer the delivery time, the less inclined the customer will be to place an order (i.e., $\beta_r < 0$).

Then, the probability that customer i will place an order from restaurant r is:

$$\begin{aligned} \Pr(z_{i,r} = 1) &= \Pr(U_{i,r} > U_{i,r'}, \forall r' \neq r) \\ &= \frac{e^{V_{i,r} + \beta_r \cdot x_{i,r}}}{e^{V_{i,r} + \beta_r \cdot x_{i,r}} + \sum_{r'} e^{V_{i,r'} + \beta_r \cdot x_{i,r'}}}. \end{aligned} \tag{8}$$

The food delivery platform offers a vast number of restaurants, providing customers with a wide range of choices. Additionally, the utility $U_{i,r}$ is influenced by numerous unobservable variables, making it challenging to precisely determine its value. In such cases where explicitly computing the denominator for each individual and alternative combination is infeasible due to the large number of alternatives (Train, 2009), a large constant value can be employed to approximate the denominator. Consequently, we assume a reduced form of the probability as follows:

$$\Pr(z_{i,r} = 1) \approx \alpha_r \cdot e^{\beta_r \cdot x_{i,r}}, \tag{9}$$

where α_r and β_r are restaurant-specific parameters. In this context, the probability is considered in its reduced form when other factors, such as price and ratings, remain unobservable. With access to additional data, we could incorporate a greater number of variables into both the utility function and the probability function. Assuming that the number of customers per unit area (e.g., km^2) is n_r, the number of orders from customers at a distance x from restaurant r can be expressed as follows:

$$O_r(x) = 2\pi n_r \alpha_r e^{\beta_r x} x. \tag{10}$$

[1] The city of Singapore is divided into five districts. In our analysis, we make the assumption that restaurants within the same district share the same location utility $V_{i,r}$.

Table 3
Notation for order-dispatching.

Sets and Variables:	
l_d	Current location of driver d
Ω_d	Ongoing carrying orders by driver d
p_d^{remain}	Remaining capacity for driver d
$c_{o,d}$	Distance between order pickup location o^+ and driver's current location l_d
RT_d	Planned delivery route for driver d
$\mathcal{O}_{\text{pick}}$	Orders that have been dispatched and are being picked up by the drivers
$\mathcal{D}_{\text{avail}}$	Drivers with available capacity and in the restaurant's DDA

Finally, defining $\alpha_r' = 2\pi n_r \alpha_r$, the number of prospective orders that restaurant r can receive within its CSA (a certain radius $\rho_{c,r}$) can be given as:

$$\begin{aligned} O_r(\rho_{c,r}) &= \int P(z_{i,r}=1)\cdot n_r \cdot 2\pi x \mathrm{d}x \\ &= \int 2\pi n_r \alpha_r \cdot e^{\beta_r x} x \mathrm{d}x \\ &= \int \alpha_r' \cdot e^{\beta_r x} x \mathrm{d}x \end{aligned} \tag{11}$$

4.3. Order dispatching and service operation

While this study does not specifically focus on order-dispatching techniques, it acknowledges the significance of dispatching decisions throughout the entire service process. This subsection provides a brief description of the order-dispatching algorithm. The notation used in the order-dispatching algorithm is provided in Table 3.

The following dispatching rules and assumptions are applied:

Algorithm 1 Order-Dispatching Algorithm

Input: $\mathcal{R}$, $\mathcal{O}$, $\mathcal{D}$, Ω_d, $\rho_{d,r}$, p_d^{remain}
Output: RT_d'

procedure DISPATCH($\mathcal{O}, \mathcal{D}$)
  for $d \in \mathcal{D}$ **do**
    if $\Omega_d \neq \emptyset$ **then**
      Get driver d current location l_d and delivery location o^- for order $o \in \Omega_d$
      Re-schedule the planned route RT_d by solving a open TSP problem with unvisited locations for carried orders Ω_d
      Update the delivery route RT_d
  Initialize dispatched and being picked up orders $\mathcal{O}_{\text{pick}} = \emptyset$
  for $d \in \mathcal{D}$ **do**
    if d is going to pick up order $o \in \mathcal{O}$ **then**
      Update $\mathcal{O}_{\text{pick}} \leftarrow \mathcal{O}_{\text{pick}} \cup o$
  for $o \in \mathcal{O}$ **do**
    Get the order restaurant $r \in \mathcal{R}$
    if $o \in \mathcal{O}_{\text{pick}}$ **then**
      Continue
    Initialize the available drivers set $\mathcal{D}_{\text{avail}} = \emptyset$
    for $d \in \mathcal{D}$ **do**
      Calculate the distance $c_{o,d}$ between order o pickup location o^+ and driver d current location l_d
      Calculate the remaining capacity p_d^{remain} for driver d
      if $c_{o,d} \leq \rho_{d,r}$ and $p_d^{\text{remain}} \geq 1$ **then**
        $\mathcal{D}_{\text{avail}} \leftarrow \mathcal{D}_{\text{avail}} \cup d$
    Assign order o to driver $d \in \mathcal{D}_{\text{avail}}$ with maximum p_d^{remain}
    do greedy insertion
    Insert pickup node o^+ and delivery node o^- to route RT_d at positions with minimum distance increased
    Update $\mathsf{RT}_d \leftarrow \mathsf{RT}_d \cup (o^+, o^-)$
  return new delivery plan route RT_d' for $d \in \mathcal{D}$

- **Dispatch fairness**: If the number of orders carried by driver d is large/small (i.e., Ω_d is large/small), s/he will be less/more likely dispatched new orders. The dispatch fairness is centered on preventing the overload of certain drivers with a disproportionately large number of orders, while others remain underutilized. Specifically, our algorithm implements a priority rule that gives precedence to drivers carrying the lowest number of orders at any given time. This method ensures that new orders are more likely to be dispatched to these drivers, thereby promoting a more balanced distribution of work.

Table 4
Features for predicting average order delivery time.

Features	Definition
$\rho_{c,r}$	Radius of CSA of restaurant r
$\rho_{d,r}$	Radius of DDA of restaurant r
$D_r(\rho_{d,r})$	Number of available drivers in DDA of restaurant r
$O_r(\rho_{c,r})$	Number of prospective orders in CSA of restaurant r

- **Order re-dispatch**: Orders that have not been picked up within a specified time frame (e.g., 10 min) but have been assigned a driver can be re-dispatched to a new driver. This rule serves to enhance the flexibility of the dispatch algorithm by allowing for order re-dispatch, which has the potential to reduce order pickup time.
- **Driver routing behavior**: Following order dispatch, drivers deliver their carried orders by solving an open Traveling Salesman Problem (TSP) to optimize their route. In this rule, we assume that drivers always follow the shortest one-way delivery travel route from their current location to the restaurant and customer locations of assigned orders. This route is derived by solving an open TSP, where the driver does not return to the starting location.[2]
- **Order cancellation**: If a newly generated order cannot find an available driver within a certain time (e.g., 15 min), it is assumed that the customer becomes impatient and cancels the order. In other words, customers have the option to cancel orders if the platform fails to find a driver within a specific time period.

To summarize, we present the rules for order-dispatching and service operations using the pseudocode in Algorithm 1. Specifically, Lines 2 – 6 reconstruct each driver's delivery route based on the orders they are carrying. Lines 7 – 10 list the orders that have already been dispatched and are currently being picked up by the designated driver. Any remaining orders are dispatched to drivers in the DDA with the largest available capacity, as demonstrated in Lines 11 – 21. Additionally, Lines 22 – 24 implement a greedy insertion approach to incorporate new orders into the driver's current route with minimal increase in distance. Notably, the value of DDA radius $\rho_{d,r}$ significantly impacts the number of available drivers on the supply side, subsequently affecting order delivery times, as evident from Lines 19 and 20 in Algorithm 1. An increase in the DDA radius $\rho_{d,r}$ while keeping the restaurant's CSA unchanged results in more available drivers on the supply side, potentially reducing order delivery times. However, a larger DDA may also lead to longer pickup travel distances for orders, potentially increasing the overall delivery time. The estimation of order delivery time will be discussed in the subsequent subsection.

4.4. Order delivery time prediction

This subsection focuses on estimating the order delivery time. Specifically, given the radii $\rho_{c,r}$ and $\rho_{d,r}$ of the CSA and DDA for a restaurant r, our objective is to derive a mathematical representation of $L_r(\rho_{c,r}, \rho_{d,r})$ that effectively approximates the average order delivery time for served orders from restaurant r. To achieve this, we adopt a data-driven approach utilizing a customized simulator and dataset (as detailed in Section 5). By employing various supervised machine learning models, we aim to obtain closed-form formulas for order delivery time, which can then be integrated into the RASO formulation introduced in Section 4.1.

Feature Generation For each restaurant r, Table 4 presents the features considered for predicting the average order delivery time.

To ensure both interpretability and accuracy of the prediction model, we propose four simple features. The CSA radius ($\rho_{c,r}$) and DDA radius ($\rho_{d,r}$) for each restaurant r are considered. $D_r(\rho_{d,r})$ represents the platform's supply, where a larger DDA allows more available drivers for order dispatch, potentially reducing the order delivery time. On the other hand, $O_r(\rho_{c,r})$ represents the platform's demand, where a larger CSA allows more customers to place orders from the restaurant, which may increase the order delivery times.

4.5. A mixed integer quadratically constrained programming model

After conducting feature engineering and building prediction models, the subsequent step involves selecting an appropriate prediction model that aligns with the proposed RASO model. In their study, Liu et al. (2021) explored various machine learning models, encompassing both linear and non-linear (tree-based) models, for delivery time predictions. In this paper, we comprehensively evaluate several regression models, including ordinary least squares (OLS), ridge regression (Ridge), linear support vector regression (SVR), model tree (Wang and Witten, 1996), classification and regression trees (Loh, 2011), and XGBoost (Chen and Guestrin, 2016). The selection process and experimental details are presented in the following section, specifically in Section 5.4.

[2] It is important to note that in real-world scenarios, due to complexities such as left turns in intersections, the actual travel distance for order delivery tends to be equal to or greater than the travel distance derived from the open TSP solution.

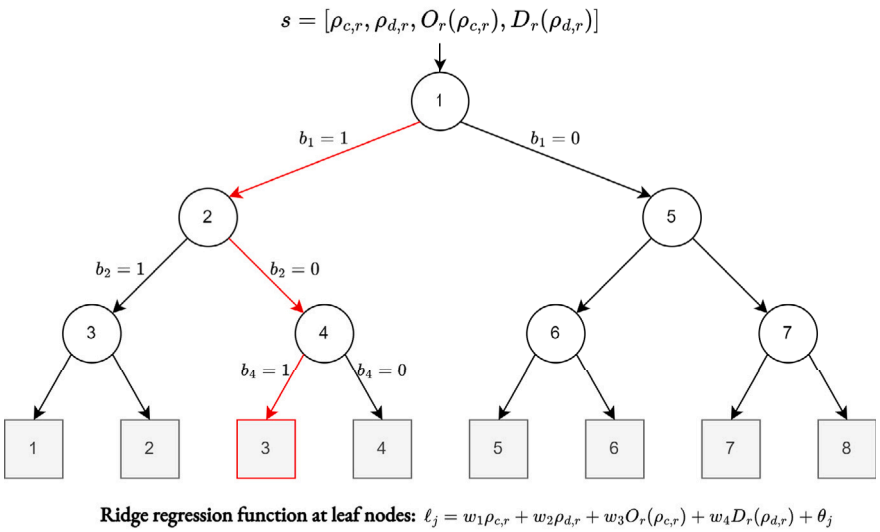

Fig. 6. Example of a model tree with ridge regression functions in the leaf nodes.

4.5.1. Model tree

In this subsection, we demonstrate how the RASO model proposed in Section 4.1 can integrate a trained machine learning model for delivery time prediction. We explore both linear and non-linear (tree-based) models and select one that exhibits both good prediction performance and optimization compatibility. Specifically, based on our numerical experiments with prediction models, detailed in Section 5, we choose the model tree as the prediction model. Unlike linear models, which assume linear functional relationships between ground truth labels and features (e.g., ordinary least squares and ridge regression), and some tree-based models that predict feature vector values through data feature splitting based on decision rules (e.g., classification and regression trees), the model tree combines linear regression models (e.g., ridge regression) for data feature splitting in a tree structure. Unlike classification and regression trees, which usually predict the mean or median value of labels in each leaf node, the model tree can fit any regression models in the leaf nodes. Formally, the model tree offers good interpretability and can capture both linear and non-linear relationships in the data, making it a suitable choice with a limited amount of training data (Quinlan et al., 1992; Frank et al., 1998; Gama, 2004; Potts and Sammut, 2005).

4.5.2. Linearization of model tree

Next, we demonstrate the integration of the model tree into the RASO formulation, resulting in a Mixed Integer Quadratically Constrained Programming (MIQCP) model. The updated model is efficiently solvable using commercial optimization solvers such as Gurobi and CPLEX within a reasonable timeframe.

Fig. 6 depicts a model tree with ridge regression functions assigned to its leaves. The tree's depth is 3, containing 7 branch nodes and 8 leaf nodes. The feature vector $\boldsymbol{s}$ characterizes the status based on decision variables. At each branch node, a decision rule based on the split feature's value is applied. For instance, in node 1 (the root), if the rule is true, the feature vector $\boldsymbol{s}$ is routed to the left; otherwise, it is routed to the right. Each leaf node corresponds to a ridge regression function ℓ_j, where j represents the index of leaf nodes. In this example, the feature vector $\boldsymbol{s}$ reaches leaf node 3 as the final destination.

Generally, a model tree is defined by (1) its tree structure, (2) decision rules at each branch node, and (3) linear regression functions assigned to each leaf node. Such a model tree can be represented as a Mixed Integer Linear Program (MILP). In the following sections, we introduce additional notation, which can be found in Table 5.

Following the decision rule at branch node $i \in \mathcal{B}$, we introduce the following set of constraints relevant to the delivery time predictions:

$$\boldsymbol{s}_k \geq v_{ik} + \mu - M b_i(\boldsymbol{s}), \tag{12}$$

$$\boldsymbol{s}_k \leq v_{ik} + M(1 - b_i(\boldsymbol{s})), \tag{13}$$

$$b_i(\boldsymbol{s}) \in \{0, 1\}, \tag{14}$$

where $b_i(\boldsymbol{s}) = 1$ indicates $\boldsymbol{s}_k$ less than v_{ik}, feature vector $\boldsymbol{s}$ will be routed to the left branch, and $b_i(\boldsymbol{s}) = 0$ indicates $\boldsymbol{s}_k$ greater than v_{ik} and will be routed to the right branch. To locate the leaf node j where feature vector $\boldsymbol{s}$ belongs, we introduce two sets, $\mathcal{B}_j^{\text{left}}(\boldsymbol{s})$ and $\mathcal{B}_j^{\text{right}}(\boldsymbol{s})$, and we have:

$$\mathcal{B} = \mathcal{B}_j^{\text{left}}(\boldsymbol{s}) \cup \mathcal{B}_j^{\text{right}}(\boldsymbol{s}), \quad \forall j \in \mathcal{F}. \tag{15}$$

Table 5
Notation for the mixed integer quadratic constrained programming (MIQCP) model.

Sets:	
$\mathcal{B}$:	branch nodes set;
$\mathcal{F}$:	leaf nodes set;
$\mathcal{B}_j^{\text{left}}(\boldsymbol{s})$:	for $\boldsymbol{s}$, set of branch nodes where the **left branch** following leaf node j;
$\mathcal{B}_j^{\text{right}}(\boldsymbol{s})$:	for $\boldsymbol{s}$, set of branch nodes where the **right branch** following leaf node j;
Input parameters:	
μ:	a small constant value;
M:	a large constant value;
$\boldsymbol{s}$:	a feature vector characterize the status based on decision variables;
ℓ_j:	linear function at leaf node j, $\forall j \in \mathcal{B}$;
$\boldsymbol{s}_k$:	value of feature k in vector $\boldsymbol{s}$;
v_{ik}:	value of feature k if it is selected for splitting at the branch node i;
Decision variables:	
$b_i(\boldsymbol{s})$:	binary variable, 1 if $\boldsymbol{s}$ branches left at node i, $\forall i \in \mathcal{B}$;
$c_j(\boldsymbol{s})$:	binary variable, 1 if $\boldsymbol{s}$ is located to the leaf node j, $\forall j \in \mathcal{B}$;
$v_j(\boldsymbol{s})$:	prediction value of $\boldsymbol{s}$ at leaf node j, $\forall j \in \mathcal{B}$.

Accordingly, we establish the following constraints for each leaf node $j \in \mathcal{F}$:

$$c_j(\boldsymbol{s}) \le b_i(\boldsymbol{s}), \quad \forall j \in \mathcal{F}, i \in \mathcal{B}_j^{\text{left}}, \tag{16}$$

$$c_j(\boldsymbol{s}) \le 1 - b_i(\boldsymbol{s}), \quad \forall j \in \mathcal{F}, i \in \mathcal{B}_j^{\text{right}}, \tag{17}$$

$$\sum_{j \in \mathcal{F}} c_j(\boldsymbol{s}) = 1, \tag{18}$$

where $c_j(\boldsymbol{s}) = 1$ indicates that feature vector $\boldsymbol{s}$ will be routed to leaf node j. Constraints (16)–(17) are responsible for enforcing the branch path, while constraint (18) ensures that each feature vector $\boldsymbol{s}$ can be routed to only one leaf node. Ultimately, the predicted value of the delivery time for feature vector $\boldsymbol{s}$ is given by:

$$v_j(\boldsymbol{s}) = \ell_j(\boldsymbol{s}) c_j(\boldsymbol{s}), \quad \forall j \in \mathcal{F}. \tag{19}$$

4.5.3. Revised RASO model

Finally, the revised RASO, with the integrated model tree predictor for delivery time called **RASO-MT**, is formulated as follows:

$$[\textbf{RASO} - \textbf{MT}] \quad \max_{\rho_{c,r}, \rho_{d,r}} \quad \sum_{r \in \mathcal{R}} O_r(\rho_{c,r}), \tag{20}$$

$$\text{s.t.} \quad \text{Constraints} \quad (3)\text{-}(5), (11)\text{-}(19),$$

$$D_r(\rho_{d,r}) = PWL(\rho_{d,r}), \quad \forall r \in \mathcal{R}, \tag{21}$$

$$\boldsymbol{s} = [\rho_{c,r}, \rho_{d,r}, O_r(\rho_{c,r}), D_r(\rho_{d,r})], \quad \forall r \in \mathcal{R}, \tag{22}$$

$$L_r(\rho_{c,r}, \rho_{d,r}) = \sum_{j \in \mathcal{F}} v_j(\boldsymbol{s}), \quad \forall r \in \mathcal{R}, \tag{23}$$

$$L_r(\rho_{c,r}, \rho_{d,r}) - L^o \le \epsilon_{\max}, \quad \forall r \in \mathcal{R}. \tag{24}$$

The objective function (20) is designed to maximize the total number of prospective orders $O_r(\rho_{c,r})$, which will be discussed in the following paragraph. Constraints (3)–(5) restrict the feasible domains of the decision variables, and constraints (11)–(19) show the integration of the order demand estimation and delivery time prediction with the optimization tool. Constraint (22) represents $\boldsymbol{s}$ as a feature vector. In the decision time, the platform can observe the drivers' current locations and compute the distance between the restaurant and those drivers. In real-world scenarios, the total number of drivers $D_r(\rho_{d,r})$ in the DDA for restaurant r with radius $\rho_{r,d}$ can be approximated using a piecewise linear function (Constraint (21)). This modeling approach is widely adopted in practice and is supported by popular commercial optimization solvers like Gurobi and CPLEX. Constraint (23) indicates that the average order delivery time equals the predicted value at the selected leaf node j. Constraint (24) is the delivery time constraint.

In the following, we provide a brief discussion on the prospective orders $O_r(\rho_{c,r})$ and served orders $O_r(\rho_{c,r}, \rho_{d,r})$. Suppose a customer places an order from a restaurant, and the platform accepts the order and proceeds to dispatch a driver for delivery. Each order can result in two outcomes: (1) if an available driver is nearby, the order is successfully delivered and classified as "served"; (2) if the platform fails to locate an available driver within a certain waiting time (e.g., 15 min), the customer cancels the order, and it is marked as "expired". In the extreme case where the DDA radius $\rho_{d,r} = 0$, all orders will expire as no drivers are available for the restaurant. From the perspective of the platform, since the aim is to maximize the number of orders served, in other words, they want each order can be delivered successfully within the promised delivery time and the number of expired orders to be as low as possible (ideally zero). We have observed that expired orders tend to occur when the service capacity is much smaller than the number of prospective orders. For instance, this can happen when the CSA radius is extremely large (e.g., 10 km) and the DDA radius is very small (e.g., 1 km). To address this issue, we impose a penalty during data preparation for training machine learning

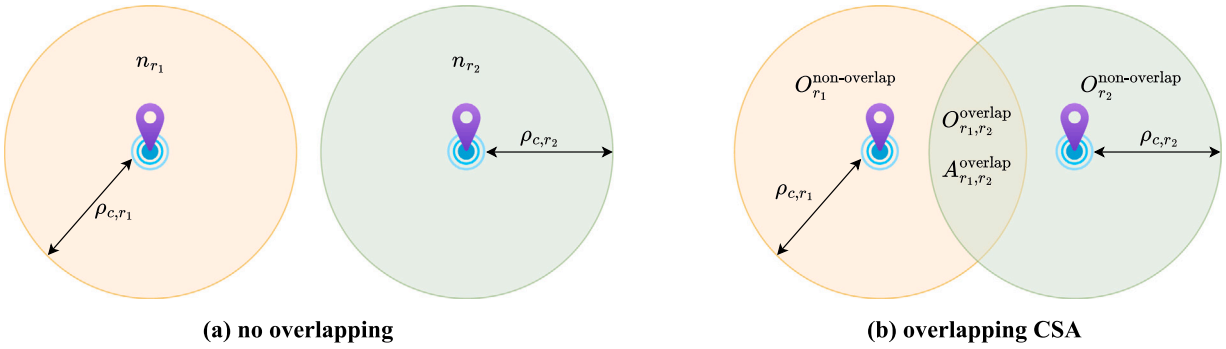

Fig. 7. An Illustrative example: with and without overlapping customer service areas (CSAs).

models used in delivery time prediction. This is achieved by setting a large delivery time Δt for expired orders. Consequently, in case of expired orders occur during the solution of the RASO problem, the corresponding predicted average order delivery time will not satisfy the time constraint (24). This approach ensures that when solving the RASO problem, the primary focus is on maximizing the total number of orders that can be successfully delivered, represented by the sum of $O_r(\rho_{c,r})$ for all restaurants in the set $\mathcal{R}$.

4.6. Overlapping customer service areas of restaurants

In the **RASO-MT** model, the objective function (20) is decomposable across restaurants due to the separate consideration of each restaurant $r \in \mathcal{R}$ in estimating the number of prospective orders $O_r(\rho_{c,r})$, as proposed in Section 4.2. However, particularly in areas of high demand such as Singapore, it is challenging to circumvent the scenario where two restaurants have overlapping customer service areas (CSAs). The presence of overlapping CSAs introduces competition between restaurants; customers within these areas may choose between multiple providers, potentially dividing the prospective orders in overlapping CSAs among the competing restaurants and compromising the accuracy of the $O_r(\rho_{c,r})$ estimates. Here, we present a simple example for illustration.

As illustrated in Fig. 7, two restaurants, r_1 and r_2, have CSAs with radii ρ_{c,r_1} and ρ_{c,r_2}, respectively. We assume the overlapping CSA is $A_{r_1,r_2}^{\text{overlap}}$, with the number of customers per unit area (km^2) for restaurant r_1 and r_2 being n_{r_1} and n_{r_2}, respectively.

Take restaurant r_1 as an example. In the absence of an overlapping CSA (i.e., $A_{r_1,r_2}^{\text{overlap}} = 0$), as shown in Fig. 7(a), customers within the CSA may choose to place orders exclusively with the restaurant. Consequently, we can approximate the value of n_{r_1} and employ Eqs. (10) and (11) to independently calculate the number of prospective orders for each restaurant. When there is an overlapping CSA (i.e., $A_{r_1,r_2}^{\text{overlap}} > 0$) as shown in Fig. 7(b), with customers shared between r_1 and r_2, the density of customers per square kilometer (km^2) in the overlapping CSA, $n_{r_1}^{\text{overlap}}$, is different to that in the non-overlapping CSA, $n_{r_1}^{\text{non-overlap}}$. To better estimate the number of prospective orders for restaurant r_1, it is necessary to calibrate the value of n_{r_1}. Consequently, we can introduce the following set of equations to more precisely approximate the values of n_{r_1}, taking into account the overlapping CSA with restaurant r_2:

$$\begin{cases} Q_{r_1}(\rho_{c,r_1}) = Q_{r_1}^{\text{non-overlap}} + Q_{r_1,r_2}^{\text{overlap}} \\ A_{r_1} = A_{r_1,r_2}^{\text{overlap}} + A_{r_1}^{\text{non-overlap}} \\ n_{r_1}^{\text{overlap}} = \dfrac{Q_{r_1,r_2}^{\text{overlap}} \cdot f_{r_1}}{A_{r_1,r_2}^{\text{overlap}}} \\ n_{r_1}^{\text{non-overlap}} = \dfrac{Q_{r_1}^{\text{non-overlap}}}{A_{r_1}^{\text{non-overlap}}} \\ n_{r_1} = \dfrac{Q_{r_1,r_2}^{\text{overlap}} \cdot f_{r_1} + Q_{r_1}^{\text{non-overlap}}}{\pi \rho_{c,r_1}^2} \\ 0 \le f_{r_1} \le 1 \end{cases}, \tag{25}$$

where f_{r_1} represents the proportion of orders placed to restaurant r_1 in the overlapping area. Similarly, we can get a set of equations for restaurant r_2 with a proportion f_{r_2}. For the overlapping areas shared by restaurants r_1 and r_2, we have $f_{r_1} + f_{r_2} = 1$. In practice, the value of f_{r_1} and f_{r_2} can be approximated using historical customer ordering data. We can generalize this to multiple restaurant scenarios and establish a set of equations for each restaurant with overlapping CSAs.

In our case study, the customer order density, n_r, can be inferred from historical customer ordering data, along with the parameter α_r, as demonstrated in Eq. (11). Since the historical data already indicates an adjustment for the presence of overlapping CSAs, the estimation of $\alpha' = 2\pi n_r \alpha_r$ from historical customer ordering data reduces the need to incorporate constraints (25) in the **RASO-MT** model. Moreover, a recent study by Meituan-Dianping (Ding et al., 2020), which aims to optimize delivery scopes similarly to CSAs in

our study, treats each restaurant as an independent entity without considering its overlapping CSAs. This provides partial justification that the prediction of total prospective orders for each restaurant independently from historical data is a good approximation, from the perspective of an industry platform. It supports using the sum of orders $O_r(\rho_{c,r})$ for each restaurant $r \in \mathcal{R}$ as the objective function within our RASO framework.

5. Simulator, dataset, and prediction models

In this section, we first introduce the simulator we developed to simulate the process of food delivery services in Section 5.1, including decisions on the radii of CSA and DDA for restaurants, order generation, dispatching, pickup, and delivery. Then in Section 5.2, we present a real-world delivery dataset from a crowd-sourcing food delivery platform. The performance of customer demand estimation is discussed in Section 5.3, followed by a discussion on the performance of order delivery time prediction models in Section 5.4.

5.1. Simulator for on-demand food delivery service

To better simulate the real-world environment in food delivery service and also to generate training samples for order delivery time prediction depicted in Fig. 5, we develop a simulator that is calibrated using real data. The simulator contains several components: service area decisions, dispatch area decisions, order generation, information collection, order-dispatching algorithm, and status updates of orders and drivers. In Algorithm 2, we show the entire simulation procedure. As can be seen, we solve the RASO problem for each time period $t \in \mathcal{T}$ to get the optimal radius for CSA and DDA for each restaurant, then for each order-dispatching interval $k \in [t, t+1)$, we proceed the following steps: (1) generating new order requests for each restaurant within the CSA; (2) gathering information about all dispatched and newly generated orders, drivers, and restaurants, such as each order's pickup and delivery locations, driver's location, driver's carried orders, restaurant dispatch area, and so on; (3) implementing the order-dispatching algorithm to dispatch orders to drivers who have available capacity in the DDA; (4) updating orders' status for orders that have been successfully dispatched, picked up and delivered, as well as for orders that have been canceled by customers; and (5) updating the drivers' planned delivery routes based on newly dispatched orders.

Algorithm 2 Simulation

Input: information of orders $\mathcal{O}$, drivers $\mathcal{D}$ and restaurants $\mathcal{R}$

procedure SIMULATION($\mathcal{O}, \mathcal{D}, \mathcal{R}$)
  for each area optimization time period $t \in \mathcal{T}$ **do**
    Area optimization: Solve the RASO model and get the optimal customer service area $A_{c,r}$ and driver dispatch area $A_{d,r}$.
    for each order dispatch time interval $k \in [t, t+1)$ **do**
      Order generation: generate new order requests for each restaurant according to the orders arrival rate within the customer service area $A_{c,r}$
      Information collection: collect information of all dispatched and newly generated orders, drivers, and restaurants;
      Order dispatch: dispatch remaining and newly generated orders to drivers in the driver dispatch area who have available capacity using Algorithm 1;
      Update orders' status: update the status of orders that have been successfully dispatched, picked up and delivered, also for orders that have been canceled by customers;
      Update drivers' status: update drivers' planned delivery route based on newly dispatched orders;
    end for
  end for

We also account for the random ready time for orders in our simulations. In food delivery services, the preparation time (which includes cooking the meal and making it ready for pickup) for restaurants is often highly uncertain. Accurately estimating the food preparation time for each order can yield significant benefits, both for delivery drivers and the customers' experience. In this study, we make the assumption that the food preparation time for each restaurant follows distinct gamma distributions, characterized by different shape parameters α and a scalar parameter β. This choice is motivated by the common use of gamma distributions for modeling waiting times in on-demand food delivery platforms (Ulmer et al., 2021; Gao et al., 2022).

5.2. Real-world dataset

The dataset used in the numerical experiments is provided by a crowd-sourcing food delivery platform in Singapore. The data contains a sample of around 80,000 order records for over 2,000 restaurants and over 30,000 customers over 8 months (October 2020 to May 2021). Each delivery record includes order and driver information, including order pickup and delivery locations, delivery distance, order accept time, driver ID, and fee paid. This study focuses on a cluster of restaurants known as "hawker centre" (or "food centre"), which represents an open-air complex commonly found in Singapore. The "hawker centre" is a popular dining destination featuring various food stalls or vendors, offering a wide variety of affordable and delicious local dishes. Each restaurant in the same hawker center shares the same radii for CSA and DDA (ρ_c, ρ_d). Fig. 8 depicts customer locations (i.e, order delivery locations) from

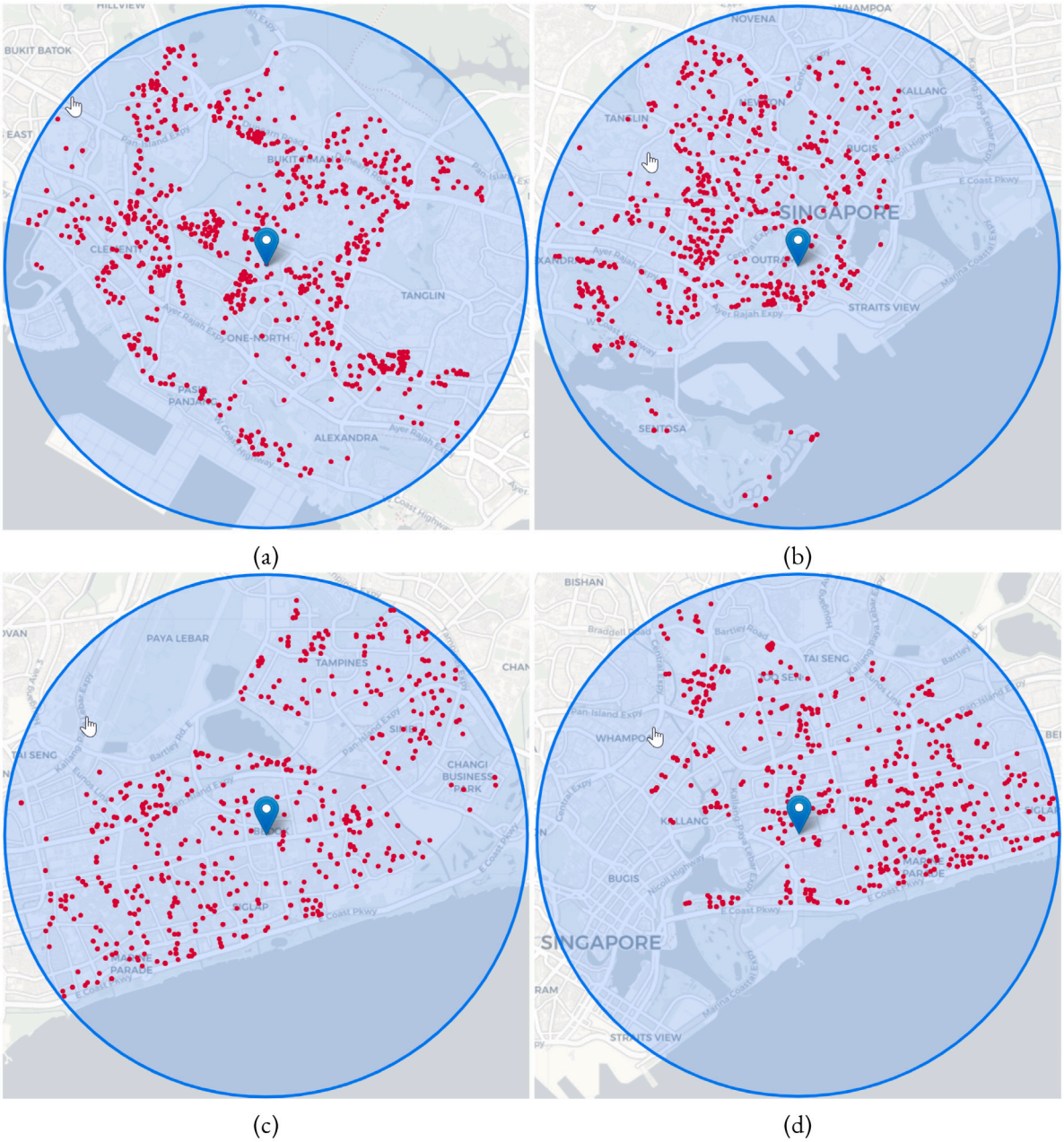

Fig. 8. Orders delivery locations of selected hawker centres: (a) Ghim Moh Market & Food Centre, (b) Maxwell Food Centre, (c) Bedok Interchange Hawker Centre, and (d) Old Airport Road Food Centre. (Blue circle represents the CSA)

four well-known hawker centres[3] in Singapore, one located in the central business district, and the others in the residential area. Customer locations are depicted by red dots and hawker centre locations are depicted by blue markers. After cleaning the data and removing some missing values, we choose the top nine restaurants based on the number of orders in the dataset for further investigation. Fig. 9 is a histogram that groups the times customers placed orders into 45-minute intervals, illustrating the average number of orders across time periods throughout the day, ranging from 6:00 to 21:45, for the four selected restaurants. The x-axis represents the time period; and y-axis shows the (re-scaled) average number of orders. Notably, the restaurants exhibit substantial

[3] Due to space limitations, we will concentrate on presenting the results of four representative hawker centres in the main body of the paper. The remaining five hawker centres are included in the appendix for reference to ensure the paper's readability.

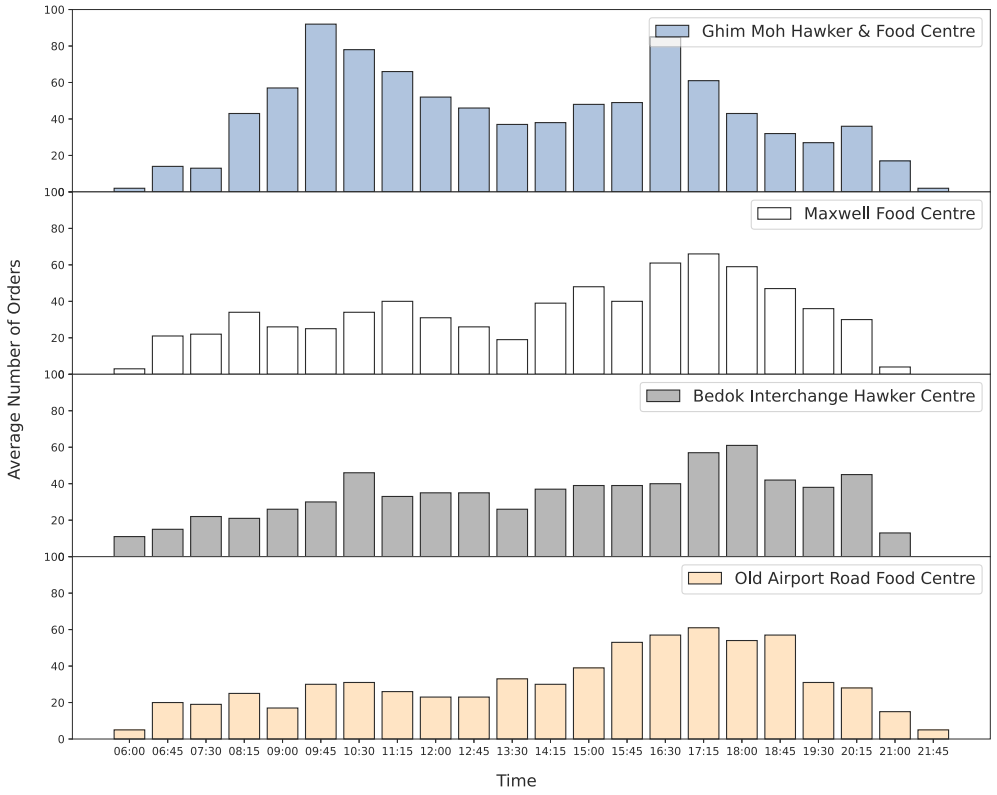

Fig. 9. (Re-scaled) Average number of orders for restaurants over time (Every 45 min).

peaks during the noon hours (9:45 to 12:00) and the evening hours (16:30 to 18:00). The distributions of order preparation time are visualized in Fig. 10.

5.3. Results on customer demand estimation

Using the real-world order data, we conduct numerical experiments to evaluate the proposed estimation of customer order demand based on the discrete choice model proposed in Section 4.2. Fig. 11 displays histograms representing the total number of customer orders in relation to the distance between the restaurant and the customer for selected hawker centres. The x-axis indicates the customer's distance (in kilometers) from the restaurant, while the y-axis represents the number of orders placed for each restaurant. It shows that as the distance from the restaurant increases, the number of orders exhibits a steep initial increase, followed by a gradual decrease. This observation aligns with the number of orders described by the expression $\alpha_r' e^{\beta_r x} x = 2\pi n_r \alpha_r e^{\beta_r x} x$, which is dependent on the distance x from the restaurant, as indicated in Eq. (10). Notably, a greater distance from the restaurant leads to an increase in the marginal service area (represented by $2\pi x n_r$). However, it also results in a reduction in customers' willingness to place orders (demonstrated by $\alpha_r e^{\beta_r x}$ with a negative β_r). To estimate the prospective orders approximation function (11) for each restaurant, non-linear least squares is employed. The coefficient of determination (R^2) is utilized as a performance metric to determine the values of α_r' and β_r for each restaurant:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i, \quad SS_{\text{res}} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad SS_{\text{tot}} = \sum_{i=1}^{n}(y_i - \bar{y})^2, \quad R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}, \tag{26}$$

where n is the total number of estimated values, and y_i and $\hat{y}_i$ are the actual value and its estimate, respectively. If the fitted value is exactly the same as the actual value, $R^2 = 1$. The fitting function of each restaurant and its R^2 value are displayed at the top right in each subfigure. We observe that the estimation works well for all hawker centres, with R^2 value ranging from 0.827 (Ghim Moh Market & Food Centre) to 0.892 (Maxwell Food Centre). This demonstrates the effectiveness of the customers' choice model for estimating the number of prospective orders with different service radii. In the appendix, we show the estimation performance for all nine hawker centres.

As comparisons, we also explored the application of other machine learning models to predict customer demand. Specifically, we refine our dataset to distill four core features for the customer demand estimation: (1) Time of Day: inferred from order timestamps, reflects the daily patterns that impact how customers place orders. (2) Day of the Week: segments the data into weekdays and

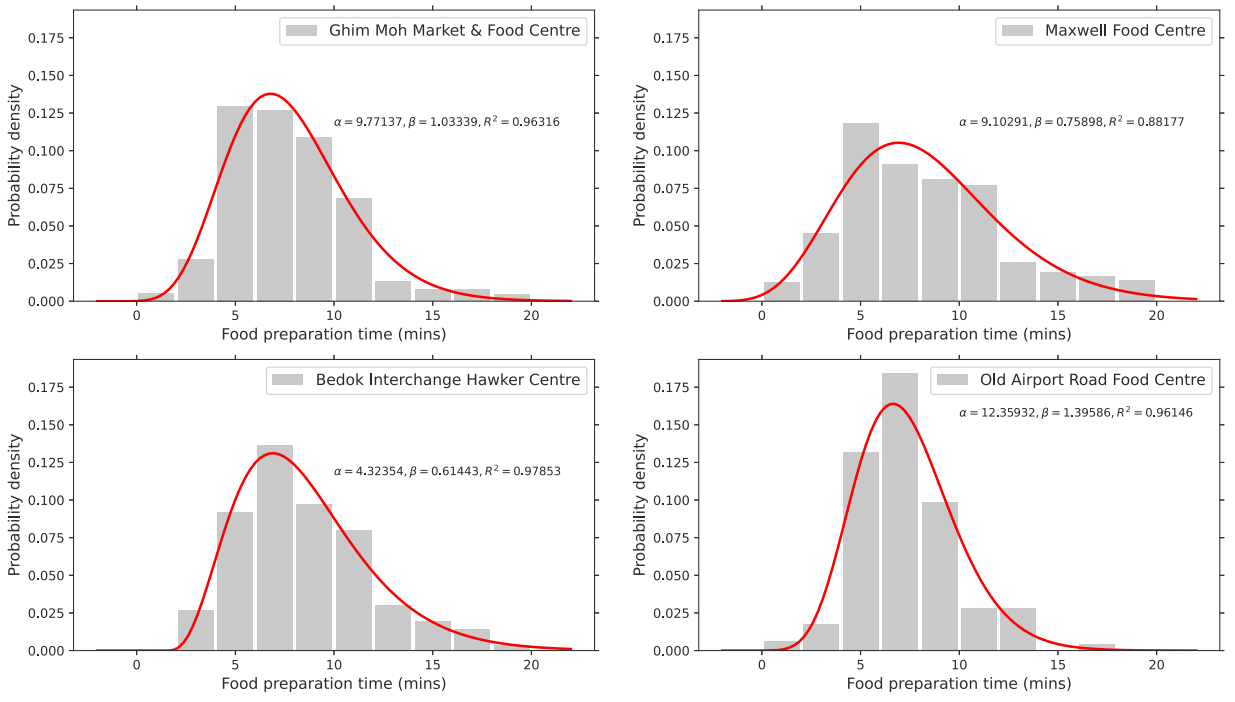

Fig. 10. Histogram of restaurants' orders preparation time. The red curve represents a gamma distribution function derived through non-linear least squares regression.

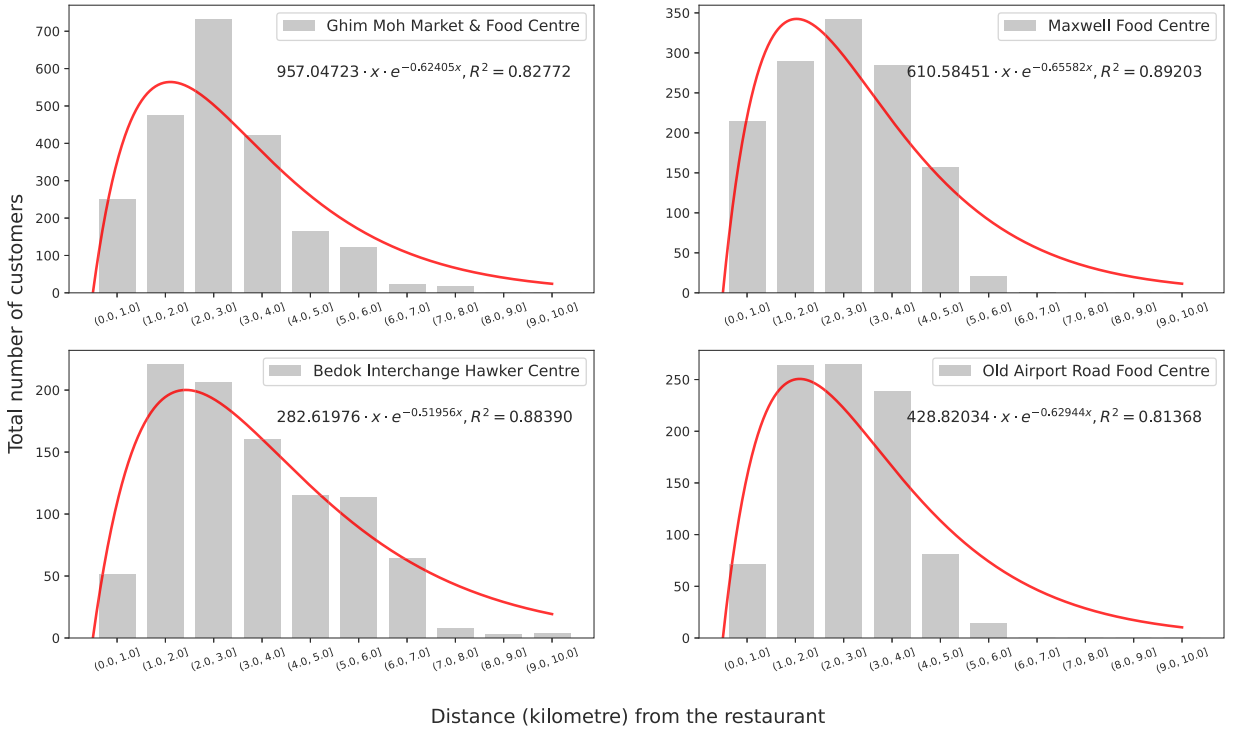

Fig. 11. Histogram of customers orders in terms of customer-restaurant distance. The red curve represents the function derived through non-linear least squares regression.

Table 6
Performance comparison of ML methods and customer choice model on customer order estimation at Ghim Moh Market & Food Centre.

Model	MAE	RMSE	R^2
Ridge Regression	9.707	154.69	0.425
Decision Tree	6.398	100.317	0.759
Random Forest	6.351	98.255	0.764
XGBoost	2.137	9.399	0.970
Customer Choice Model	3.424	52.623	0.828

Table 7
Details of experimental instances.

Instance	Hawker Centre	Area	Orders	Drivers	α_r'	β_r
1	Ghim Moh Market & Food Centre	Residential	[50, 80]	[20, 30]	957.04623	−0.62405
2	Maxwell Food Centre	Central	[20, 50]	[10, 20]	610.58451	−0.65582
3	Bedok Interchange Hawker Centre	Residential	[15, 35]	[5, 15]	282.61976	−0.51956
4	Old Airport Road Food Centre	Residential	[20, 40]	[5, 15]	428.82034	−0.62944
5	Kovan Food Centre	Residential	[60, 90]	[20, 30]	217.19399	−0.44593
6	Bukit Merah View Market & Food Centre	Central	[30, 60]	[10, 20]	935.43501	−0.77316
7	Hong Lim Food Centre	Central	[25, 45]	[5, 15]	499.51778	−0.6803
8	Bukit Timah Market & Food Centre	Residential	[80, 105]	[25, 35]	185.56244	−0.39559
9	Alexandra Village Food Centre	Residential	[35, 55]	[10, 20]	252.13369	−0.52519

weekends, thereby capturing fluctuations in weekly demand. (3) Restaurant's Customer Service Area: determined by the radius $\rho_{c,r}$, quantifies a restaurant's potential customer base. (4) Delivery Distance: calculated from the distance between the restaurant and delivery locations, provides insights into logistical efficiency and customer proximity. The results of the machine learning (ML) methods' performance and our customer choice model in estimating customer orders are presented in Table 6.

Our experimental results demonstrate that XGBoost outperforms other models across all metrics, such as MSE, RMSE, and the R^2 score, thereby indicating its superior predictive performance. Furthermore, tree-based ML models exhibit robust predictive capabilities for customer orders. Notably, our customer choice model attains lower MSE and RMSE values, in addition to a higher R^2 score, when compared to both Decision Tree and Random Forest models. Nonetheless, although XGBoost demonstrates remarkable performance, translating its complex, non-linear predictions into a closed-form expression for seamless integration with our optimization model is a significant challenge. Given these considerations, we decide to continue with the proposed customer choice model for demand estimation. Our decision is based on balancing predictive accuracy and compatibility with the optimization model, particularly focusing on the requirement for a closed-form demand function that can be integrated seamlessly into the optimization framework. We contend that the customer choice model, despite its inherent simplifications, represents a viable compromise, offering an adequately precise depiction of customer demand and ensuring compatibility with our optimization strategy.

5.4. Model selection for order delivery time prediction

For order delivery time prediction, we evaluate six machine learning models: ordinary least squares (OLS), ridge regression (Ridge), linear support vector regression (SVR), model tree, regression tree, and XGBoost (Gradient Boosting Decision Tree). We implement 5-fold cross-validation to select the best parameters (e.g., coefficient value for the regulation term, maximum depth of the tree in tree-based models) for all models. In the experiment, nine instances were generated for different restaurants (with different α_r' and β_r) and different numbers of orders and drivers. The composition of training instances was designed to cover a broad spectrum of scenarios encountered in real-world applications, which ensures that the model is exposed to diverse data patterns. As shown in Fig. 12, nine representative hawker centres were selected for the case study, with some located in city central areas and others in residential areas. The hourly orders placed at these hawker centres ranged from low (15 to 35 orders) to high (80 to 100 orders). The detailed information on instances is shown in Table 7. We simulate deliveries for each restaurant from 11:00 to 12:00—which generates over 8,000 data points for each instance—with 80% as the training and validation set and 20% as the test set. We fine-tuned the parameters of the machine learning models used to optimize performance according to the metrics. For the calibration of coefficients in Ridge Regression, we optimized the 'alpha' parameter (representing the regularization strength) through cross-validation to achieve a balance between model complexity and prediction accuracy. For Linear SVR, we concentrated on adjusting the 'C' and 'max_iter' parameters to enhance the model's performance, utilizing grid search to determine the optimal values. For both the Regression Tree and Model Tree, we modified the 'max_depth', 'min_samples_split', and 'min_samples_leaf' parameters to mitigate overfitting and ensure the model's capacity to discern underlying data patterns. The 'Learning_rate' for XGBoost was set at 0.3. The configurations of the parameters are summarized in Table 8. All training, validation, and testing were conducted using Python 3.9, leveraging the scikit-learn (Pedregosa et al., 2011) and XGBoost packages (Chen and Guestrin, 2016).

We use the mean absolute percentage error (MAPE) and R^2 score as performance metrics for the prediction models:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\%. \tag{27}$$

Fig. 12. Selected hawker centres locations in Singapore.

Table 8
Parameter configurations for ML methods.

Model	Parameter	Value
Ridge Regression	Alpha	0.5
Linear SVR	C	1,000
Linear SVR	Max_iter	10,000
Regression/Model Tree	Max_depth	5
Regression/Model Tree	Min_samples_split	2
Regression/Model Tree	Min_samples_leaf	10
XGBoost	Learning_rate	0.3

The performance of the machine learning models in predicting average order delivery time is shown in Table 9. All tree-based models exhibit strong predictive accuracy for average delivery time, as indicated by their R^2 scores. Conversely, the linear models (i.e., OLS, Ridge, and SVR) exhibit significantly lower R^2 scores and higher MAPE values when compared to the tree-based models (i.e., model tree, regression tree, and XGBoost). Particularly noteworthy is the model tree, which consistently outperforms all other models, achieving the lowest MAPE across all instances as well as the highest R^2 score, surpassing XGBoost, linear models (OLS, Ridge, and SVR), and the regression tree. Considering prediction accuracy, compatibility, and computational performance, we opt for the model tree as the delivery time prediction model and seamlessly integrate it into the RASO-MT model using the linearization method introduced in Section 4.5.2.

6. Experiments and discussion

In this section, we conduct a set of numerical experiments to evaluate the performance of the proposed **RASO-MT** model comparing it with several benchmarks in practice and discuss its computational scalability. Then, we discuss the impact of some important factors and variables on the performance of food delivery services, such as the threshold of order delay and temporal distribution of customer order demand. We run the experiments in Python 3.9 using Gurobi 11.0.0, on a 2.5 GHz Xeon CPU.

6.1. Model performance

We report the performance of the proposed **RASO-MT** model in terms of the following four metrics: (1) growth rate of orders served; (2) average delivery time of served orders; (3) average travel distance per driver, defined as the total distance traveled to deliver their dispatched orders; and (4) on-time rate, which is the ratio of orders delivered in promised time to total orders arriving within a specific period of time. We use **Fixed-CSA-DDA** as a baseline, where both the restaurant's CSA and DDA are fixed, with radii set to 5 km (commonly used choices made by practitioners). For convenience, the index r is omitted for the intermediate variables and decision variables defined in Table 2 when it is clear.

Table 9
Performance of ML methods on average delivery time prediction.

(a) Ghim Moh Market & Food Centre

Model	R^2	MAPE
Ordinal Least Square	0.578	0.567
Ridge Regression	0.577	0.566
Linear SVR	0.344	0.325
Regression Tree	0.900	0.161
Model Tree	**0.912**	**0.147**
XGBoost	0.893	0.176

(b) Maxwell Food Centre

Model	R^2	MAPE
Ordinal Least Square	0.483	0.659
Ridge Regression	0.483	0.658
Linear SVR	0.198	0.351
Regression Tree	0.883	0.199
Model Tree	**0.888**	**0.176**
XGBoost	0.848	0.248

(c) Bedok Interchange Hawker Centre

Model	R^2	MAPE
Ordinal Least Square	0.485	0.626
Ridge Regression	0.485	0.625
Linear SVR	0.251	0.345
Regression Tree	0.914	0.167
Model Tree	**0.922**	**0.154**
XGBoost	0.882	0.212

(d) Old Airport Road Food Centre

Model	R^2	MAPE
Ordinal Least Square	0.489	0.699
Ridge Regression	0.489	0.698
Linear SVR	0.275	0.393
Regression Tree	0.903	0.170
Model Tree	**0.910**	**0.158**
XGBoost	0.834	0.264

Table 10
Growth rate of served orders for selected hawker centres.

Hawker Centre	Ghim Moh	Maxwell	Bedok interchange	Old Airport Road
Growth Rate	20.5%	19.0%	29.6%	19.4%

We conducted experiments during lunchtime (11:00 to 12:00), varying the number of drivers in the system from small to medium and large scales, with each increment involving an addition of five drivers. The range of the number of drivers for each hawker center is shown in Table 7. Each driver is assumed to have a service capacity of 10 orders ($p = 10$). Moreover, we set the promised delivery time for orders (L^o) to 50 min with the threshold for order delay, $\epsilon_{\max} = 0$, which is commensurate with common customer expectations (e.g., Liu et al. (2021)). We recorded the metric for results before and after the deployment of the **RASO-MT** as $\mathbb{M}_{\text{Fix}}$ and $\mathbb{M}_{\text{RASO-MT}}$, respectively, which were used to verify the effectiveness of our model. Then we define the $\mathbb{M}_{\text{Rat}}$ as the changing ratio between $\mathbb{M}_{\text{Fix}}$ and $\mathbb{M}_{\text{RASO-MT}}$ as follows:

$$\mathbb{M}_{\text{Rat}} = \frac{\mathbb{M}_{\text{RASO-MT}} - \mathbb{M}_{\text{Fix}}}{\mathbb{M}_{\text{Fix}}} \tag{28}$$

The results are presented in Table 10, demonstrating that the **RASO-MT** model's newly introduced restaurant's CSA and DDA lead to significant improvements in the number of served orders, ranging from 19.0% to 29.6% for the selected four prominent hawker centres when compared to the conventional fixed CSA and DDA approach. Detailed experimental outcomes are depicted in Fig. 13. Specifically, the average travel distance of drivers increased by 16.5% to 20.3%, 11.1% to 26.0%, 15.1% to 27.0%, and 5.6% to 13.7% for the four hawker centres, respectively. It is also evident that there was a decline in delivery efficiency to a certain degree. The average order delivery time experienced increases of 2.4% to 6.3%, 6.0% to 9.7%, 4.3% to 27.7%, and 4.2% to 15.3% for the corresponding centres. These findings indicate that the proposed method demonstrates superior performance with a sufficient number of drivers (e.g., large instance), as the corresponding rise in the average order delivery time remains comparatively modest. As the number of orders served grows, the on-time delivery rate is observed to decrease, with reductions of up to 2.5%, 7.6%, 6.9%, and 7.0%, respectively (Fig. 13(c)). Full details of the evaluation results are provided in Table A.2 in Appendix.

To comprehensively evaluate our proposed area sizing optimization strategy, we also assess its performance against additional benchmark policies for longer service times, ranging from 09:00 to 15:00. The two new benchmarks are:

- **Fixed-CSA**: The restaurant's CSA is fixed with a radius of 5 km, while the DDA is optimized.
- **Fixed-DDA**: The restaurant's DDA is fixed with a radius of 5 km, while the CSA is optimized.[4]

We have selected Ghim Moh Market & Food Centre, with learned parameters $\alpha_r' = 957.04723$ and $\beta_r = -0.62405$ to conduct a comprehensive and in-depth analysis. We also analyze the effects of different parameter settings, including the threshold of delay and order arrival rate. Additionally, we assume the number of drivers in the system equals 15 for all time periods $t \in \mathcal{T}$, and these drivers are randomly distributed in the area. The maximum and minimum radius for both restaurants' CSA ($\rho_c^{\max}$, $\rho_c^{\min}$) (the index r is omitted when it is clear) and DDA ($\rho_d^{\max}$, $\rho_d^{\min}$) are set to be 2 km and 10 km, respectively.

Table 11 shows the number of orders served and the average order delivery time for the proposed **RASO-MT** model compared to other benchmark policies. To compare the policies, we calculate the relative improvement in terms of the number of orders served

[4] The radius for a restaurant's CSA is set to around 4 km in Ding et al. (2020). However, in our paper, we set the fixed radius of restaurant's CSA or DDA slightly higher, to 5 km, to account for the higher maximum delivery distance of 10 km in our dataset compared to Ding et al. (2020).

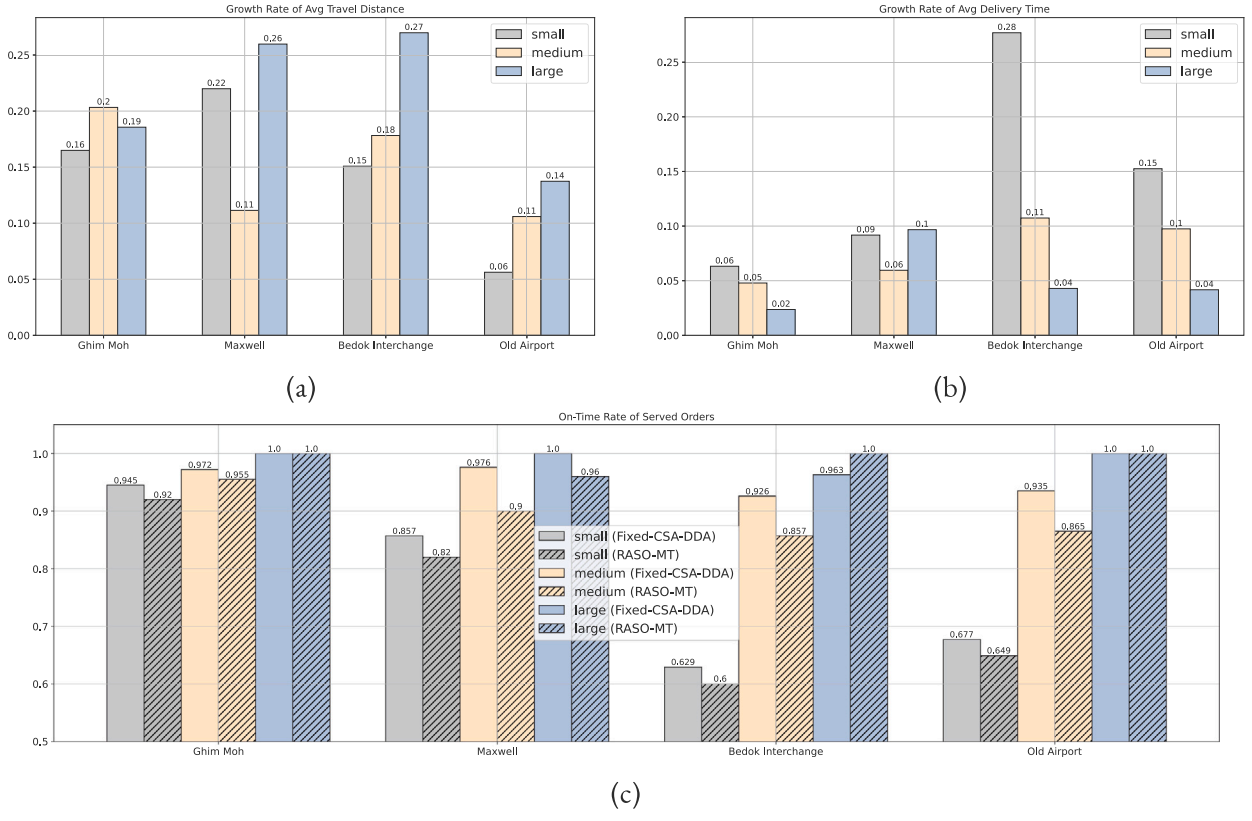

Fig. 13. Experiment results for selected hawker centres: (a) growth rate of average travel distance, (b) growth rate of average delivery time, and (c) on-time rate of served orders.

Table 11
Number of orders served and average order delivery time for different polices.

Policy	RASO-MT	Fixed-DDA	Fixed-CSA	Fixed-CSA-DDA
Number of Orders Served	**802**	686	673	673
Order Delivery Time (minute)	**42.7**	40.4	39.5	39.2

$\mathbb{M}^{o}_{\text{Rat}}$, and the actual average order delivery time for each time period $t \in \mathcal{T}$. Compared with policies **Fixed-DDA**, **Fixed-CSA** and **Fixed-CSA-DDA**, we find that the **RASO-MT** model serves the most orders with 16.9%, 19.2% and 19.2% improvements, and also causes the average delivery time increased by 5.7%, 8.0% and 8.9%, respectively. It is worth noting that the average delivery time of orders for Fixed-CSA is slightly higher than that of Fixed-CSA-DDA. This is because, for Fixed-CSA-DDA, both the CSA and DDA are fixed at 5 km, whereas for Fixed-CSA, since the CSA is fixed at 5 km, the model only ensures that the delivery time constraint is not violated and finds it unnecessary to expand the DDA to bring in more drivers for delivery. Therefore, the DDA is set to be slightly less than 5 km, resulting in fewer drivers being used for delivery and consequently, a slightly higher average delivery time of orders compared to Fixed-CSA-DDA. The results indicate that adjusting the CSA and DDA simultaneously for restaurants can result in substantial service improvements at the cost of slightly longer delivery times.

6.2. Comparison with grid search and computational scalability

We further underscore the advantages of the **RASO-MT** framework, which integrates a predictive tree-based model into the MIQCP optimization framework by comparing it with a straightforward grid search method.

- **Grid Search**: This method determines the CSA and DDA radii for restaurants by incrementally increasing them from 2 km to 10 km in 0.2-kilometer steps until the predicted average customer delay surpasses the threshold.

Table 12 presents the comparison between the **RASO-MT** framework and the **Grid Search** method, emphasizing the number of orders served and the average order delivery time. The results demonstrate that our framework achieves a higher number of orders with reduced average delivery times.

Table 12
Comparison of RASO-MT and grid search methods at Ghim Moh Market & Food Centre.

Policy	RASO-MT	Grid search
Number of Orders Served	**802**	791
Order Delivery Time (minute)	**42.7**	45.0

Table 13
Characteristics of the MIQCP model for Ghim Moh Market & Food Centre solved by Gurobi 11.0.0.

Model	Value
Rows & Columns	4 & 8
Variables	8 (continuous), 30 (integers with 29 binary)
Quadratic constraints	3
General constraints	129
Presolved Rows & Columns	12 & 1,108
SOS constraints	1
Computational time	$\approx$ 2.1 s

Additionally, we discuss the computational scalability of the **RASO-MT** framework. As detailed in Section 4.5, representing a trained model tree as a Mixed Integer Linear Program (MILP) introduces additional decision variables for each branch node and leaf node, increasing the **RASO-MT** model's complexity. Specifically, the model's complexity escalates with the tree's size, due to an increase in the required binary decision variables for linearization. Due to the four straightforward yet effective features extracted for delivery time prediction, the model tree depth that achieves the best validation and test performance does not exceed 5. As a result, the integration of the model tree into our **RASO-MT** framework requires no more than 32 additional binary decision variables. Moreover, leveraging Gurobi 11.0.0's robust capability for MIQCP problems, all experimental instances were resolved within 5 s. Table 13 presents the detailed attributes and the computational time of the MIQCP model for Ghim Moh Market & Food Centre as optimized by Gurobi 11.0.0. Considering that the optimization of CSA and DDA is a real-time decision-making problem at the restaurant level, the computational efforts required by the solution approach are acceptable.

6.3. The impact of the threshold for order delay

To understand the impact of the threshold for order delay on performance, we calculate the number of orders served with the varying values of $\epsilon_{\max}$ from −10 min (orders are required to deliver within 40 min) to 10 min (orders are required to be delivered within 60 min) for all policies.[5] However, the delivery time constraint may be violated when we have a negative value of $\epsilon_{\max}$ for **Fixed-CSA-DDA**, since the fixed radius of CSA $\rho_c = 5$ km and radius of DDA $\rho_d = 5$ km were specially selected to allow orders to be delivered within 50 min. To ensure fairness in comparison, we decrease the radius of CSA ρ_c for **Fixed-CSA-DDA** from 5 km to 4.5 and 4 km when $\epsilon_{\max} = -5$ and $\epsilon_{\max} = -10$, respectively, to satisfy the delivery time constraint (24).

As shown in Fig. 14(a), as $\epsilon_{\max}$ becomes larger, both **RASO-MT** and **Fixed-DDA** can serve more orders. We also observe that the **RASO-MT** policy can provide the best solution quality with the largest number of orders served. When $\epsilon_{\max}$ is set to −10, -5, 0, 5, and 10 min, **RASO-MT** serves 3.1%, 13.4%, 19.2%, 20.6% and 20.6% more orders compared to **Fixed-CSA**, and serves 11.0%, 16.6%, 16.9%, 10.5% and 6.8% more orders compared to **Fixed-DDA**. Furthermore, compared with policies with a fixed radius of CSA ρ_c, the relative improvement in the number of orders served by our **RASO-MT** framework will gradually increase with longer allowed delays, until it reaches the maximum value of 20.6% (ρ_c equals 10 km). This indicates that our **RASO-MT** is more advantageous when customers are more patient with delivery times. Finally, for **RASO-MT**, we see that the increased number of orders served is 69, 39, 10, and 0 with each 5-minute increment in the value of ϵ_{max} from −10 to 10. These observations imply that allowing more time for delivery helps achieve more orders, while the benefits progressively decrease.

We also calculate the actual order delivery time in each time period $t \in \mathcal{T}$ with different $\epsilon_{\max}$, as depicted in box plots in Fig. 14(b). As can be observed, for policies that can optimize the radius of CSA ρ_c (i.e., **RASO-MT** and **Fixed-DDA**) to serve more orders, the average order delivery time increases proportionally with the number of orders served. For example, we find that the average order delivery time for **RASO-MT** is always larger than **Fixed-DDA** across all $\epsilon_{\max}$, since **RASO-MT** can always serve more orders than **Fixed-DDA**. However, for **Fixed-CSA** that can only optimize the radius of DDA ρ_d, we also find that as $\epsilon_{\max}$ increases, so does the actual average order delivery time. The explanation for this is a larger $\epsilon_{\max}$ will make the policy presume that customers are more lenient with the order delivery time, resulting in a smaller DDA radius, fewer available drivers, and a longer delivery time.

It is worth noting that **Fixed-CSA-DDA** serves fewer orders at $\epsilon_{\max}$ equals to −10 and -5, which are 582 and 638, respectively, compared to **Fixed-CSA**, which serves 673 orders for both $\epsilon_{\max}$ values of −10 and -5. This happens because we decrease the radius of

[5] In industry practice, the platforms usually require drivers an ETA which is different from the ETA promised to the customers. This difference can be represented by the parameter $\epsilon_{\max}$. From another perspective, the impact of $\epsilon_{\max}$ can also be interpreted as the impact of ETA (i.e., L^o) when $\epsilon_{\max} = 0$. For example, assume the initial $L^o = 50$ and $\epsilon_{\max} = 0$. If customers are more patient with delivery times, the platform can increase L^o to 60 min. Practically, this is equivalent to raising $\epsilon_{\max}$ to 10 min while keeping $L^o = 50$ unchanged.

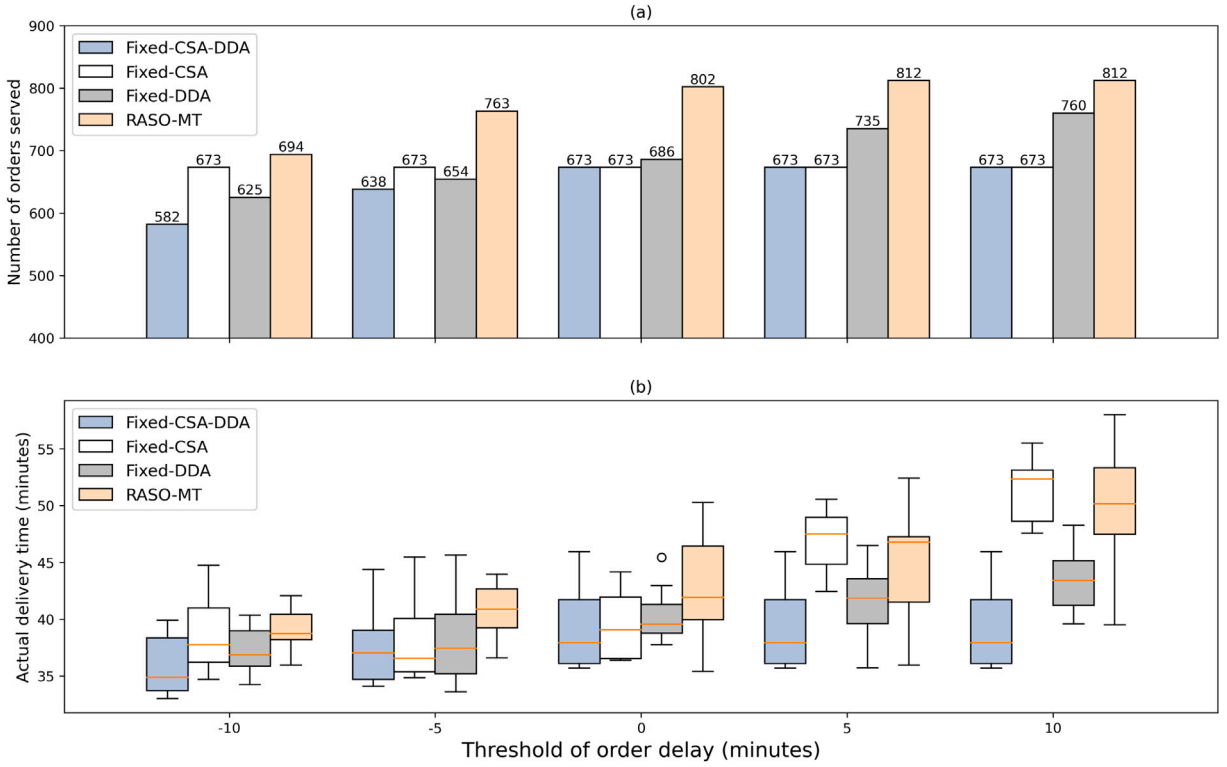

Fig. 14. Impact of the threshold for order delay ($\epsilon_{\max}$) on (a) number of orders served and (b) actual delivery time of orders.

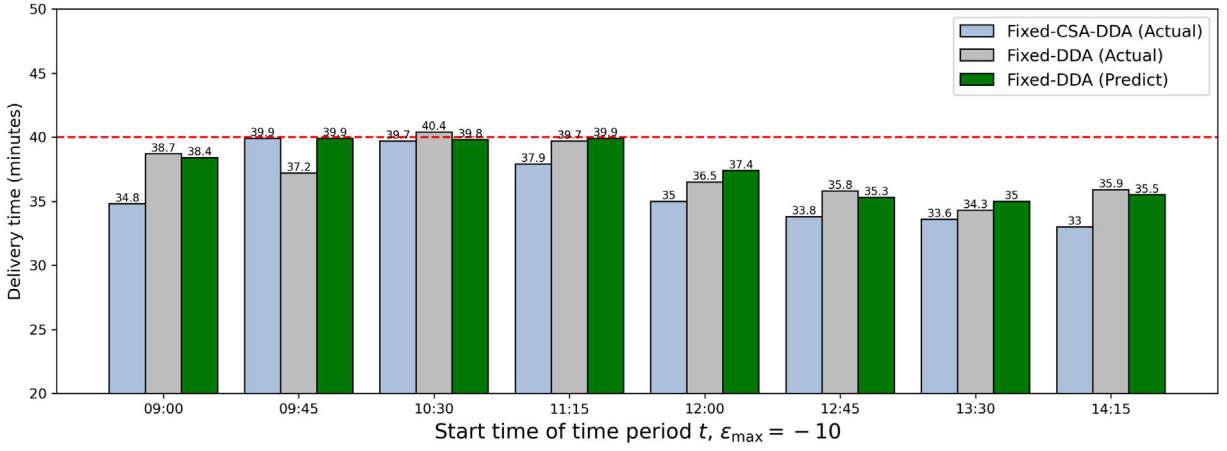

Fig. 15. Comparisons of average order delivery time with $\epsilon_{\max}$ equals to −10.

CSA ρ_c from 5 km to 4 km ($\epsilon_{\max} = -10$) and 4.5 km ($\epsilon_{\max} = -5$) to ensure the delivery time constraint (24) that the predicted average order delivery time $L_r(\rho_c, \rho_d)$ must be less than or equal to $L^o + \epsilon_{\max}$ (by default, the promised delivery time $L^o = 50$) min in each time period $t \in \mathcal{T}$. To better comprehend the importance of the delivery time constraint, we compare **Fixed-CSA-DDA** and **Fixed-DDA**, the two policies with the fewest served orders. We compare the actual average order delivery time of **Fixed-CSA-DDA**, **Fixed-DDA** and the predicted average order delivery time of **Fixed-DDA** with $\epsilon_{\max}$ values −10 for $t \in \mathcal{T}$ from 09:00 to 15:00.[6]. The results can be found in Fig. 15. For **Fixed-DDA**, we observe that the predicted order delivery time is always less than 40 min ($L^o + \epsilon_{\max}$), and

[6] Given the same instance, the "actual" means the calculated average order delivery time achieved via the simulation procedure described in Algorithm 2 The "predicted" denotes the predicted average order delivery time by the trained model tree in Section 5.4.

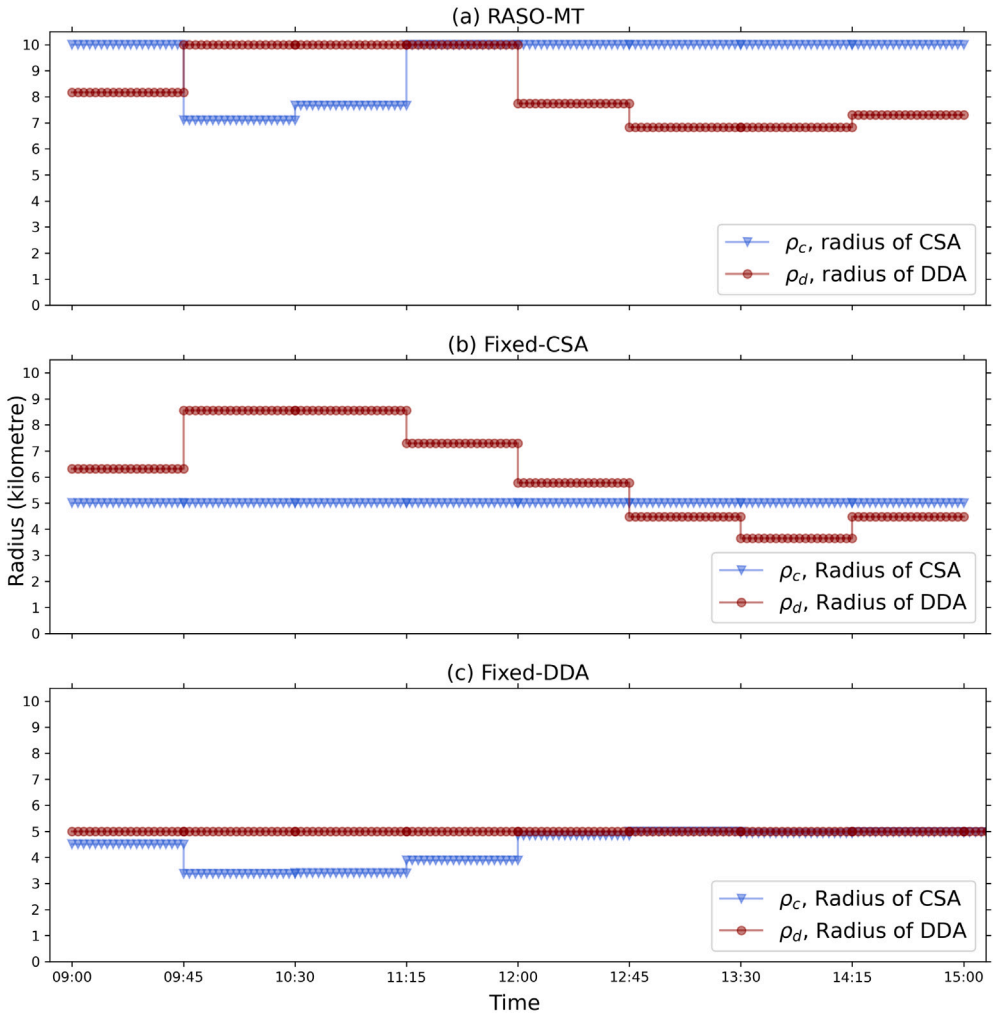

Fig. 16. Radius over time for policies: (a) RASO-MT, (b) Fixed-CSA, and (c) Fixed-DDA.

the actual average order delivery time only exceeds 40 min slightly in time period 10:30 to 11:15 due to the prediction errors. The actual delivery time of **Fixed-CSA-DDA** never exceeds 40 min even between both 09:45 to 10:30 and 10:30 to 11:15 when the peak of orders occurs. We also find that the actual delivery time of **Fixed-DDA** is greater than that of **Fixed-CSA-DDA** in every time period t. This is because the more orders that are served, the longer the average order delivery time will be, and **Fixed-DDA** can serve more orders by adjusting CSA radius ρ_c.

6.4. Comparison of radius adjustments over time

Next, we investigate how these policies generate the radii of the restaurant's CSA ρ_c and DDA ρ_d over time $t \in \mathcal{T}$, except for **Fixed-CSA-DDA**. Fig. 16 shows the radius adjustments of the three policies. When only one of the areas can be adjusted, we find that policies **Fixed-CSA** and **Fixed-DDA** react to the peak periods (from 09:45 to 12:00) by setting a larger radius of DDA ρ_d or a smaller radius of CSA ρ_c, respectively, and to the off-peaks (from 09:00 to 09:45 and 12:00 to 15:00) by setting a smaller radius of DDA ρ_d or a larger radius of CSA ρ_c, respectively. In addition, it is interesting to observe that **RASO-MT** can adjust the CSA and DDA simultaneously by setting the radius of DDA ρ_d to the maximum value and a relatively moderate radius of CSA ρ_c during the peak periods. During the off-peak periods, as the platform aims to serve more orders, the radius of CSA ρ_c is set to the maximum value and the radius of DDA ρ_d can be smaller to meet the demand. Overall, we observe that it is always beneficial to maintain a large radius of CSA ρ_c when the restaurant has sufficient service capacity, which can bring more drivers to satisfy the order demand, and the radius of CSA ρ_c will only be decreased to ensure orders will not be delayed when no more drivers can be found (ρ_d reaches its maximum value, 10 km).

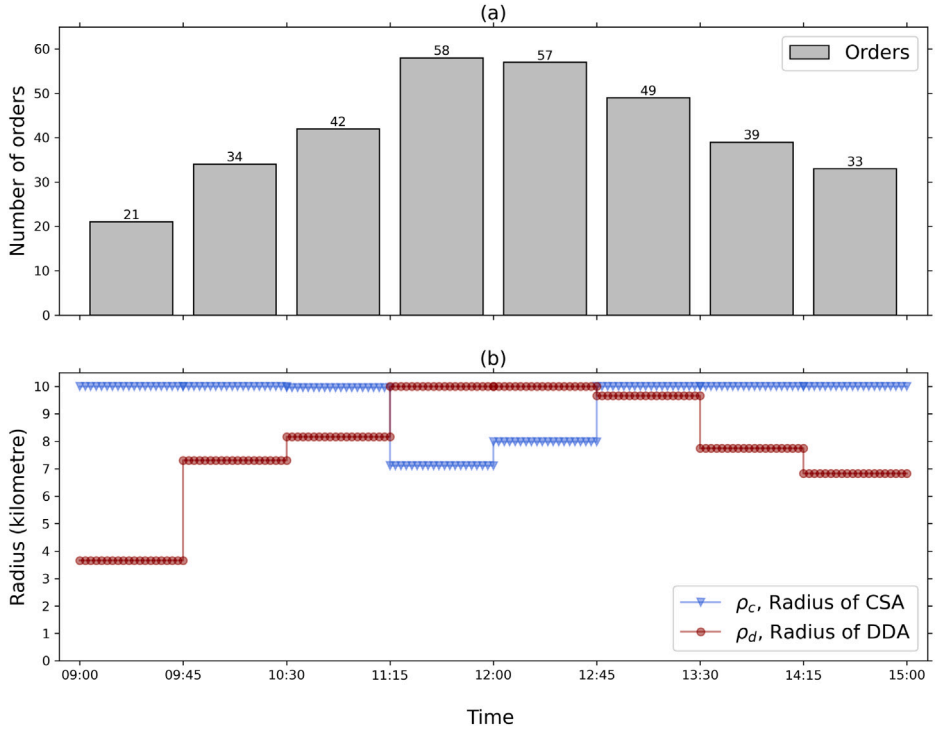

Fig. 17. Radius over time (RASO-MT) with customer orders arrival distribution.

6.5. Comparison with different customer order arrival rates

Last, we analyze the radius adjustment decisions over time $t \in \mathcal{T}$ for different customer order arrival rates. In a peak scenario, where customer order arrival rates are more than double those of off-peak periods, we illustrate the radius adjustment decisions with **RASO-MT** in Fig. 17. The results demonstrate that the DDA radius adjustments align with the changes in customer order arrival rates. Moreover, the CSA radius ρ_c only decreases to around 7 km during peak periods from 11:15 to 12:45, which corresponds to the highest order peak.

By combining the results presented in Figs. 16 and 17, we observe that our **RASO-MT** approach may primarily adjust the radius of the restaurant's DDA ρ_d first, and then adapts the CSA radius ρ_c to meet the average order delivery time constraint. This adjustment allows us to reduce the CSA radius when the restaurant encounters a shortage of service capacity during peak periods.

7. Conclusion

This paper introduces the restaurant area sizing optimization problem (RASO) as a new operational problem for managing supply and demand in on-demand food delivery services. The main objective is to maximize the total number of orders served while ensuring a required service level for order delivery time. Initially, we examine the relationship between the radius of the customer service area and the number of customer orders received by the restaurant, focusing on the demand side. Subsequently, we analyze the number of drivers, which depends on the radius of the driver dispatch area, and explore various factors to predict the average order delivery time on the supply side. To model the problem, we integrate closed-form formulations for order estimation using a customer choice model and delivery time prediction with model tree. The resulting integrated model is formulated as an MIQCP (Mixed Integer Quadratically Constrained Program) and can be efficiently solved using Gurobi as the optimization solver.

We utilize a customized simulator that simulates order generation, placement, and dispatch, along with real-world food delivery data provided by our industry partner to conduct extensive experiments. The objective is to evaluate the performance of our proposed RASO-MT model and compare it with other benchmark area sizing policies. The results show a significant performance improvement achieved through our approach, indicating that simultaneous adjustments of the radii of the customer service area (CSA) and driver dispatch area (DDA) can substantially increase the total number of orders served within an acceptable delivery time. Furthermore, we investigate the impact of various factors related to demand, supply, and the threshold for order delay on radius adjustment decisions over time.

Several future research directions can be explored. First, although overlapping CSAs were discussed in Section 4.6, future research should investigate their nuanced effects on customer demand estimation. We intend to refine demand estimation collecting more real-world delivery data and explore advanced deep learning methods for more accurate customer demand modeling. Another

natural avenue involves modeling the RASO problem as a Markov Decision Process and utilizing reinforcement learning approaches to establish a policy for sequential decision-making on CSA and DDA from end to end. Additionally, synergizing our area sizing optimization with other powerful tools, such as surge pricing, could be explored. For instance, in scenarios with extreme under-supply where increasing the DDA fails to bring sufficient supply to serve orders placed within the CSA with a radius $\rho_c^{\min}$, an integrated optimization method that combines area sizing and surge pricing could prove valuable.

CRediT authorship contribution statement

Jingfeng Yang: Data curation, Formal analysis, Methodology, Software, Visualization, Writing – original draft. **Hoong Chuin Lau:** Conceptualization, Investigation, Methodology, Project administration, Supervision, Writing – review & editing. **Hai Wang:** Conceptualization, Data curation, Funding acquisition, Investigation, Methodology, Project administration, Supervision, Writing – review & editing.

Acknowledgments

The corresponding author acknowledges the great support from the Lee Kong Chian Fellowship awarded by Singapore Management University.

Author's Note: The majority of the research presented in this paper was conducted while the first author was a Ph.D. student at the School of Computing and Information Systems, Singapore Management University. The subsequent revisions and finalization of the manuscript were completed in the author's current capacity as a Research Scientist at the Singapore Institute of Manufacturing Technology. We acknowledge both institutions for their support and resources that significantly contributed to the development of this work.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to improve readability and language of the work. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Appendix

See Figs. A.1 and A.2 and Tables A.1 and A.2

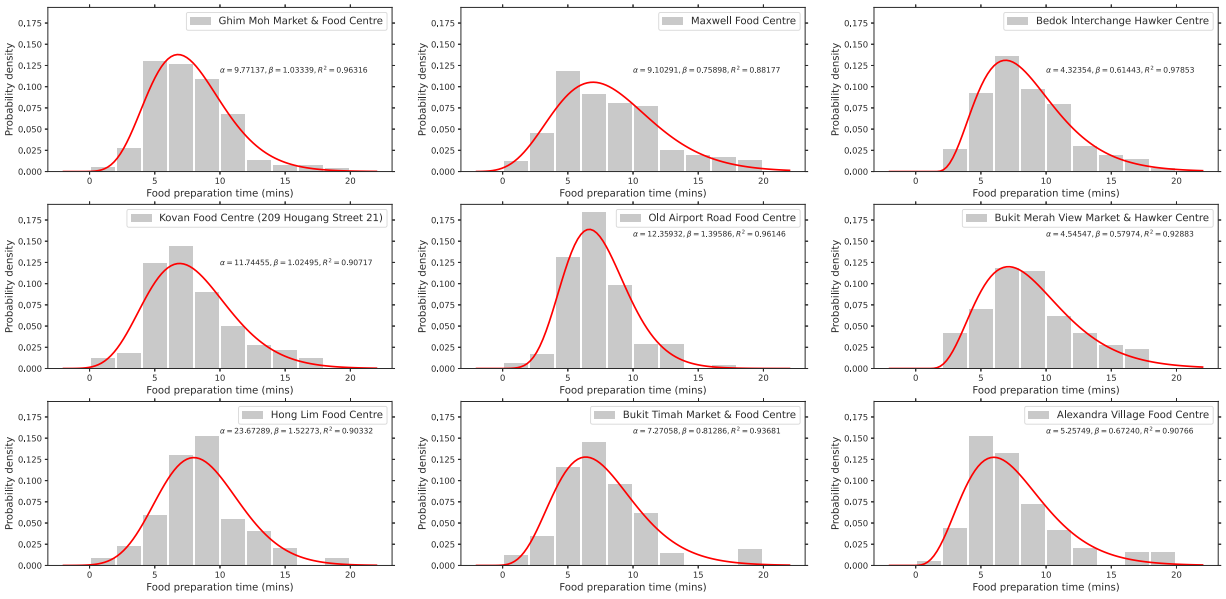

Fig. A.1. Histogram of customer restaurants' orders preparation time for all nine hawker centres.

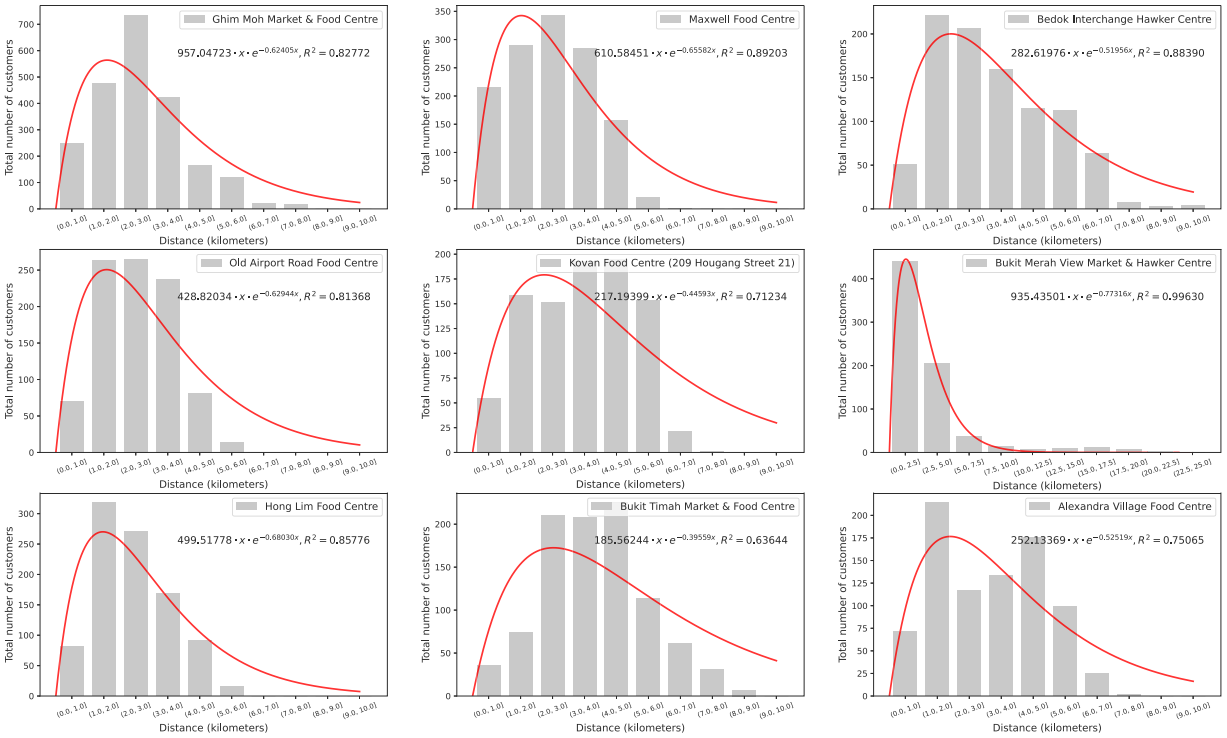

Fig. A.2. Histograms of customer orders based on customer-restaurant distance segments for all nine hawker centres.

Table A.1
Performance of ML methods on average delivery time prediction.

(a) Kovan Food Centre

Model	R^2	MAPE
Ordinal Least Square	0.598	0.507
Ridge Regression	0.598	0.506
Linear SVR	0.481	0.366
Regression Tree	0.902	0.153
Model Tree	**0.922**	**0.133**
XGBoost	0.851	0.182

(b) Bukit Merah View Market & Hawker Centre

Model	R^2	MAPE
Ordinal Least Square	0.429	0.524
Ridge Regression	0.428	0.523
Linear SVR	0.117	0.244
Regression Tree	0.845	0.189
Model Tree	**0.856**	**0.169**
XGBoost	0.826	0.205

(c) Hong Lim Food Centre

Model	R^2	MAPE
Ordinal Least Square	0.452	0.648
Ridge Regression	0.452	0.647
Linear SVR	0.263	0.356
Regression Tree	0.890	0.195
Model Tree	**0.895**	**0.179**
XGBoost	0.804	0.268

(d) Bukit Timah Market & Food Centre

Model	R^2	MAPE
Ordinal Least Square	0.571	0.551
Ridge Regression	0.572	0.551
Linear SVR	0.450	0.373
Regression Tree	0.919	0.139
Model Tree	**0.939**	**0.113**
XGBoost	0.900	0.171

(e) Alexandra Village Food Centre

Model	R^2	MAPE
Ordinal Least Square	0.487	0.565
Ridge Regression	0.487	0.564
Linear SVR	0.270	0.315
Regression Tree	0.891	0.183
Model Tree	**0.901**	**0.162**
XGBoost	0.842	0.219

Table A.2
Experimental results for all nine hawker centres.

(a) Ghim Moh Hawker & Food Centre

Method/Served Orders	Drivers	Size	Total distance (km)	Avg distance (km)	Delayed orders	Avg delay (s)	On-time rate	Avg delivery time (s)
Fixed-CSA-DDA (73)	20	small	311.52	15.58	0	0	1.000	2,307
	25	medium	320.95	12.84	0	0	1.000	2,244
	30	large	329.77	10.99	0	0	1.000	2,215
RASO-MT (88)	20	small	362.97	18.15	7	40.00	0.920	2,377
	25	medium	386.36	15.45	4	8.88	0.955	2,294
	30	large	391.00	13.03	0	0	1.000	2,239

(b) Maxwell Food Centre

Method/Served Orders	Drivers	Size	Total distance (km)	Avg distance (km)	Delayed orders	Avg delay (s)	On-time rate	Avg delivery time (s)
Fixed-CSA-DDA (42)	10	small	170.47	17.05	2	16.59	0.952	2,508
	15	medium	192.29	12.82	1	2.76	0.976	2,342
	20	large	196.71	9.84	0	0	1.000	2,234
RASO-MT (50)	10	small	208.03	20.80	9	106.93	0.820	2,628
	15	medium	228.93	15.26	5	31.71	0.900	2,410
	20	large	245.41	12.27	2	28.10	0.960	2,334

(c) Bedok Interchange Hawker Centre

Method/Served Orders	Drivers	Size	Total distance (km)	Avg distance (km)	Delayed orders	Avg delay (s)	On-time rate	Avg delivery time (s)
Fixed-CSA-DDA (27)	5	small	107.06	21.41	10	137.96	0.629	2,767
	10	medium	135.82	13.58	2	10.07	0.926	2,392
	15	large	137.91	9.19	1	0.52	0.963	2,295
RASO-MT (35)	5	small	123.20	24.64	14	663.70	0.600	3,201
	10	medium	160.00	16.00	5	63.56	0.857	2,520
	15	large	175.00	11.67	0	0	1.000	2,342

(d) Old Airport Road Food Centre

Method/Served Orders	Drivers	Size	Total distance (km)	Avg distance (km)	Delayed orders	Avg delay (s)	On-time rate	Avg delivery time (s)
Fixed-CSA-DDA (31)	5	small	113.77	22.75	10	257.06	0.677	2,754
	10	medium	133.05	13.31	2	2.58	0.935	2,298
	15	large	145.18	9.68	0	0	1.000	2,231
RASO-MT (37)	5	small	120.17	24.03	13	504.10	0.649	2,991
	10	medium	147.16	14.72	5	70.58	0.865	2,405
	15	large	165.13	11.01	0	0	1.000	2,274

(e) Kovan Food Centre

Method/Served Orders	Drivers	Size	Total distance (km)	Avg distance (km)	Delayed orders	Avg delay (s)	On-time rate	Avg delivery time (s)
Fixed-CSA-DDA (65)	20	small	342.43	17.12	6	38.25	0.907	2,565
	25	medium	355.72	13.43	3	21.36	0.953	2,495
	30	large	374.81	12.49	0	0	1.000	2,440
RASO-MT (90)	20	small	426.25	21.31	15	75.43	0.833	2,767
	25	medium	428.28	17.13	15	63.66	0.833	2,706
	30	large	463.96	15.46	6	21.93	0.933	2,526

(f) Bukit Merah View Market & Hawker Centre

Method/Served Orders	Drivers	Size	Total distance (km)	Avg distance (km)	Delayed orders	Avg delay (s)	On-time rate	Avg delivery time (s)
Fixed-CSA-DDA (55)	10	small	182.95	18.21	0	0	1.000	2,316
	15	medium	208.17	13.88	0	0	1.000	2,154
	20	large	220.93	11.05	0	0	1.000	2,125
RASO-MT (60)	10	small	184.99	18.50	0	0	1.000	2,435
	15	medium	219.87	14.66	0	0	1.000	2,189
	20	large	220.36	11.02	0	0	1.000	2,151

(g) Hong Lim Food Centre

Method/Served Orders	Drivers	Size	Total distance (km)	Avg distance (km)	Delayed orders	Avg delay (s)	On-time rate	Avg delivery time (s)
Fixed-CSA-DDA (40)	5	small	106.10	21.22	13	191.55	0.675	2,962
	10	medium	156.67	15.67	1	1.88	0.975	2,348
	15	large	178.00	11.87	0	0	1.000	2,289
RASO-MT (47)	5	small	120.76	24.15	19	267.55	0.596	3,071
	10	medium	182.06	18.21	1	9.74	0.978	2,449
	15	large	203.74	13.58	0	0	1.000	2,295

(*continued on next page*)

Table A.2 (*continued*).

(h) Bukit Timah Market & Food Centre

Method/Served Orders	Drivers	Size	Total distance (km)	Avg distance (km)	Delayed orders	Avg delay (s)	On-time rate	Avg delivery time (s)
Fixed-CSA-DDA (67)	25	small	378.47	15.14	3	29.61	0.955	2,569
	30	medium	397.74	13.26	1	11.90	0.985	2,531
	35	large	412.66	11.79	0	0	1.000	2,511
RASO-MT (104)	25	small	527.06	21.08	22	115.71	0.788	2,863
	30	medium	539.93	17.99	15	62.04	0.856	2,710
	35	large	544.70	15.56	14	59.39	0.865	2,685

(i) Alexandra Village Food Centre

Method/Served Orders	Drivers	Size	Total distance (km)	Avg distance (km)	Delayed orders	Avg delay (s)	On-time rate	Avg delivery time (s)
Fixed-CSA-DDA (42)	10	small	179.20	17.92	2	36.83	0.952	2,508
	15	medium	203.17	13.54	1	7.67	0.976	2,408
	20	large	227.35	11.37	0	0	1.000	2,366
RASO-MT (56)	10	small	223.19	22.32	15	188.12	0.732	2,878
	15	medium	255.57	17.04	2	12.52	0.964	2,421
	20	large	258.08	12.90	0	0	1.000	2,383

Note. Drivers: Number of drivers in the system; **Size**: Size of the instance; **Total Distance (km)**: Total travel distance of all drivers for the delivery; **Avg Distance (km)**: Average travel distance per driver to deliver their dispatched orders; **Delayed Orders**: Number of orders that cannot be delivered within the promised delivery time; **Avg Delay (s)**: Average delay of all orders; **On-Time Rate**: Ratio of orders delivered on time to total orders; **Avg Delivery Time (s)**: Average delivery time for served orders.

References

Agatz, N., Cho, S.H., Sun, H., Wang, H., 2024. Transportation-enabled services: Concept, framework, and research opportunities. Serv. Sci. (1), 1–21.

Agussurja, L., Cheng, S.F., Lau, H.C., 2019. A state aggregation approach for stochastic multiperiod last-mile ride-sharing problems. Transp. Sci. 53 (1), 148–166.

Angrist, J.D., Caldwell, S., Hall, J.V., 2021. Uber versus taxi: A driver's eye view. Am. Econ. J.: Appl. Econ. 13 (3), 272–308.

Auad, R., Erera, A., Savelsbergh, M., 2020. Using simple integer programs to assess capacity requirements and demand management strategies in meal delivery. Preprint, Optimization Online.

Bahrami, S., Nourinejad, M., Nesheli, M.M., Yin, Y., 2022. Optimal composition of solo and pool services for on-demand ride-hailing. Transp. Res. Part E: Logist. Transp. Rev. 161, 102680.

Bahrami, S., Nourinejad, M., Yin, Y., Wang, H., 2023. The three-sided market of on-demand delivery. Transp. Res. Part E: Logist. Transp. Rev. 179, 103313.

Bai, J., So, K.C., Tang, C.S., Chen, X., Wang, H., 2019. Coordinating supply and demand on an on-demand service platform with impatient customers. Manuf. Serv. Oper. Manag. 21 (3), 556–570.

Banerjee, D., Erera, A.L., Stroh, A.M., Toriello, A., 2023. Who has access to e-commerce and when? Time-varying service regions in same-day delivery. Transp. Res. B 170, 148–168.

Beardwood, J., Halton, J.H., Hammersley, J.M., 1959. The shortest path through many points. In: Mathematical Proceedings of the Cambridge Philosophical Society, vol. 55, (4), Cambridge University Press, pp. 299–327.

Bimpikis, K., Candogan, O., Saban, D., 2019. Spatial pricing in ride-sharing networks. Oper. Res. 67 (3), 744–769.

Bozanta, A., Cevik, M., Kavaklioglu, C., Kavuk, E.M., Tosun, A., Sonuc, S.B., Duranel, A., Basar, A., 2022. Courier routing and assignment for food delivery service using reinforcement learning. Comput. Ind. Eng. 164, 107871.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining. pp. 785–794.

Chen, Y., Wang, H., 2018. Pricing for a last-mile transportation system. Transp. Res. B 107, 57–69.

Chen, J.F., Wang, L., Ren, H., Pan, J., Wang, S., Zheng, J., Wang, X., 2022. An imitation learning-enhanced iterated matching algorithm for on-demand food delivery. IEEE Trans. Intell. Transp. Syst. 23 (10), 18603–18619.

Ding, X., Zhang, R., Mao, Z., Xing, K., Du, F., Liu, X., Wei, G., Yin, F., He, R., Sun, Z., 2020. Delivery scope: A new way of restaurant retrieval for on-demand food delivery service. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 3026–3034.

Du, J., Zhang, Z., Wang, X., Lau, H.C., 2023. A hierarchical optimization approach for dynamic pickup and delivery problem with LIFO constraints. Transp. Res. Part E: Logist. Transp. Rev. 175, 103131.

Feldman, P., Frazelle, A.E., Swinney, R., 2018. Service delivery platforms: Pricing and revenue implications. 3258739, Available at SSRN.

Frank, E., Wang, Y., Inglis, S., Holmes, G., Witten, I.H., 1998. Using model trees for classification. Mach. Learn. 32 (1), 63–76.

Gama, J., 2004. Functional trees. Mach. Learn. 55 (3), 219–250.

Gao, C., Zhang, F., Wu, G., Hu, Q., Ru, Q., Hao, J., He, R., Sun, Z., 2021. A deep learning method for route and time prediction in food delivery service. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 2879–2889.

Gao, C., Zhang, F., Zhou, Y., Feng, R., Ru, Q., Bian, K., He, R., Sun, Z., 2022. Applying deep learning based probabilistic forecasting to food preparation time for on-demand delivery service. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 2924–2934.

Guo, X., Haupt, A., Wang, H., Qadri, R., Zhao, J., 2023. Understanding multi-homing and switching by platform drivers. Transp. Res. C 154, 104233.

Hildebrandt, F.D., Ulmer, M.W., 2021. Supervised learning for arrival time estimations in restaurant meal delivery. Transp. Sci..

Ke, J., Chen, X.M., Yang, H., Li, S., 2022. Coordinating supply and demand in ride-sourcing markets with pre-assigned pooling service and traffic congestion externality. Transp. Res. Part E: Logist. Transp. Rev. 166, 102887.

Ke, J., Yang, H., Li, X., Wang, H., Ye, J., 2020. Pricing and equilibrium in on-demand ride-pooling markets. Transp. Res. B 139, 411–431.

Li, X., Li, X., Wang, H., Shi, J., Aneja, Y.P., 2022. Supply regulation under the exclusion policy in a ride-sourcing market. Transp. Res. B 166, 69–94.

Liang, J., Ke, J., Wang, H., Ye, H., Tang, J., 2023. A Poisson-based distribution learning framework for short-term prediction of food delivery demand ranges. IEEE Trans. Intell. Transp. Syst. 24 (12), 14556–14569.

Liu, S., He, L., Shen, Z.J.M., 2021. On-time last-mile delivery: Order assignment with travel-time predictors. Manage. Sci. 67 (7), 4095–4119.

Liu, Y., Ouyang, Y., 2023. Planning ride-pooling services with detour restrictions for spatially heterogeneous demand: A multi-zone queuing network approach. Transp. Res. B 174, 102779.

Liu, T., Xu, Z., Vignon, D., Yin, Y., Li, Q., Qin, Z., 2023. Effects of threshold-based incentives on drivers' labor supply behavior. Transp. Res. C 152, 104140.

Loh, W.Y., 2011. Classification and regression trees. Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 1 (1), 14–23.

Luo, Q., Nagarajan, V., Sundt, A., Yin, Y., Vincent, J., Shahabi, M., 2023. Efficient algorithms for stochastic ride-pooling assignment with mixed fleets. Transp. Sci..

Lyu, G., Cheung, W.C., Teo, C.P., Wang, H., 2024. Multiobjective stochastic optimization: A case of real-time matching in ride-sourcing markets. Manuf. Serv. Oper. Manag. 26 (2), 500–518.
MacKay, A., Svartbäck, D., Ekholm, A.G., 2022. Dynamic pricing and demand volatility: Evidence from restaurant food delivery. Available at SSRN 4164271.
Mankad, S., Shunko, M., Yu, Q., 2019. How to find your most valuable service outlets: Measuring influence using network analysis. Available at SSRN.
McKinsey & Company, 2021. Ordering in: The rapid evolution of food delivery. https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/ordering-in-the-rapid-evolution-of-food-delivery. Last Accessed on 14 July 2022.
Nextbite, 2021. What consumers really want from food delivery. https://www.nrn.com/delivery-takeout-solutions/what-consumers-really-want-food-delivery, Last Accessed on 5 June 2023.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.
Potts, D., Sammut, C., 2005. Incremental learning of linear model trees. Mach. Learn. 61 (1), 5–48.
Psaraftis, H.N., Wen, M., Kontovas, C.A., 2016. Dynamic vehicle routing problems: Three decades and counting. Networks 67 (1), 3–31.
Qin, Z., Tang, X., Jiao, Y., Zhang, F., Xu, Z., Zhu, H., Ye, J., 2020. Ride-hailing order dispatching at didi via reinforcement learning. INFORMS J. Appl. Anal. 50 (5), 272–286.
Quinlan, J.R., et al., 1992. Learning with continuous classes. In: 5th Australian Joint Conference on Artificial Intelligence, vol. 92, World Scientific, pp. 343–348.
Reyes, D., Erera, A., Savelsbergh, M., Sahasrabudhe, S., O'Neil, R., 2018. The meal delivery routing problem. Optim. Online 6571.
Salari, N., Liu, S., Shen, Z.-J.M., 2022. Real-time delivery time forecasting and promising in online retailing: when will your package arrive? Manuf. Serv. Oper. Manag..
Sun, H., Wang, H., Wan, Z., 2019. Model and analysis of labor supply for ride-sharing platforms in the presence of sample self-selection and endogeneity. Transp. Res. B 125, 76–93.
Taylor, T.A., 2018. On-demand service platforms. Manuf. Serv. Oper Manag. 20 (4), 704–720.
Tong, T., Dai, H., Xiao, Q., Yan, N., 2020. Will dynamic pricing outperform? Theoretical analysis and empirical evidence from O2O on-demand food service market. Int. J. Prod. Econ. 219, 375–385.
Train, K.E., 2009. Discrete Choice Methods with Simulation. Cambridge University Press.
Ulmer, M.W., Erera, A., Savelsbergh, M., 2022. Dynamic service area sizing in urban delivery. OR Spectrum 1–31.
Ulmer, M.W., Thomas, B.W., Campbell, A.M., Woyak, N., 2021. The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. Transp. Sci. 55 (1), 75–100.
Vignon, D., Yin, Y., Ke, J., 2023. Regulating the ride-hailing market in the age of uberization. Transp. Res. Part E: Logist. Transp. Rev. 169, 102969.
Wang, H., 2022. Transportation-enabled urban services: A brief discussion. Multimodal Transp. 1 (2), 100007.
Wang, Z., Fu, K., Ye, J., 2018. Learning to estimate the travel time. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 858–866.
Wang, H., Odoni, A., 2016. Approximating the performance of a "last mile" transportation system. Transp. Sci. 50 (2), 659–675.
Wang, H., Tang, X., Kuo, Y.H., Kifer, D., Li, Z., 2019. A simple baseline for travel time estimation using large-scale trip data. ACM Trans. Intell. Syst. Technol. 10 (2), 1–22.
Wang, Y., Witten, I.H., 1996. Induction of model trees for predicting continuous classes.
Wang, H., Yang, H., 2019. Ridesourcing systems: A framework and review. Transp. Res. B 129, 122–155.
Wang, Y., Zheng, Y., Xue, Y., 2014. Travel time estimation of a path using sparse trajectories. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 25–34.
Weng, W., Yu, Y., 2021. Labor-right protecting dispatch of meal delivery platforms. In: 60th IEEE Conference on Decision and Control, CDC 2021, Austin, TX, USA, December 14-17, 2021. IEEE, pp. 1349–1355. http://dx.doi.org/10.1109/CDC45484.2021.9683147.
Yang, H., Qin, X., Ke, J., Ye, J., 2020a. Optimizing matching time interval and matching radius in on-demand ride-sourcing markets. Transp. Res. B 131, 84–105.
Yang, H., Shao, C., Wang, H., Ye, J., 2020b. Integrated reward scheme and surge pricing in a ridesourcing market. Transp. Res. B 134, 126–142.
Yildiz, B., Savelsbergh, M., 2019a. Provably high-quality solutions for the meal delivery routing problem. Transp. Sci. 53 (5), 1372–1388.
Yildiz, B., Savelsbergh, M., 2019b. Service and capacity planning in crowd-sourced delivery. Transp. Res. C 100, 177–199.
Zhang, K., Nie, Y.M., 2021. To pool or not to pool: Equilibrium, pricing and regulation. Transp. Res. B 151, 59–90.
Zhou, Y., Yang, H., Ke, J., Wang, H., Li, X., 2022. Competition and third-party platform-integration in ride-sourcing markets. Transp. Res. B 159, 76–103.
Zhu, Z., Ke, J., Wang, H., 2021. A mean-field Markov decision process model for spatial-temporal subsidies in ride-sourcing markets. Transp. Res. B 150, 540–565.