Decision models on personal shopper platform operations optimization

Lu Zhen [a], Xueting He [a], Huiwen Wang [a], Gilbert Laporte [b,c], Zheyi Tan [a]

[a] *School of Management, Shanghai University, Shanghai, China*
[b] *Department of Decision Sciences, HEC Montréal, Montréal, Canada*
[c] *School of Management, University of Bath, Bath, United Kingdom*

Abstract: The 'lazy economy' gives rise to an emerging business mode, called personal shopper platforms (PSPs). A customer who needs some goods urgently can release an order on a PSP, which is then assigned by the platform to a personal shopper, who will buy the goods at a nearby retail store and deliver them to the customer within a short time interval. Since the development of PSP is relatively new, the decision mechanisms and policies are at an early stage. The operations of the PSPs can be optimized through operations research methodologies. This study proposes a series of mixed integer programming (MIP) models as well as improved dynamic programming-based algorithms to support operational decisions on order assignment and shopper routing, as well as strategic decisions on the PSP mode adoption and territory planning. Some intuitive but practical criteria are also designed to accelerate the proposed algorithms so that they can be applied to large-scale realistic instances. The proposed algorithm can solve the basic case with 1,000 orders and 1,050 shoppers (about 10^7 variables and 10^7 constraints in the MIP models) in half a minute. A realistic case in the Changning district of Shanghai is also used to validate the effectiveness of the proposed models and the efficiency of the algorithms. This study provides a comprehensive driven-model decision methodology for this emerging service industry mode.

Keywords: Personal shoppers; order assignment; improved dynamic programming; last-mile delivery; new business model.

1. Introduction

In the era of online shopping, various novel business modes are constantly emerging. Recently, a new mode called personal shopper platform (PSP) has gained in popularity in some Chinese metropolises. Customers who need some goods urgently can release an order on a PSP such as "Hummingbird PSP" and "MEITUAN PSP", as shown in Figure 1. The PSP will assign the order to a personal shopper who will buy the goods at a nearby retail store and deliver them to the customer within about one hour. In contrast, under the traditional delivery mode of online shopping, it usually takes more than about ten hours for the goods to reach the customers after they have been ordered from an e-tailer such as Amazon, Tmall, eBay or JD.

It is well known that goods purchased on the traditional online shopping malls are cheaper than those available in the retail stores used by PSPs, which are usually some small convenience stores near the customers. In addition, the delivery cost on PSPs is not negligible, whereas it is lower on traditional online shopping malls and can be waived in most cases (for example, when an order costs more than about 15 USD). However, there are still plenty of customers buying commodities from PSPs because they offer timely deliveries. Thus a PSP can be viewed as an alternative "last-mile" delivery mode in which commodities are picked up from large automated warehouses, routed through multiple tiers of distribution centers, arrive at small warehouses near residential areas (these small warehouses are called by "front storehouses" in China), and are finally delivered to end customers. In a PSP, the retail stores act as the front storehouses. A PSP differs from the traditional delivery mode in that the fulfillment of customer orders begins at widely scattered retail stores rather than at a few large automated warehouses located in city outskirts, and usually very far from the customers. Therefore, the PSP mode offers some advantages and is becoming a promising business model for online shopping.

Figure 1: Some interfaces on a platform called "Hummingbird PSP"

PSPs are still at an early stage and can be improved through the scientific management of the platform, a more efficient assignment of orders to shoppers, and a rational incentive policy for shoppers. We use the incentive policy for shoppers as an example to explain the current limitation of the PSPs' operations management. In the current PSPs, the reward for shoppers is identical for all the orders, irrespective of the length of the route travelled to fulfill an order and of the shoppers' current locations. This policy is unfair for shoppers to some extent. In addition to the route length differences between orders, the differences in the numbers of lines between orders are another source of unfairness; here a line in an order denotes the information on one commodity as well as its required quantity. Some orders may be uneconomical for a shopper due to the length of travel needed to fulfill the order, the number of lines in the order, or the number of stores related to the order. In this case, some shoppers may quit the PSP due to the unfairness of the reward policy. Without a large number of available online shoppers, the PSPs cannot guarantee a high service level including a short order fulfillment time, a high response rate of orders, a relatively low delivery expense, all of which should be maintained and improved on the basis of the economies of scale. In addition, the potential delay penalty cost of assigning an order to a shopper, whose location is not well suited to the order's location, should also be taken into account when the PSPs make the assignment decisions, which could help increase customer satisfaction Finally, the PSPs should be able to decide whether or not to accept an order considering the profit it generates. These concerns motivate the design of models and algorithms to support the order assignment decisions.

This study proposes a methodology to assign orders to shoppers on a PSP. The mixed integer programming (MIP) models proposed in this study consider route lengths, rewarding rules for shoppers, shoppers' route, a penalty cost for the late delivery, and the possible rejection of orders by the PSP. Moreover, a PSP could assign more than one order to a shopper if these orders are similar with respect to either their content or the related store locations. In addition to supporting operational decisions, the models can also be used to support some strategic level decisions such as the PSP mode adoption, i.e., the decision on whether or not the PSP mode should be adopted. This study may contribute the literature through four perspectives. First, this study proposes a comprehensive set of MIP models for the newly emerging business mode. Second, some practical algorithms as well as acceleration tactics are suggested for solving the models in large-scale instances. Third, some managerial implications for practitioners are also obtained on the basis of our proposed model and algorithm. Last but not the least, this study demonstrates the possible application of the proposed methodologies in both the operational level and the strategic level decisions for this new mode.

The remainder of this paper is organized as follows. Related works are reviewed in the next section. Some MIP models are proposed in Section 3 for the basic cases on assigning one order per shopper. Section 4 extends the models by considering the assignment of multiple orders to one shopper. Based on the previously proposed operation level decision models, Section 5 presents a model-driven methodology to guide some strategic decisions for the PSPs. Conclusions follow in Section 6.

2. Related works

There exist relatively few studies directly related to the delivery mode of personal shoppers. As already mentioned, the PSP mode is a new type of e-commerce platform, and provides an alternative way for the traditional last-mile delivery in the context of online shopping. In addition, most shoppers work part-time or on a temporary basis, thus the PSP mode may also be related to the social logistics or shared mobility-based delivery. Therefore, this section mainly reviews the recent literature from three streams. The first stream is about the analytical studies for e-commerce on-demand platforms (Taylor, 2018); the second stream is about last-mile delivery for the online retailors; and the third stream is about the optimization models and their applications in social logistics or crowdsourced delivery.

From the perceptive of the on-demand service platforms, the PSPs and the ride sharing platforms are similar with each other to some extent. Plenty of studies have been conducted in the fields of the ride sharing; comprehensive reviews and past developments on ride sharing can be seen in Agatz et al. (2012)

and Tafreshian et al. (2020). Najmi et al. (2017) propose an on-demand ride sharing system with a novel objective function and dynamic matching policy. The core of the PSPs is also a matching mechanism between personal shoppers and customers. However, the customers' requirements in the PSPs are different from the ones in the ride sharing platforms. For the PSPs, a customer's requirement is a set of possible routes from a set of stores containing the customer's required goods to the customer's location; while for the ride sharing platforms, a customer's requirement is represented by a pair of one origin and one destination. For another type of on-demand service platform with endogenous supply (earning-sensitive independent providers) and endogenous demand (price-sensitive customers), Bai et al. (2019) present an analytical model. Hong et al. (2019) propose a model and an ant colony optimization based algorithm for an on-demand delivery service system. Boysen et al. (2019) study some deterministic matching problems for on-demand service platforms and provide a classification scheme for various optimization problems occurring in different areas of the sharing economy. Özkan (2020) investigate the interrelationship between the above mentioned matching decision in ride sharing platforms with the pricing decision; a more advanced mechanism on joint matching and pricing is proposed. This study mainly focuses on matching and does not consider the pricing; the adopted methodology is also different from the above one. Benjaafar et al. (2022) develop an equilibrium model to investigate how to determine a proper labor pool size and the nominal wage for maintaining agent welfare. This study mainly differs from the above studies in the background because this personal shopper industry is very new and has not been popular in most cities and countries. Although both the PSPs and some existing on-demand service platforms are based on the matching between supply and demand sides, the details on determining the matching degree in quantitative way are significantly different.

Regarding the application of operations research in last-mile delivery under some recent new business modes, Yildiz and Savelsbergh (2019a) propose a novel model formulation for the timely delivery of food orders where the food becomes available within minutes of the ordering time. The model is solved through simultaneous column and row generation. Similarly, for another food delivery problem, Liu et al. (2021) put forward a framework by integrating travel-time predictors with order-assignment optimization, which captures the drivers' routing behavior so as to improve the on-time performance of last-mile delivery services. Wang (2019) proposes an MIP model for an emerging last-mile transportation service, in which routes and schedules are optimized for a fleet of delivery vehicles with the aim of minimizing passenger waiting and riding time. Sun et al. (2018) design a branch-and-price based exact solution method for a transportation service provider considering pickup & delivery, time windows and time-dependent travel time on arcs; they further extended the above study to consider profits in problem objective (Sun et al., 2020). The above studies focus on the routing and scheduling decisions for couriers in the last-mile delivery with considering plenty of realistic factors such as uncertain random travel time; while this study focuses on the order assignment in new business mode, which is an alternative way for the traditional last-mile mode. Deng et al. (2021) apply game theory to compare two methods for improving the last-mile delivery performance: one is based on an urban consolidation center for the bundling of shipments from multiple carriers, while the other is based on a peer-to-peer platform for capacity sharing. These decisions belong to more strategical level in the last-mile delivery mode. Although majority of this study mainly belongs to the operational level decision, another strategical level decision on the PSP model adoption is also investigated in this study.

For the gradually blooming industry on the crowdsourcing delivery, Qi et al. (2018) design some analytical models as well as empirical parameter estimates for a large-scale logistics system that integrates shared mobility for home delivery services. Yildiz and Savelsbergh (2019b) consider the service coverage decision, self-selecting behavior of couriers, and hybrid delivery capacity in a crowdsourced meal delivery platform; a very comprehensive framework is proposed. However, this framework may not be applicable for this study's background on personal shoppers; in addition, the shoppers (couriers) cannot reject orders assigned by the PSPs in this study. Dayarian and Savelsbergh (2020) develop two rolling-horizon dispatching approaches for a highly dynamic and stochastic same-day delivery context, in which in-store customers supplement company drivers and deliver online orders on their way home. Driven by the concept of delivery-as-a-service, Wang et al. (2020a) investigate a new variant of last-mile delivery that integrates the scheduling of E-commerce parcels and Online-to-Offline parcels. Yildiz (2021) proposes a courier based crowdsourcing model for urban express package deliveries, and designs a dynamic programming algorithm for a real-time network based context. Fatehi and Wagner (2021) propose a robust crowdsourcing optimization model to study labor planning and pricing for crowdsourced last-mile delivery systems with guaranteed delivery time windows (e.g., same-day or two-hour deliveries). From the above, we can see that the crowdsourcing delivery is now being adopted by more and more online business modes and has attracted a lot of attention from academia and has been investigated through plenty of perspectives. This paper studies a new mode in the crowdsourcing service industry rather the traditional delivery from the decision maker's depot to the customers; out work may contribute to the literature on the crowdsourcing service industry.

Finally, some vehicle routing problem (VRP) studies should also be mentioned here. The routing decisions embedded in this paper are related to the multi-vehicle pickup and delivery problem with time windows, because a shopper needs to pick up commodities from more than one store for multiple orders and then deliver them to the customers. However, there exists only a very limited literature on this VRP variant. Naccache et al. (2018) design an adaptive large neighborhood search heuristic for this problem, in which vehicles are used to collect a set of items defined within client requests; a request contains several pickups of different items, followed by a single delivery at the client location. Aziez et al. (2020) develop an exact branch-and-cut algorithm with several families of cuts. Different from the above studies, the main decision in this study is the order assignment in the platform, several realistic factors related to this new mode must be considered.

This study contributes the literature in that it constitutes the first work to analyze PSPs which is an emerging business mode. We propose a comprehensive and practical methodology to increase profit at both the operational and strategic levels, which may potentially be useful to practitioners.

3. Model for assigning one order to each shopper

We first consider a simple but also the most commonly context in which a personal shopper is assigned to at most one order. This section contains three subsections. The first subsection presents the basic case of the one-order-to-one-shopper assignment. The second subsection extends it to a more generic case, in which a shopper in the candidate set for assignment may have an ongoing delivery order. For the two cases, models as well as algorithms are proposed in the two subsections; both of the models' objectives are to maximize the profit for the platform's profit. In the third subsection the performance of the proposed algorithms is validated through some experiments.

3.1 Basic case

This section proposes a decision framework for assigning customer orders to shoppers on a PSP. The decision framework embeds two MIP models: the first one is about the assignment decision; the second one is used to calculate a parameter used in the first model. It should be noted that the second model could be integrated into the first model so as to construct one model. However, the integrated model is too complex to be solved in a fast way, especially for some large-scale applications. Therefore, this study uses this practical way of decomposing the decision into two steps: we use the second model to calculate parameters, and then use the first model to solve the assignment decision by using the previously obtained parameters. The decision framework for the second case (in the next subsection) is also based on this decomposition way, which is more practical in realistic applications and is also easier for understand.

3.1.1 Assignment model

In the model for the assignment of orders released by customers to available shoppers, we assume that the revenue earned by fulfilling each order is a parameter, which is calculated by some confidential rules according to an agreement between a PSP and stores. The cost for a shopper to fulfill an order, which may contain multiple lines and may need the shopper to visit more than one stores, is not always the same and depends on the features of the order (e.g., customer location, locations of stores related to this order, commodities' value, weight, ease of buying) and the shopper (e.g., rank, current location). The above mentioned cost not only includes the fee paid to the shopper, which is related to the above features, but also the potential delay delivery penalty, which is related to the locations of the shopper, the customer, and the stores. It should be noted that the PSP can choose not to accept some orders during this assignment decision, the objective of which is to maximize the total profit for the PSP.

This study assumes the shoppers cannot reject the orders assigned by the PSPs and customers also cannot reject the assigned shoppers. These assumptions obey the realistic environment of the PSPs nowadays, although some other crowdsourced delivery platforms such as meal delivery from restaurants allow couriers to reject assigned orders (Yildiz and Savelsbergh, 2019b), and allow customers to reject assigned couriers due to long waiting time (Özkan, 2020). It is noted that the decision maker of the assignment model proposed here is the platform, whose objective is the maximization of its profit. The model is solved for a batch of receiving orders that are accumulated in a short time; thus the model is solved by the platform in a frequent way. When the model is being solved, we assume all the shoppers who are available at the moment are willing (ready) to receive the orders assigned by the platform, and all the orders are not canceled by the customers at the moment when the orders have already been in the batch of assigning. In the realistic PSPs, an online shopper is not allowed to reject the assigned orders; otherwise he (she) can set the status as offline. As aforementioned at the beginning of this paper, the customers on the PSPs usually need some goods urgently, and they unlikely cancel their released orders on their own initiative. For all the model formulations in this study, both shoppers and customers are not decision makers. Some behavior factors such as the shoppers' rejection decision for repositioning toward faraway hotspots are not taken into account (Urata et al. 2021). All of the above obeys the background of the assignment model and the nowadays reality of the PSP industry.

Before formulating the models, the parameters and decision variables used in the models are listed as follows.

Indices and sets:

r index of an order, the set of all orders is R.

e index of a personal shopper, the set of all shoppers is E.

Parameters:

p_r revenue for the platform if order r is fulfilled.

c_{er} cost for shopper e to fulfill order r.

Variables:

α_{er} binary, equals one if order r is assigned to shopper e, zero otherwise.

Model:

$[\boldsymbol{M\_Asgn}]$ Maximize $\sum_{r\in R}\sum_{e\in E}(p_r - c_{er})\alpha_{er}$ (1)

subject to $\sum_{e\in E}\alpha_{er} \le 1 \qquad r \in R$ (2)

$\sum_{r\in R}\alpha_{er} \le 1 \qquad e \in E$ (3)

$\alpha_{er} \in \{0,1\} \qquad r \in R, e \in E.$ (4)

Objective (1) maximizes the platform's profit resulting from accepting and fulfilling a batch of arriving orders. In the PSP mode, a customer pays the fee of buying his/her required goods to the platform. The fee is usually expensive than the goods' regular price. When a shopper has fulfilled this order, the platform pays the price of the goods to the store that provides the goods for the customer, and also pays the delivery fee to the shopper. Thus the revenue for the platform if an order is fulfilled, i.e. p_r, can be calculated by the platform before making the assignment decision. The cost parameter c_{er} for shopper e to fulfill order r is calculated by another MIP model, which is elaborated in the next subsection. This calculation process can be executed before solving $[M\_Asgn]$. Constraints (2) ensure that each order is assigned to at most one shopper since some orders may be refused by the platform. Constraints (3) guarantee that each shopper fulfills at most one order. Although a shopper can fulfill more than one order in the same trip (an assignment of one batch of orders), the one-shopper-one-order case is the most common because of the requirement of timely delivery. Constraints (4) define the domains of the variables.

As an explorative study on this new business mode (PSPs), this paper mainly aims to providing a simple but practical methodology that can support the most urgent need for running this type of platform. All the parameters' values for the model is the real-time (the latest and exact) data. For a long operation period, the model can be run in the widely used rolling horizon manner. In this case, the obtained decisions are real-time as the input data is real-time; there is no need to input some probabilities based predication (inexact) data such as the uncertain demands (arrival time, volume, goods) of customers and uncertain travel time on routes in the future.

3.1.2 The model for calculating c_{er}

This subsection details the calculation of the cost parameter c_{er} used in $[M\_Asgn]$. An order may contain multiple lines and may need a shopper to visit more than one store. The route for shopper e fulfilling order r should also be optimized in the model M_{er} which computes the cost parameter c_{er}. The parameters and variables used in M_{er} are first defined as follows:

Indices and sets:

- i index of a location of a store, shopper e 's original position, order r's shipping address.
- I_r set of all locations of stores related to order r.
- $l(e)$ shopper e 's original location.
- $\dot{l}(r)$ order r's shipping location.

Parameters:

- $d_{i,i'}$ travel cost between locations i and i'.
- t_r due time of fulfilling order r.
- f unit penalty cost of late delivery.

Variables:

- $\beta_{i,i'}$ binary, equals one if shopper e travels from location i to i', zero otherwise.
- μ_i arrival time at location i.

Model :

$$[\boldsymbol{M_{er}}]\ c_{er} = \text{Minimize } \textstyle\sum_{i,i' \in I_r \cup \{l(e)\} \cup \{\dot{l}(r)\}} d_{i,i'}\beta_{i,i'} + f \cdot (\mu_{\dot{l}(r)} - t_r)^+ \tag{5}$$

subject to

$$\textstyle\sum_{i \in I_r} \beta_{l(e),i} = \sum_{i \in I_r} \beta_{i,\dot{l}(r)} = 1 \tag{6}$$

$$\textstyle\sum_{i' \in I_r \cup \{\dot{l}(r)\}} \beta_{i,i'} = \sum_{i' \in I_r \cup \{l(e)\}} \beta_{i',i} = 1 \qquad i \in I_r \tag{7}$$

$$\mu_{i'} \geq \mu_i + d_{ii'} - M(1 - \beta_{ii'}) \qquad i \in I_r \cup \{l(e)\}, i' \in I_r \cup \{\dot{l}(r)\} \tag{8}$$

$$\beta_{i,i'} \in \{0,1\} \qquad i, i' \in I_r \tag{9}$$

$$\mu_i \geq 0 \qquad i \in I_r. \tag{10}$$

Objective (5) minimizes the cost for shopper e of fulfilling order r, which includes the route length-dependent reward paid to the shopper and the potential penalty cost of late delivery. Constraint (6) ensures that the route's origin is the shopper's current location and its destination is the order's shipping address. Constraints (7) link all the stores related to the order and form a route for the shopper. Constraints (8) connect the arrival times of two consecutive locations in the route and prevent subtours. Constraints (9) and (10) define the domains of the decision variables.

3.1.3 Criteria for reducing domain of (e, r) *pairs*

In order to speed up the solution of the assignment model, we reduce the solution space by ignoring some (e, r) pairs because of their obvious very high cost c_{er}. We define some reasonable criteria for judging whether (e, r) pairs need be calculated by the model M_{er}; the associated uncalculated c_{er} parameters are then set at a sufficiently large value. Here we consider four criteria for reducing the domains of the (e, r) pairs.

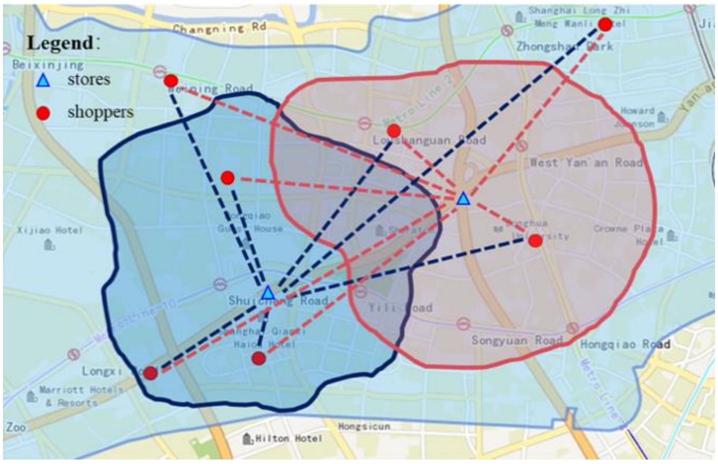


Figure 2: An example of two regions that support selecting candidate shoppers for each order

Criterion 1: Average travel cost

Step 1: For each store i related to the order r, $i \in I_r$, calculate the average travel cost between store i and all available shoppers $e \in E$, denoted by $\gamma_i = a \cdot \frac{\sum_{e \in E} d_{i,l(e)}}{|E|}, \forall i \in I_r$, where a is a positive coefficient.

Step 2: Define a region ε_i with the location (x_i, y_i) of store i as its center, the travel cost between all the points in the region and the center (x_i, y_i) is no greater than γ_i. The region ε_i is formally denoted as: $\varepsilon_i = \{(x, y) | d_{(x_i, y_i),(x,y)} \leq \gamma_i\}, \forall i \in I_r$. Figure 2 depicts two regions.

Step 3: Define the union of all regions ε_i that correspond to all the stores i related to order r; $\bar{\mathcal{E}}_r = \bigcup_{i \in I_r} \varepsilon_i$. We select the shoppers who are located in region $\bar{\mathcal{E}}_r$. As shown in Figure 2, five shoppers are selected because they are in the union of the two regions. We run the model M_{er} for each shopper in $\bar{\mathcal{E}}_r$, while for the shoppers outside $\bar{\mathcal{E}}_r$, we do not run the model M_{er} and we set the parameter c_{er} at a sufficiently large value.

It should be mentioned that If none shopper could be selected in Step 3, it implies the coefficient a is set as a too small value in Step 1. We just adjust the value of a to prevent the infeasibility in Step 3.

Criterion 2: Average travel cost and due time

The penalty of the delay delivery should also be taken into account when determining $\bar{\mathcal{E}}_r$. More specifically, the definition of ε_i is changed to $\varepsilon_i = \left\{(x,y)\middle| d_{(x_i,y_i),(x,y)} \le \gamma_i + b \cdot \frac{t_r}{\sum_{r\in R} t_r/|R|}\right\}, \forall i \in I_r$, where b is a positive coefficient. The core idea behind the above change is that if some orders are not very urgent, i.e., t_r is large, the region ε_i will also be large. Except for this modification, steps 1 and 3 are the same as for criterion 1.

Criterion 3: Average travel cost and shopper/order ratio

We first execute steps 1 and 2 of criterion 1. For each region ε_i, we calculate the total number of shoppers ($\mathbb{E}$) and the number of orders ($\mathbb{R}$) located in it. The ratio $\mathbb{E}/\mathbb{R}$ reflects the sufficient degree of the service providers for fulfilling the service requirements. The threshold for the ratio is one, which represents a balance status. If the ratio is less than one, region ε_i is changed to $\varepsilon_i = \left\{(x,y)\middle| d_{(x_i,y_i),(x,y)} \le \gamma_i + c \cdot \frac{\sum_{e\in E} d_{i,l(e)}}{|E|}\right\}, \forall i \in I_r$, where c is a positive coefficient. The core idea behind this change is that if shoppers are a relatively few in a region, that region should be extended. We then execute step 3 of criterion 1.

Criterion 4: Average travel cost, due time and shopper/order ratio

This criterion is the combination of the above three criteria. More specifically, we first execute steps 1 and 2 of criterion 2. We then modify region ε_i according to criterion 3 if the ratio $\mathbb{E}/\mathbb{R}$ is less than one. In this case, the three factors (average travel cost, due time and shopper/order ratio) are considered.

3.1.4 Dynamic programming based solution method

Dynamic programming can be used to solve the model M_{er}. Let $V = I_r \cup \{l(e), \dot{l}(r)\}$ denote the set of all store locations related to order r, which also includes initial location of express shopper e and shipping location of order r. The set V' is a subset of V. We define $C(i, V)$ and μ_i respectively as the minimum total cost and the corresponding time spent to reach i via all locations in V. The optimal travel route for a shopper fulfilling one single order can be obtained through the following recursive equations:

Initial state function values: $C(i, \{l(e)\}) = d_{i,l(e)}$, $\forall i \in I_r$.

State transition equation: $C(i, V') = \min_{k\in V'}\{d_{i,k} + C(k, V'\backslash\{k\})\}$, $\forall i \in I_r, i \notin V'$.

Objective state function values: $C\left(\dot{l}(r), V\backslash\{\dot{l}(r)\}\right) = \min_{V\backslash\{\dot{l}(r)\}}\{d_{i,k} + C\left(k, V\backslash\{\dot{l}(r), k\}\right) + f \cdot (\mu_i - t_r)\}$

Dynamic programming is used to solve model M_{er} for all the (e, r) pairs in the reduced domain as explained in Section 3.1.3. This method is much more efficient than applying CPLEX to solve M_{er}, as shown by our computational experiments in Section 3.3. Given all the c_{er} values obtained by the model M_{er} as well as the dynamic programming algorithm, we use CPLEX to solve the assignment model M_Assgn of Section 3.1. Since CPLEX can solve M_Assgn efficiently, there is no point in designing an elaborate algorithm for it.

3.2 Case of shoppers with ongoing deliveries

This section extends the previous model to the case where a shopper who fulfills an order can also be assigned newly coming orders. This implies that the set of candidate shoppers contains not only the idle shoppers, but also those with ongoing deliveries. This context is reasonable because the newly assigned order for a shopper may include stores that are near to its current location, or the newly assigned order's shipping location is near to its ongoing order's shipping location. The objective is still the maximization of the profit for the platform during the assignment of a batch of incoming orders. The larger is the set of candidate shoppers, the better performance can be achieved from a theoretical perspective.

3.2.1 Assignment model

Although the objective is still profit maximization, the assignment model needs to be updated. More specifically, it should exclude the cost of fulfilling an ongoing order's remainder part, because this part is determined in the decision of assigning the previous batch of orders.

Newly added parameters:

g_e cost of fulfilling remainder part of ongoing order undertaken by shopper e; it includes the potential penalty cost of late delivery for the ongoing order.

Extended model:

$[M\_Asgn^E]$ Maximize $\sum_{r\in R}\sum_{e\in E}\big(p_r-(c_{er}-g_e)\big)\alpha_{er}$ (11)

subject to $\sum_{e\in E}\alpha_{er}\le 1$ $\quad r\in R$ (12)

$\sum_{r\in R}\alpha_{er}\le 1$ $\quad e\in E$ (13)

$\alpha_{er}\in\{0,1\}$ $\quad r\in R, e\in E.$ (14)

Objective (11) maximizes the platform's profit derived from fulfilling a batch of arriving orders, which is essentially the same as the objective of the first assignment model proposed in Section 3.1. It should be noted that all the proposed decision models for supporting the decision of the PSP aim to maximizing the platform's final profit. In Objective (11), the cost parameter c_{er} for an idle shopper e to fulfill order r is calculated by the previously defined MIP model M_{er}; the parameter c_{er} for a shopper e with ongoing order to fulfill order r is calculated by another MIP model M_{er}^E, which is elaborated in the next subsection. Constraints (12)–(14) are the same as in the previously defined assignment model.

3.2.2 The model of calculating c_{er} for ongoing shoppers

Newly defined indices and sets:

$\tilde{r}$ index of the ongoing order undertaken by shopper e.

$I_{\tilde{r}}$ set of locations of unvisited stores related to order $\tilde{r}$.

I' $I_{\tilde{r}}\cup I_r$.

p dummy point of termination.

Extended model:

$[M_{er}^E]$ c_{er} = Minimize $\sum_{i\in I'\cup\{l(e)\},i'\in I'\cup\{\hat{l}(r),\hat{l}(\tilde{r})\}} d_{ii'}\beta_{ii'}+f\cdot\left[(\mu_{\hat{l}(r)}-t_{\tilde{r}})^{+}+(\mu_{\hat{l}(\tilde{r})}-t_{\tilde{r}})^{+}\right]$ (15)

subject to $\sum_{i\in I'\cup\{\dot{l}(r),\dot{l}(\tilde{r})\}}\beta_{l(e)i} = \sum_{i\in I'\cup\{\dot{l}(r),\dot{l}(\tilde{r})\}}\beta_{ip} = 1$ (16)

$$\sum_{i'\in I'\cup\{\dot{l}(r),\dot{l}(\tilde{r}),p\}}\beta_{ii'} = \sum_{i'\in I'\cup\{\dot{l}(r),\dot{l}(\tilde{r}),\{l(e)\}}\beta_{i'i} = 1 \qquad i\in I'\cup\{\dot{l}(r),\dot{l}(\tilde{r})\} \tag{17}$$

$$\mu_{\dot{l}(r)} \geq \mu_i + 1 \qquad \forall i\in I_r \tag{18}$$

$$\mu_{\dot{l}(\tilde{r})} \geq \mu_i + 1 \qquad \forall i\in I_{\tilde{r}} \tag{19}$$

$$\mu_{i'} \geq \mu_i + d_{ii'} - M(1-\beta_{ii'}) \qquad i\in I'\cup\{\dot{l}(r),\dot{l}(\tilde{r}),\{l(e)\},i'\in I'\cup\{\dot{l}(r),\dot{l}(\tilde{r}),p\} \tag{20}$$

$$\beta_{ii'}\in\{0,1\} \qquad i\in I'\cup\{\dot{l}(r),\dot{l}(\tilde{r}),\{l(e)\},i'\in I'\cup\{\dot{l}(r),\dot{l}(\tilde{r}),p\} \tag{21}$$

$$\mu_i \geq 0 \qquad i\in I'\cup\{\dot{l}(r),\dot{l}(\tilde{r}),l(e),p\}. \tag{22}$$

Objective (15) minimizes the cost for shopper e of fulfilling order r, which also includes the penalty for delayed deliveries. Constraint (16) ensures that the beginning and end of each shopper's route origin and destination are its current location and the dummy point of termination, respectively. Constraints (17) guarantee that all store locations and shipping locations related to the ongoing order $\tilde{r}$ and the assigned order r are visited, and these locations are connected sequentially. Constraints (18) ensure that the shipping location of order r can be visited when all the stores related to that order have been visited. Similarly, constraints (19) mean that the shipping location of order $\tilde{r}$ is visited when all the stores related to that order have been visited. Constraints (20) connect the arrival time of two consecutive locations in the route and prevent subtours. Constraints (21) and (22) define the domains of the decision variables.

3.2.3 Dynamic programming based solution method

The extended model M_{er}^{E} is also solved by dynamic programming. Let $\tilde{V} = I'\cup\{l(e),\dot{l}(\tilde{r}),\dot{l}(r),p\}$ denote the set of all store locations related to the orders r and $\tilde{r}$, the initial location of shopper e, the shipping locations of the two orders, and a dummy location representing end point of route. The set $\tilde{V}'$ is a subset of $\tilde{V}$. The optimal route of a shopper, who may be in the process of executing a delivery order performed, to complete both the ongoing order and an additional order is calculated by the following recursive equations:

Initial state function values: $C(i,\{l(e)\}) = d_{i,l(e)},\ \forall i\in I_{\tilde{r}}\cup I_r$.

State transition equation:

$$C(i,\tilde{V}') = \begin{cases} \min_{k\in\tilde{V}'}\{d_{i,k} + C(k,\tilde{V}'\backslash\{k\}) + f\cdot(\mu_i - t_i)^+\} & \forall i\in\{\dot{l}(\tilde{r}),\dot{l}(r)\}, i\notin\tilde{V}', I_i\subseteq\tilde{V}' \\ \min_{k\in\tilde{V}'}\{d_{p,k} + C(k,\tilde{V}'\backslash\{k\})\} & \forall i\in I', i\notin V' \\ \infty & \text{otherwise} \end{cases}$$

Objective state function values: $C(p,\tilde{V}\backslash\{p\}) = \min_{k\in\tilde{V}\backslash\{p\}}\{d_{p,k} + C(k,\tilde{V}\backslash\{p,k\})\}$.

After applying dynamic programming to solve the model M_{er}^{E} for all the (e,r) pairs in a reduced domain, we use the CPLEX to solve the assignment model M_Asgn^{E} directly.

3.3 Algorithmic performance

We have conducted several numerical experiments to assess the performance of the proposed algorithm for the basic case and the extended case.

3.3.1 Instance setting

Experiments consider a selected list of store locations in the Changning district, which has a population of 0.7 million, an area of 38 square kilometers. The PSP mode is more suitable for the urban areas than the suburban areas. Among the seven urban districts in Shanghai (i.e., the most modern city and the economic center in China), the Changning district may be the most representative because it ranks at the median to upper level with respect to some criteria such as the residents' income per capita, the GDP per capita, the GDP per area, district area, population. It may also be the most balanced district in the aspect of merging CBDs and residential areas. Therefore, we choose the Changning district as the background of case study in this paper. Figure 3 depicts the locations of about one hundred stores distributed in this district. All experiments in this paper were implemented and performed on a workstation with two Xeon E5-2643 V4 CPUs (12 cores) with a 3.4 GHz processing speed and 256 GB of memory running Windows 10. The models were implemented with CPLEX 12.6.1 in concert with C# (VS2019).

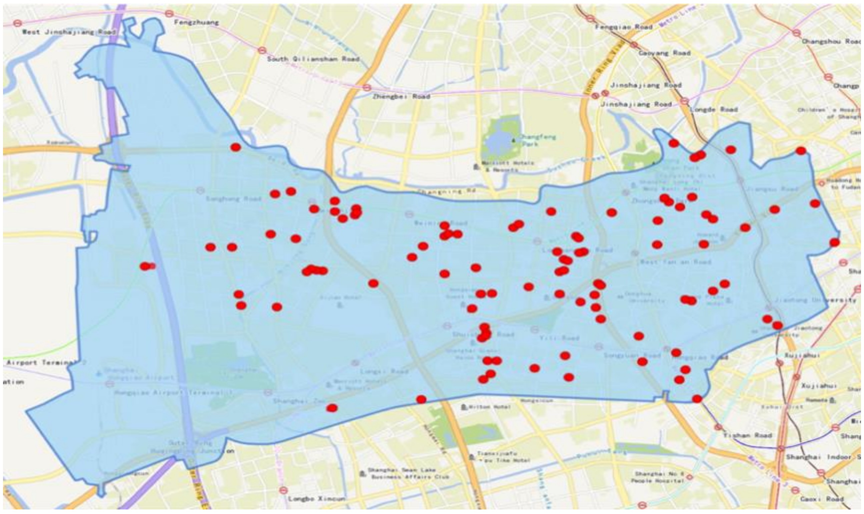


Figure 3: Stores on a PSP in the Changning district, Shanghai, China

Table 1 lists the experiment setting ($|R|$ the number of orders, and $|E|$ the number of available shoppers during the planning horizon) for the six groups of problem instances used in this study. In this study, the planning horizon is set at 15 minutes, which means that the proposed models are run every 15 minutes for a batch of arriving orders; it also imposes an implicit requirement on the maximum computation time for the proposed models, which should not exceed 15 minutes; otherwise the proposed methodology is impractical. In addition, the commodities in each order are related to at most three stores in the numerical experiments. The coefficients a, b, c used in the proposed criteria of Section 3.1.3 are set equal to 0.4, 1.2, 0.3, respectively.

Table 1: Experimental setting and problem scales of six instance groups

	Order quantity	Shopper quantity	Basic case	Extended case

		$\|R\|$	$\|E\|$	Num. of var.	Num. of cons.	Num. of var.	Num. of cons.
Small scale	ISG1	25	35	10^3	10^3	10^3	10^3
	ISG2	50	70	10^4	10^4	10^4	10^4
	ISG3	100	120	10^5	10^5	10^5	10^5
Large scale	ISG4	600	650	10^6	10^6	10^6	10^6
	ISG5	800	850	10^6	10^6	10^6	10^6
	ISG6	1000	1050	10^7	10^7	10^7	10^7

Notes: (1) The basic case, elaborated in Section 3.1, does not consider candidate shoppers who may undertake ongoing delivery tasks; however, the extended case, elaborated in Section 3.2, considers candidate shoppers who may undertake ongoing delivery tasks. (2) Num. of var. and Num. of cons. denote the approximate number of variables and constraints contained in the MIP models in the two cases.

3.3.2 Comparison with optimal results in small-scale instances

Some comparative experiments are conducted on small-scale instances. To reduce the domain of (e,r) pairs, five options are considered as "universal", "criterion 1", "criterion 2", "criterion 3", "criterion 4" shown in the second column of Table 2. "Universal" means that all (e,r) pairs are considered; the c_{er} parameters for all the (e,r) pairs are calculated by solving the M_{er} model; the set of (e,r) pairs is the universal set without reduction. "Criterion X" means that the set domain of the (e,r) pairs has been reduced according to criterion X; then we need not solve the M_{er} model all the possible (e,r) pairs. For the solution method of the M_{er} model, two options are considered: one is CPLEX and the other is dynamic programming. Thus ten results are obtained for each instance with respect to the objective value and the computation time. From results in Table 2, four conclusions emerge:

(1) It is very efficient to use the dynamic programming to solve the M_{er} model. The computation time (t'_{dp}) is much shorter than the time of CPLEX (t'_{cplex}), and the results obtained by the two methods are the same for each row ($obj_{cplex} = obj_{algorithm}$).

(2) In the time needed to compute the assignment decisions (t_{cplex} or $t_{algorithm}$), the time (t'_{cplex} or t'_{dp}) needed to solve the M_{er} model constitutes the major part; the ratios t'_{cplex}/t_{cplex} and $t'_{dp}/t_{algorithm}$ approach one.

(3) The four criteria are very effective for reducing the computation time (the t_{cplex} or $t_{algorithm}$ for one criterion is always shorter than the time for universal). The reduction effect for $t_{algorithm}$ is much more significant than for t_{cplex}, which implies the combination of the dynamic programming and the criteria applied to reduce the domain of (e,r) pairs is a very effective way of solving this problem.

(4) In addition to reducing computation time effectively, the performance of the four reduction criteria is also beneficial with respect to solution quality. The column gap shows most solutions are as good as those obtained without reduction. All cases corresponding to criterion 4 have zero gaps, which validates the better performance of this criterion with respect to the other criteria. In the following experiments, we adopt criterion 4.

Table 2: Algorithmic performance of the basic model in small-scale instances

Instance groups	Domain of (e, r) pairs	obj_{cplex}	$obj_{algorithm}$	gap	t_{cplex}	t'_{cplex}	$t_{algorithm}$	t'_{dp}
ISG1	Universal	201.28	201.28	——	34.34	34.31	0.24	0.21
	Criterion 1	197.50	197.50	1.88%	5.99	5.95	0.07	0.03
	Criterion 2	201.00	201.00	0.14%	6.75	6.73	0.06	0.04
	Criterion 3	197.78	197.78	1.74%	13.38	13.34	0.09	0.05
	Criterion 4	201.28	201.28	0.00%	12.98	12.92	0.13	0.07
ISG2	Universal	401.99	401.99	——	133.50	133.45	0.22	0.17
	Criterion 1	396.68	396.68	1.32%	24.68	24.65	0.06	0.03
	Criterion 2	401.85	401.85	0.03%	27.86	27.83	0.06	0.03
	Criterion 3	396.82	396.82	1.29%	55.48	55.45	0.09	0.06
	Criterion 4	401.99	401.99	0.00%	56.78	56.75	0.10	0.07
ISG3	Universal	784.62	784.62	——	465.43	465.33	0.82	0.72
	Criterion 1	781.33	781.33	0.42%	76.55	76.46	0.22	0.13
	Criterion 2	784.62	784.62	0.00%	86.40	86.33	0.19	0.12
	Criterion 3	784.62	784.62	0.00%	166.37	166.30	0.29	0.22
	Criterion 4	784.62	784.62	0.00%	164.23	164.16	0.31	0.24

Notes: (1) obj_{cplex} and t_{cplex} denote the objective value and computation time when using CPLEX to solve the M_{er} model and the M_Asgn model. (2) $obj_{algorithm}$ and $t_{algorithm}$ denote the objective value and computation time when using DP to solve the M_{er} model and using the CPLEX to solve M_Asgn model. (3) As part of the total duration needed to solve the different types of models by two ways, t'_{cplex} is the time needed to solve M_{er} by CPLEX, while t'_{dp} is the time needed to solve M_{er} by the DP. (4) gap is the relative gap between the objective obtained by the way of "universal" and the objective obtained by applying "criterion X".

Section 3.2 proposes an extended model considering the potential ongoing tasks for some shoppers. For this model, another series of comparative experiments are conducted on small-scale instances. The results are shown in Table 3. Since we consider the ongoing tasks at the moment of making the assignment decision, the second column in Table 3 lists four scenarios with different percentages of shoppers who are undertaking some ongoing tasks. In the four scenarios of each instance group ISG1, ISG2 or ISG3, we select 5%, 10%, 15% and 20% of the shoppers from all the available shoppers and set their status as being fulfilling some ongoing tasks. The setting for other parameters is the same as the data in Table 1. The results in Table 3 demonstrate that our proposed algorithm yields optimal result on all instances, while the computation time is much shorter than that of CPLEX. Another finding is that when the percentage of shoppers with ongoing tasks is large, the computation time does not seem to increase for the small-scale instances.

Table 3: Algorithmic performance of the extended case on small-scale instances

Instance groups	Percentage of shoppers with ongoing tasks	obj_{cplex}	$obj_{algorithm}$	gap	t_{cplex}	t'_{cplex}	$t_{algorithm}$	t'_{dp}
ISG1	5%	131.59	131.59	0.00%	34.18	34.17	0.03	0.02
	10%	181.96	181.96	0.00%	37.82	37.81	0.03	0.02
	15%	157.88	157.88	0.00%	44.29	44.27	0.04	0.02
	20%	174.39	174.39	0.00%	54.76	54.74	0.04	0.02

ISG2	5%	424.57	424.57	0.00%	154.33	154.30	0.13	0.10
	10%	397.33	397.33	0.00%	185.39	185.35	0.12	0.08
	15%	436.87	436.87	0.00%	172.23	172.20	0.12	0.09
	20%	475.61	475.62	0.00%	197.49	197.46	0.13	0.10
ISG3	5%	850.02	850.02	0.00%	517.20	517.11	0.37	0.28
	10%	738.47	738.47	0.00%	701.34	701.27	0.36	0.29
	15%	788.91	788.91	0.00%	597.77	597.70	0.39	0.32
	20%	861.60	861.60	0.00%	828.26	828.17	0.44	0.35

3.3.3 Performance of the extended case in large-scale instances

Besides the previous small-scale instances, some experiments are also conducted on large-scale instances. Since the extended case includes the basic case, we only show the experiments performed on the extended case. Because CPLEX cannot solve the model M_{er}^{E} in the large-scale instances, we implement a practical three-step decision rule as the reference point in the experiments conducted to assess the relative advantage of our methodology. It is described as follows.

Step 1: Determine the priority sequence of orders according to the orders' revenue p_r (primary criterion) and the orders' due time t_r (secondary criterion). It is practical to set an order with high priority if it has high revenue and an urgent due time.

Step 2: Assign the shopper with the largest matching degree to each order one at a time according to the orders' priority sequence. For each pair of a shopper and order, the matching degree is determined as follows. We assume that a shopper fulfills an order according to the following sequence: the shopper's original location $l(e)$→the farthest store s_1 from the order's destination→the nearest store s_2 from store s_1→the nearest store s_3 from store s_2→⋯→the order's destination $\dot{l}(r)$. We use an example with three shoppers e_1, e_2, e_3, one order r, two related stores s_1 and s_2 to explain how to determine the matching degree between a shopper (e_1, e_2, or e_3) and the order r. If shopper e_1 is idle, the matching degree is set as $d_{l(e),s_1}$, i.e., the distance from the shopper's current location to the farthest store. If shopper e_2 is undertaking order $\tilde{r}_2$ and the unvisited stores related to $\tilde{r}_2$ do not contain stores s_1 and s_2, the matching degree is set as $d_{\dot{l}(\tilde{r}_2),s_1}$, i.e., the distance from the shopper's previous order destination to the farthest store. If shopper e_3 is undertaking order $\tilde{r}_3$ and the next visiting store is the store s_2, the matching degree is set as $d_{\dot{l}(\tilde{r}_3),s_1} - (d_{s_1,s_2} + d_{s_2,\dot{l}(r)} - d_{s_1,\dot{l}(r)})$; in the previous formula, the part "$d_{s_1,s_2} + d_{s_2,\dot{l}(r)} - d_{s_1,\dot{l}(r)}$" is reduced because store s_2 need not be visited again when fulfilling the order r, then some travel distance could be saved.

Step 3: Calculate the profit of the PSP, i.e., the objective of the original model, according to the cost of the above determined shoppers' routes and fulfilled orders' profit.

Table 4 provides comparative results between the plan obtained by our proposed algorithm and the plan obtained by the above decision rule. Our main findings are as follows:

(1) The proposed algorithm can obtain better plans than the decision rule. The relative gap between them is about 13.72% with respect to the objective value.

(2) For the large-scale instances with about one thousand orders and shoppers, our proposed algorithm obtains a good assignment plan in about half a minute, which is very fast for an MIP model with about 10 million variables (mostly integer) and about 10 million constraints.

(3) The influence of the percentage of shoppers with ongoing tasks on the final result is demonstrated on the large-scale instances. The larger is the percentage, the longer is the computation time (i.e., the $t_{algorithm}$ column in Table 4) and the more significant is the relative advantage of the proposed algorithm (i.e., the gap column).

Table 4: Performance of the extended case in large-scale instances

Instance groups	Percentage of shoppers with ongoing tasks	obj_{model}	obj_{rule}	gap	$t_{algorithm}$	t'_{dp}
ISG4	5%	5384.34	4999.13	7.15%	11.10	8.64
	10%	5374.14	4696.62	12.61%	11.67	9.23
	15%	5863.76	4941.71	15.72%	11.97	9.70
	20%	5709.09	4566.42	20.01%	12.59	10.28
ISG5	5%	7009.38	6588.74	6.00%	19.46	15.26
	10%	7505.71	6578.65	12.35%	20.44	16.22
	15%	7652.60	6407.94	16.26%	20.66	16.26
	20%	7709.70	6244.46	19.01%	23.09	18.27
ISG6	5%	8864.84	8159.09	7.96%	29.68	22.61
	10%	9542.26	8424.02	11.72%	33.54	25.32
	15%	9965.83	8475.23	14.96%	34.26	26.38
	20%	10291.08	8144.05	20.86%	36.11	28.40
			Average:	13.72%		

4. Model for assigning multiple orders to each shopper

In reality, a shopper is usually assigned more than one order. This section further extends the models developed in Sections 3.1 and 3.2 to a more generic context, in which a shopper can be assigned multiple orders.

4.1 Revising model M_{er} for a dummy order with multiple shipping locations

Based on the methodology proposed in Section 3, the order in the assignment model can be regarded a dummy order with one, two, or more realistic orders. Thus the key issue is to adapt the original model M_{er} to the context in which an order r is the combination of multiple actual orders.

Suppose the upper limit of orders for each shopper is N, then the number of the dummy orders in the assignment model is $C_{|R|}^1 + C_{|R|}^2 + \cdots + C_{|R|}^N$, here $|R|$ is the total number of all the actual orders.

4.1.1 Assignment model

Newly defined indices and sets:

o index of a dummy order, the set of all dummy orders is O.

R_o set of all orders included in dummy order $o \in O$.

Newly added parameters:

c_{eo} cost for shopper o to fulfill dummy order o.

p_o revenue for the platform if dummy order o is fulfilled.

$y_{o,r}$ equals one if order r is included in dummy order o, otherwise zero.

Newly added variables:

δ_{eo} binary, equals one if dummy order o is assigned to shopper e, zero otherwise.

$$[\boldsymbol{M\_Asgn\_Mtpl}] \text{ Maximize } \sum_{o\in O}\sum_{e\in E}(p_o - c_{eo})\delta_{eo} \tag{23}$$

subject to

$$\sum_{e\in E}\sum_{o\in O} y_{o,r}\delta_{eo} \leq 1 \qquad r \in R \tag{24}$$

$$\sum_{o\in O}\delta_{eo} \leq 1 \qquad e \in E \tag{25}$$

$$\delta_{eo} \in \{0,1\} \qquad e \in E, o \in O. \tag{26}$$

Objective (23) maximizes the platform's profit derived from fulfilling a batch of arriving orders. The cost parameter c_{eo} for an idle shopper e to fulfill dummy order o is calculated by an MIP modelM^E_{eo}, which is defined in the next subsection. Constraints (24) ensure that each dummy order is assigned to at most one shopper. Constraints (25) guarantee that each shopper fulfills at most one dummy order. Constraints (26) define the domains of the decision variables.

4.1.2 Model for calculating c_{eo} for shoppers performing dummy orders

Indices and sets:

$\dot{l}(r)$ index of order r's shipping location, $r \in R_o$.

$\dot{L}$ set of locations of the 'actual' orders included in dummy order o.

I'' set of stores related to all the 'actual' orders, which are included in the dummy order o.

$$[\boldsymbol{M^E_{eo}}]\ c_{eo} = \text{ Minimize } \sum_{i,i'\in I''\cup\dot{L}\cup\{l(e),p\}} d_{ii'}\beta_{ii'} + f\cdot\sum_{\dot{l}(r)\in\dot{L}}(\mu_{\dot{l}(r)} - t_r)^+ \tag{27}$$

subject to

$$\sum_{i\in I''\cup\dot{L}}\beta_{l(e)i} = \sum_{i\in I''\cup\dot{L}}\beta_{ip} = 1 \tag{28}$$

$$\sum_{i'\in I''\cup\dot{L}\cup\{p\}}\beta_{ii'} = \sum_{i'\in I''\cup\dot{L}\cup\{l(e)\}}\beta_{i'i} = 1 \qquad i \in I''\cup\dot{L} \tag{29}$$

$$\mu_{\dot{l}(r)} \geq \mu_i \qquad \forall i \in I_r, \dot{l}(r) \in \dot{L}, r \in R_x \tag{30}$$

$$\mu_{i'} \geq \mu_i + d_{ii'} - M(1-\beta_{ii'}) \qquad i \in I''\cup\dot{L}\cup\{l(e)\}, i' \in I''\cup\dot{L}\cup\{p\} \tag{31}$$

$$\beta_{ii'} \in \{0,1\} \qquad i \in I''\cup\dot{L}\cup\{l(e)\}, i' \in I''\cup\dot{L}\cup\{p\} \tag{32}$$

$$\mu_i \geq 0 \qquad i \in I''\cup\dot{L}\cup\{l(e)\}\cup\{p\}. \tag{33}$$

Objective (27) minimizes the cost of shopper e fulfilling dummy order o, which includes the penalty for delayed deliveries. Constraint (28) ensures that the origin and destination of each shopper's route are its current location and the dummy point of termination, respectively. Constraints (29) guarantee that all the stores' locations and shipping locations related to the order $r \in R_o$, and these locations are connected sequentially. Constraints (30) ensure that the shipping location of order $r \in R_o$ can be visited when all the stores related to the order have been visited. Constraints (31) connect the arrival time of two consecutive locations in the route and avoid the potential subtours. Constraints (32) and (33) define the domains of the decision variables.

4.1.3 Dynamic programming based solution method

The model M_{eo}^{E} is also solved by the dynamic programming. Let $\bar{V} = I'' \cup \{\hat{l}(r), r \in R_o\} \cup \{l(e), p\}$ denote the set of all stores' locations related to realistic order r included in dummy order, shipping locations of actual orders, initial location of shopper e and a dummy location representing end point of route. The set $\bar{V}'$ is a subset of $\bar{V}$. The optimal travel route for a shopper performing multiple orders can be obtained through the following recursive equations:

Initial state function values: $C(i, \{l(e)\}) = d_{i,l(e)},\ \forall i \in I''$.

State transition equation:

$$C(i, \tilde{V}') = \begin{cases} \min_{k \in \tilde{V}'}\{d_{i,k} + C(k, \tilde{V}' \backslash \{k\}) + f \cdot (\mu_i - t_i)^+\} & \forall i \in \{\hat{l}(r), r \in R_o\}, i \notin \tilde{V}', I_i \subseteq \tilde{V}' \\ \min_{k \in \tilde{V}'}\{d_{p,k} + C(k, \tilde{V}' \backslash \{k\})\} & \forall i \in I'', i \notin \tilde{V}' \\ \infty & \text{otherwise} \end{cases}$$

Objective state function values: $C(p, \bar{V} \backslash \{p\}) = \min_{k \in \bar{V} \backslash \{p\}}\{d_{p,k} + C(k, \bar{V} \backslash \{p, k\})\}$.

As the number of the locations is $|\bar{V}|$, the time complexity and space complexity of this algorithm are $O(|\bar{V}|^2 \times 2^{|\bar{V}|})$ and $O(|\bar{V}| \times 2^{|\bar{V}|})$, respectively. When the value of $|\bar{V}|$ grows, the computation time will increase significantly, and the computer may run out of memory. According to our test, the above traditional dynamic programming algorithm is probably suitable for solving the problem instances, in which the number of locations is less than about twenty. For further increasing the solvable problem scale, an improved algorithm is proposed in the next subsection.

4.1.4 An improved algorithm

Based on the above traditional dynamic programming algorithm, this subsection proposes an improved algorithm for solving large-scale problem instances. Different from the algorithm in the previous subsection, the improved algorithm uses a new concept "sub-sequence of locations". The final solution to our problem, i.e., a sequence of locations, is a composition of some sub-sequences. Given a set of sub-sequences, we obtain the optimal order of the sub-sequences by using the dynamic programming; then the set of sub-sequences is improved iteratively so as to obtain satisfying sequence of locations for visit. By using the above idea, the improved algorithm could solve some large-scale instances with more locations.

It should be noted that approximate dynamic programming (ADP) algorithm is widely used to solve some large-scale problems in recent years; the ADP uses approximated value function to replace the value function in the Bellman's equation. Some excellent works have been conducted to use ADP to solve some vehicle routing problems such as He et al. (2018), Novoa et al. (2009), Ulmer et al. (2019), which use rollout algorithm, offline–online policy, neural network to accelerate the computation of value functions, respectively. However, we implemented the ADP to solve our problem and found the ADP may not be suitable for the problem, because the transition of states is uncertain in the ADP while the transition in our problem is deterministic. Thus some tactics used in the ADP could not be used in our algorithm to approximate the computation of value functions so that the solving process is accelerated. Some excellent works also have proposed some new tactics in the dynamic programming for solving scheduling problems such as Wang et al. (2020b), Yıldırım and Yıldız (2021). After implementing these tactics for our problem, we found that the nonlinear features considered in the dynamic programming designed by Wang et al. (2020b) may be unnecessary for our problem, the forward dynamic programming used in Yıldırım and Yıldız (2021) used longer computation time than our backward dynamic programming used in this study. Based on the above analysis and attempts, we use another idea addressed in the previous paragraph to improve the solving efficiency of the previously proposed algorithm. The detailed flow of the improved algorithm is as follows.

Step 1: Obtain an initial route, i.e., sequence of locations, by using a greedy heuristic. The heuristic starts from an empty route; the shopper e's original location $l(e)$ is set as the current node; we select the location (e.g., location i) that is the closest to the current node and add it into the route; the location i is set as the current node; the above procedure is repeated until the termination point p becomes the current node. Then an initial route is constructed.

The route is constructed on the basis of the set of locations, i.e., $\bar{V} = I'' \cup \{\dot{l}(r), r \in R_o\} \cup \{l(e), p\}$. For example, the initial route could be $l(e) \rightarrow s_1 \rightarrow s_3 \rightarrow s_2 \rightarrow s_5 \rightarrow \dot{l}(r_1) \rightarrow s_4 \rightarrow s_6 \rightarrow \dot{l}(r_2) \rightarrow p$; here $R_o = \{r_1, r_2\}, I_{r_1} = \{s_1, s_2, s_3\}, I_{r_2} = \{s_4, s_5, s_6\}$.

Step 2: Divide the route into m sub-sequences; here $1 \leq m \leq |\bar{V}|$. First, we set shopper e's original location, each order r's shipping location, and dummy point of termination as sub-sequences, i.e., $l(e), \dot{l}(r_1), \dot{l}(r_2), p$, the four sub-sequences in the above example. Then we divide the remainder route "$s_1 \rightarrow s_3 \rightarrow s_2 \rightarrow s_5, s_4 \rightarrow s_6$" in a random way; for example, three more sub-sequences are obtained, which are denoted by a set $S = \{s_1 \rightarrow s_3, s_2 \rightarrow s_5, s_4 \rightarrow s_6\}$. The set of sub-sequences is denoted by $\hat{V}$, $\hat{V} = \{l(e)\} \cup \{\dot{l}(r), r \in R_o\} \cup \{p\} \cup S$. In the above example, the route is divided into seven sub-sequences.

Step 3: Use the dynamic programming to the optimal order of the m sub-sequences. Here some notations are newly defined. h_i denotes the direction of the i^{th} sub-sequence; it equals one for forward direction (from the front to the end), and equals zero for backward direction (from the end to the front). $P(i, h_i)$ denotes the last location in the i^{th} sub-sequence according to the direction specified by h_i. $\hat{V}'$ is defined as a subset of $\hat{V}$, the set of sub-sequences. $I_{\hat{V}'}$ is the set of all the locations in the sub-sequences, which are contained in the set $\hat{V}'$. A traditional dynamic programming is adopted to obtain the optimal order of the sub-sequences; some important equations used in the dynamic programming are as follows.

Initial state function values: $C(i, h_i, \{l(e)\}) = d_{P(i,h_i),l(e)}$, $\forall i \in S$.

State transition equation: $C(i, h_i, \hat{V}')$

$$= \begin{cases} \min_{k \in \hat{V}'} \{d_{i,k} + C(k, h_i, \hat{V}' \backslash \{k\}) + f \cdot (\mu_i - t_i)^+\} & \forall i \in \{\dot{l}(r), r \in R_o\}, i \notin \hat{V}', I_i \subseteq I_{\hat{V}'} \\ \min_{k \in \hat{V}'} \{d_{i,k} + C(k, h_i, \hat{V}' \backslash \{k\})\} & \forall i \in I'', i \notin \hat{V}' \\ \infty & \text{otherwise} \end{cases}.$$

Objective state function values: $C(p, h_p, \hat{V}' \backslash \{p\}) = min_{k \in \hat{V} \backslash \{p\}} \{d_{p,k} + C(k, \hat{V}' \backslash \{p, k\})\}$.

Step 4: Construct sequence of locations (i.e., a new route) according to the above obtained optimal order of the sub-sequences, which is based on $C(p, h_p, \hat{V}'\backslash\{p\})$.

Step 5: If the best solved solution has not been improved for a certain number of consecutive iterations, the algorithm is terminated and outputs the route that is the best so far; otherwise the algorithm goes to Step 2 with the above obtained new route.

The above two algorithms proposed in subsections 4.1.3 and this 4.1.4 could act as two alternative ways to solve the model M_{eo}^{E}. More specifically, for all the (e, o) pairs in a reduced domain, if $|\bar{V}|$ is relatively small, we adopt the dynamic programming elaborated in subsection 4.1.3; otherwise, the improved algorithm elaborated in this subsection could be adopted.

4.2 Criterion for reducing domain of dummy orders

In order to improve the decision efficiency of the proposed methodology, we use a criterion to reduce the domain of the dummy orders. The core idea behind the criterion is to select orders that contain similar commodities, near located stores and customers' shipping address, and combine them as one dummy order. Then we do not need to enumerate all possible combinations of orders. We should avoid the combination of some orders with related stores and customers that are far away from each other. More specifically, for an order r, we define Ω_r as the set of orders that can be combined with the order r. Three conditions are used to judge whether an order r' should be included in the set Ω_r:

(1) If order r' is in region $\bar{\mathcal{E}}_r$ defined in step 3 of criterion 1 (in Section 3.1.3), it is included in Ω_r.

(2) Define a circular region ζ_i with store location (x_i, y_i) as center and $\max\{\tau_r, \tau_{r'}\}$ as radius; that is $\zeta_i = \{(x, y) | d_{(x_i, y_i),(x,y)} \leq \max\{\tau_r, \tau_{r'}\}\}, \forall i \in I_r$, here $\tau_r = \frac{t_r \cdot v}{d + e \cdot |I_r|}$, v is the speed of a shopper, d and e are coefficients. Then, define $\bar{Z}_r$ is the union of the regions ζ_i for all the stores related to order r, i.e., $\bar{Z}_r = \cup_{i \in I_r} \zeta_i$. If order r' is in $\bar{Z}_r$, it is included in Ω_r.

(3) Define a circular region η_r with order's shipping location (x_r, y_r) as center and $\max\{\tau_r, \tau_{r'}\}$ as radius; that is $\eta_r = \{(x, y) | d_{(x_r, y_r),(x,y)} \leq d \cdot \max\{\tau_r, \tau_{r'}\}\}$. If order r' is in η_r, it is included in Ω_r.

It should be noted that for two orders r and r', only when $r \in \Omega_{r'}$ and $r' \in \Omega_r$, these two orders could be combined into a dummy order.

4.3 Algorithmic performance

Several numerical experiments are conducted in this section to validate the performance of the proposed algorithms for the context in which each personal shopper can be assigned with multiple orders. Since one shopper can fulfill more than one order, the instance setting is adjusted so that the number of orders is a bit more than the number of shoppers. The experimental setting and problem scales of three instance groups are listed in Table 5. The three parameters used in the previous criterion v, d, e are set as 18, 5 and 2 km/h, respectively.

Table 5: Experimental setting and scales of instances for the context "one shopper – multiple orders"

	$\|R\|$	$\|E\|$	$\|O\|$	Num. of var.	Num. of cons.
ISG1	25	15	2625	10^6	10^6
ISG2	35	20	7174	10^7	10^7
ISG3	50	30	20874	10^7	10^8

Notes: (1) $|R|$, $|O|$, and $|E|$ denote the number of actual orders, shoppers, and all the dummy orders, respectively. (2) Num. of var. and cons. denote approximate number of variables and constraints embedded in MIP models.

A series of comparative experiments are conducted to compare the dynamic programming based algorithm (elaborated in Section 4.1.3) and the improved algorithm (elaborated in Section 4.1.3). As the latter one is better than the former one, we choose the latter one for the further comparative experiments. We compare the improved algorithm without using the criterion (elaborated in Section 4.2) and the algorithm with using the criterion. As mentioned in Section 4.2, the criterion is used to reducing the computation time without loss of too much solution quality. The first series of comparative results (i.e., the value gap_A^I in Table 6) will demonstrate the effect of tactics (using the concept of sub-sequences) used in the improved algorithm; the second series of comparative results (i.e., the value gap_I^{I+C} in Table 6) will demonstrate the effect of the criterion. From results in Table 6, three conclusions are drawn as follows:

(1) The improved algorithm is better than the traditional dynamic programming based algorithm, because the former one can achieve the same objective value as the latter one but use shorter time than the latter one. The values in the column gap_A^I in Table are zero; while the computation time t_I is shorter than the time t_A. This comparative results validates the effect of tactics (using the concept of sub-sequences) used in the improved algorithm.

(2) The proposed criterion is efficient in reducing the computation time while maintaining solution quality. The value of t_{I+C} is much lower than that of t_I, while the value of gap_I^{I+C} is very small. This comparative results validates the effect of the criterion proposed in Section 4.2.

(3) The designed algorithm as well as the criterion are potentially useful and efficient for practitioners. For the largest-scale instance with about ten million variables (mostly integer) and about a hundred million constraints, our algorithm can solve the problem within about half an hour.

Table 6: Algorithmic performance for the context "one shopper – multiple orders"

Instance groups	Algorithm		Improved algorithm			Improved algorithm + Criterion		
	obj_A	t_A (s)	obj_I	gap_A^I (%)	t_I (s)	obj_{I+C}	gap_I^{I+C} (%)	t_{I+C} (s)
ISG1	443.28	366.73	443.28	0.00	271.95	441.48	0.41	116.38
	331.35	324.58	331.35	0.00	209.66	331.35	0.00	133.17
	270.08	385.44	270.08	0.00	239.71	270.08	0.00	153.57
	299.57	212.97	299.57	0.00	175.82	299.57	0.00	115.28
	240.67	238.06	240.67	0.00	181.09	238.76	0.79	90.32
ISG2	370.14	1067.04	370.14	0.00	719.15	370.14	0.00	507.77
	474.53	1117.32	474.53	0.00	816.14	474.53	0.00	539.78

	464.13	1055.28	464.13	0.00	698.04	463.96	0.04	431.83
	417.89	897.21	417.89	0.00	675.70	417.89	0.00	400.97
	428.05	858.66	428.05	0.00	627.64	426.26	0.42	424.95
	669.34	3415.30	669.34	0.00	2859.50	669.24	0.01	1499.16
	700.81	3632.57	700.81	0.00	2845.98	700.81	0.00	1690.72
ISG3	715.85	3525.17	715.85	0.00	2939.53	715.85	0.00	1851.73
	662.68	3225.21	662.68	0.00	2984.14	662.68	0.00	1535.43
	667.78	3091.22	667.78	0.00	2672.64	667.32	0.07	1550.22

Notes: (1) "Algorithm" is the solution method which uses the dynamic programming (elaborated in Section 4.1.3) to solve the M_{er} model and uses the CPLEX to solve M_Asgn model; the objective value and computation time are denoted by obj_A and t_A, respectively. (2) "Improved algorithm" is the solution methods which uses an improved algorithm (elaborated in Section 4.1.4) to solve the M_{er} model and uses the CPLEX to solve M_Asgn model; the objective value and computation time are denoted by obj_I and t_I, respectively. (3) "Improved algorithm + Criterion" is the solution method which applies the criterion (elaborated in Section 4.2), uses the improved algorithm to solve the M_{er} model, and uses the CPLEX to solve M_Asgn model; the objective value and computation time are denoted by obj_{I+C} and t_{I+C}, respectively. (4) $gap_A^I = (obj_I - obj_A)/obj_A$; $gap_I^{I+C} = (obj_{I+C} - obj_I)/obj_I$.

5. Model-driven strategic decisions for PSPs

The above proposed MIP models can also support PSP operators making some strategic level decisions such as deciding whether or not the PSP could be implemented in a district, given the estimated distribution of potential shoppers, customers and stores distributions. More specifically, we use the proposed models to run a large number of instances generated according to the realistic distributions or data patterns. Then, based on the solved results, we calibrate some formulas that could easily be used by decision makers when making some long-term decisions.

5.1 Supporting PSP mode adoption decision

As already mentioned, the adoption decision for this PSP mode mainly depends on the estimation of profit. While the fixed cost for establishing an online platform as well as its supporting team is relatively easy to estimate, the main difficulty behind this strategic level decision on mode adoption is the estimation of the revenue for a PSP operator.

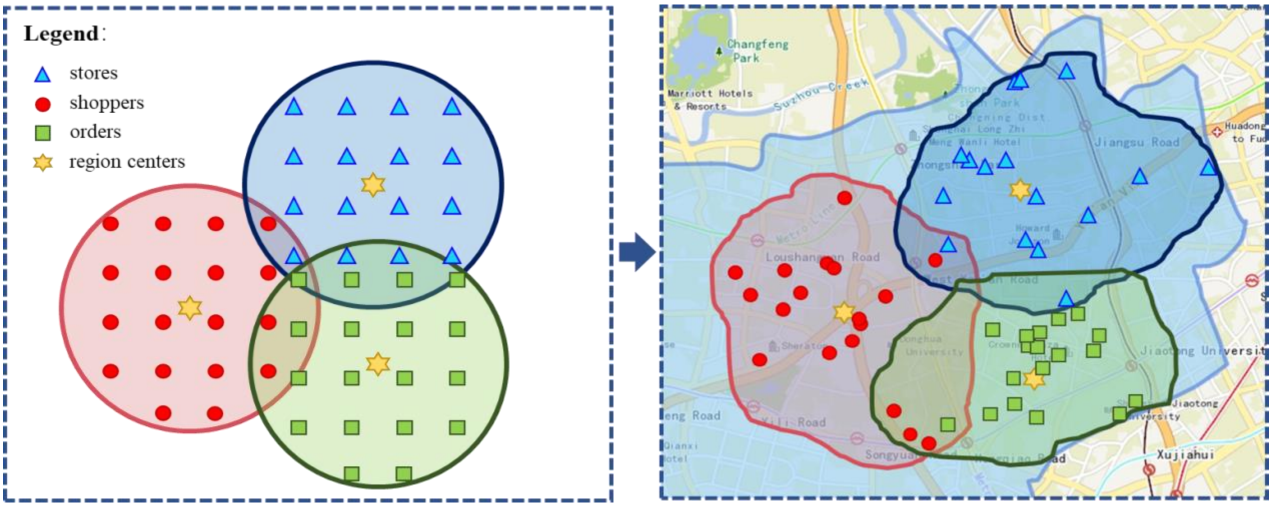
When a decision maker considers implementing this PSP mode in a district, the regions of potential customers, shoppers, and stores may be the most concerning issue for the decision maker. Where are these three types of players located? What are the densities of these three regions? What are the relative positions of these regions? The answers to these questions will affect the revenue estimation of a PSP. Therefore, the decision maker could benefit from a formula to estimate the revenue of a PSP during planning horizon, given three regions of shoppers, customers, and stores.

The objective of the MIP models proposed in Sections 3 and 4 is the maximization of a PSP's profit in fulfilling a batch of incoming orders. Given the estimate total number of orders during a planning horizon (e.g., one year), it is easy to use the models to calculate the PSP's annual profit derived from fulfilling orders. To this end, we use the MIP models to solve a large number of instances generated according to realistic distributions and data patterns. These instances differ from each other in some aspects such as the density of the three regions (customers, shoppers, and stores), the distance between these regions' centers, the regions' relative positions, the orders' average arriving rate (i.e., the number of orders over the planning horizon). Then, based on the solved results on these instances, we calibrate a formula using these variables. The estimated data can also be obtained through market surveys or other channels that have sufficient data on online shopping transactions.

More specifically, we define ρ_R, ρ_E, ρ_S to denote the densities of three regions (customers, shoppers, and stores), respectively, and we define r_R, r_E, r_S to denote the radius of each of the three regions. The distance between centers of customer region and store region is defined as d_{RS}; the distance between centers of customer region and shopper region is defined as d_{RE}, and the angle between these two segments (d_{RS} and d_{RE}) is defined as $\theta_{\angle SRE}$. These three variables d_{RS}, d_{RE} and $\theta_{\angle SRE}$ reflect the relative positions of the three regions. The order arrival rate is defined as λ_R orders per day. We aim to calibrate a function $f(\lambda_R, \rho_R, \rho_E, \rho_S, r_R, r_E, r_S, d_{RS}, d_{RE}, \theta_{\angle SRE})$ to estimate the annual revenue of a PSP. This function may not be calibrated easily through a formula due to the complexity of the problem context and to the number of independent variables; the calibration should be based on a very large number of instances since otherwise the obtained function will be less convincing for decision makers. For this case with 10 independent variables, if we set 10 different values (scales) for each one, we would need to run the model for 10^{10} times in order to obtain the instances used for the calibration. However, we can apply machine learning (ML) based methodologies to establish the mapping relationship between a number of inputs and the output (Bertsimas and Kallus, 2020). Due to the limitation of space and the limited contribution on the ML methodologies, we can only sketch and briefly illustrate process of applying some ML techniques such as the BP neural network, random forest, and K nearest neighbor in the calibration process for the $f(\cdot)$. Taking BP neural network as an example, a Multilayer Feed-Forward Neural Network (MFFNN) with ten input nodes and one output node is constructed. Based on the training data, the above-mentioned ten variables should be normalized, which subsequently constitute an input vector $\boldsymbol{x} = [\lambda_R, \rho_R, \rho_E, \rho_S, r_R, r_E, r_S, d_{RS}, d_{RE}, \theta_{\angle SRE}]^T$ and flow into the MFFNN. Then, Back Propagation (BP) is used to modify the weights and bias iteratively. Whenever the gap between the expected value (gained by the BPNN) and the target value (actual annual profit) is less than the predetermined threshold, a trained neural network has been obtained.

In the remainder of this subsection, we use the example of the Changning district to explain the core idea of calibrating a formula on estimating annual revenue of a PSP in a district. As shown in Figure 4, there are three regions with 18 orders, 18 shoppers, and 16 stores during a planning horizon of 15 minutes. The radius of the three regions is 1.5 km. In the real case, we use the real travel distance as a measure of the radius of regions, thus the boundaries of regions are not standard circles. According to the measurement on a real map, the distance between the centers of the customer and of the store regions (d_{RS}) is 1.536 km; the distance between the centers of the customer and shopper regions (d_{RE}) is 1.581 km; and the angle between these two segments (d_{RS} and d_{RE}) is 66.173°, i.e., $\theta_{\angle SRE}$. For this real case, we use our proposed models and algorithm to run 50 randomly generated instances (random demand of orders); the average profit is 94 (for $d_{RS} = 1536$, $d_{RE} = 1581$, $\theta_{\angle ERS} = 66.173°$).

We then apply the ML to calibrate a formula, and compare the result calculated on the basis of the formula with the above real case's result. As shown in the left part of Figure 4, we use a theoretical case to generate numerous instances in order to calibrate the formula. When constructing the theoretical case, we use the order arrival rate, the radius and densities of the regions as the values in the real case. Then these factors are not embedded in the formula as independent variables. Therefore, we calibrate a function $f(d_{RS}, d_{RE}, \theta_{\angle SRE})$ in this case study.



Theoretical case *A real case*

Figure 4: Regions of customer orders, shoppers and stores on a PSP

Based on the theoretical case, we generate 8,000 groups of instances, each containing 50 instances with the same value of d_{RS}、d_{RE}、$\theta_{\angle ERS}$ and different demand of orders. On the basis of the 400,000 instances, we run the model, obtain results, and calibrate a formula for the function $f(d_{RS}, d_{RE}, \theta_{\angle SRE})$ $= 242.7952 + 0.014 * \sin(23.4118 * \theta_{\angle ERS} + 452.2943) + 0.1004 * \sin(31.5335 * \theta_{\angle ERS} + 44.7728) + \left(267.1207 * \sin(0.1692 * \theta_{\angle ERS} + 5.7535) + 89.9942 * \sin(0.809 * \theta_{\angle ERS} + 1.7939)\right) * d_{RS} + \left(-29.4533 + 4.8174 * \sin(3.4585 * \theta_{\angle ERS} - 3.7298)\right) * d_{RE} + \left(-11.2016 - 6.4419 * \sin(1.4864 * \theta_{\angle ERS} + 390.3841)\right) * {d_{RS}}^2 + \left(140.5808 * \sin(0.0843 * \theta_{\angle ERS} + 15.5233) + 7.2409 * \sin(2.2506 * \theta_{\angle ERS} + 6.869)\right) * d_{RS} * d_{RE} + \left(-15.7608 * \sin(0.0007 * \theta_{\angle ERS} + 7.6675) + 10.5655 * \sin(0.1604 * \theta_{\angle ERS} + 0.5742)\right) * {d_{RE}}^2$, R-square = 0.9969.

According to the above calibrated formula based on the theoretical case, the result for the setting ($d_{RS} = 1536$, $d_{RE} = 1581$, $\theta_{\angle ERS} = 66.173°$) is 92.58, which represents a gap 1.5% from the realistic result (i.e., 94). This small gap validates the feasibility of the above proposed methodology that supports the PSP model adoption decision in a specific district.

5.2 Supporting territory planning under the PSP mode

If a large district is to be implemented with a PSP, how to make the territory planning for this district is yet another important strategic level decision. As in typical delivery service platforms, the area of interest is partitioned into several territories. The customers (orders) in a territory are usually served by the shoppers in the same territory. Then an important long-term decision is to perform the territory planning so that the PSP's profit is maximized. This subsection elaborates on how to use the previously proposed MIP models of order assignment to support this strategic level decision on territory planning.

Although there exist some theoretical studies on territory planning (Carlsson 2012; Carlsson and Devulapalli 2013) which are based on a continuous plane, the business areas or residential areas have usually been well recognized in reality; the boundaries of these areas are relatively deterministic. Thus the territory planning in this study's context is actually a discrete problem that consists of deciding which areas should be merged as a "large area" (called by "territory" in this study) so that the PSP's profit is maximized.

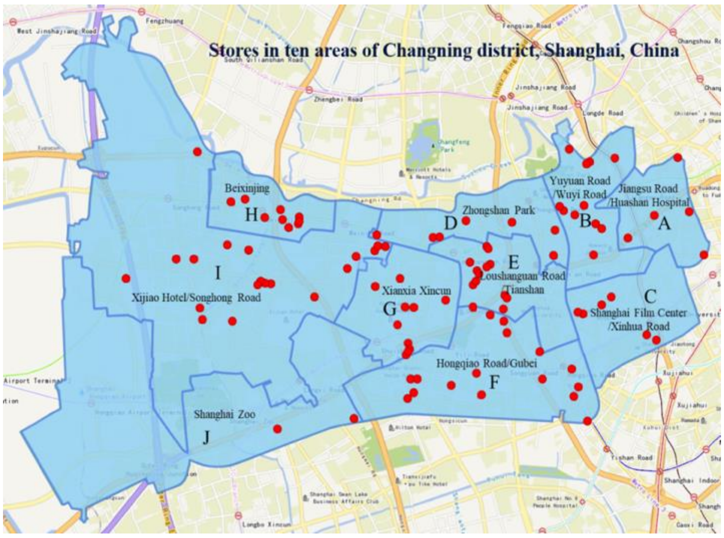
Given two adjacent areas $\mathcal{R}_a$ and $\mathcal{R}_{a'}$ with parameters related to customer orders, stores shoppers in the areas, we use the previously proposed MIP models of order assignment to obtain the PSP's profit denoted by $f(\mathcal{R}_a)$ and $f(\mathcal{R}_{a'})$, respectively. If we merge the two areas, we can also use the MIP models to obtain the PSP's profit denoted by $f(\mathcal{R}_a \cup \mathcal{R}_{a'})$. Since the demands are random, we can calculate the expected value of the profit under a set of random demands. The above mentioned three results are denoted by $\mathbb{E}_{\tilde{d}_a} f(\mathcal{R}_a)$, $\mathbb{E}_{\tilde{d}_{a'}} f(\mathcal{R}_{a'})$, and $\mathbb{E}_{\tilde{d}_a,\tilde{d}_{a'}} f(\mathcal{R}_a \cup \mathcal{R}_{a'})$; here $\tilde{d}_a$ and $\tilde{d}_{a'}$ denote the set of random demands in areas $\mathcal{R}_a$ and $\mathcal{R}_{a'}$, respectively.

Given a set of well recognized areas in a district, $\boldsymbol{\mathcal{R}} = \{\mathcal{R}_a\}_{a\in A}$, A is the set of all the areas' indices. For each pair of two adjacent areas $\mathcal{R}_a$ and $\mathcal{R}_{a'}$, we define a percentage $\Delta_{a,a'}$ as

$$\Delta_{a,a'} = \frac{\mathbb{E}_{\tilde{d}_a,\tilde{d}_{a'}} f(\mathcal{R}_a \cup \mathcal{R}_{a'}) - \mathbb{E}_{\tilde{d}_a} f(\mathcal{R}_a) - \mathbb{E}_{\tilde{d}_{a'}} f(\mathcal{R}_{a'})}{\mathbb{E}_{\tilde{d}_a} f(\mathcal{R}_a) + \mathbb{E}_{\tilde{d}_{a'}} f(\mathcal{R}_{a'})}, \ \forall a, a' \in A.$$

The value $\Delta_{a,a'}$ is non-negative. For all pairs of two adjacent areas, we calculate all the values of $\Delta_{a,a'}$, and merge the two areas with the largest $\Delta_{a,a'}$ value. For the remainder set of $|A| - 1$ areas, we repeat the above procedure until no pair of two adjacent areas can be merged with a positive $\Delta_{a,a'}$ value larger than a threshold. The decision maker can set this threshold in advance according to some implicit factors that have effect on this area-merging decision.

We use again the case of the Changning district of Shanghai to illustrate this procedure. As already mentioned, the areas in a district are usually well known by practitioners. Figure 5 depicts 10 such areas in the Changning district as well as the numbers of orders and available shoppers in each area during a planning horizon, which is set to 15 minutes in this study.



Area index	Area name	Numbers of orders $\|R\|$	Numbers of shoppers $\|E\|$
A	Jiangsu Road/Huashan Hospital	54	34
B	Yuyuan Road/Wuyi Road	50	46
C	Shanghai Film Center/Xinhua Road	88	72
D	Zhongshan Park	39	33
E	Loushanguan Road/Tianshan	26	20
F	Hongqiao Road/Gubei	93	79
G	Xianxia Xincun	48	44
H	Beixinjing	30	34
I	Xijiao Hotel/Songhong Road	86	82
J	Shanghai Zoo	25	22

Figure 5: Initially recognized areas in Changning district and numbers of orders and shoppers in areas

We calculate the $\Delta_{a,a'}$ for each pair of adjacent areas. All calculations are based on 10 instances with randomly generated demand information of orders. The average values are recorded in Table 7, which illustrates the whole decision process for merging areas. The threshold of $\Delta_{a,a'}$ value for the area-merging decision is set at 1.5% in this case study. There are four iterations in this decision process. As shown in Table 7, the largest $\Delta_{a,a'}$ value in the first iteration is 3% for the merge of areas D and E; then the two areas are merged as an area K. In the second iteration, the largest $\Delta_{a,a'}$ value is 2.7% for the merge of areas H and I; then the two areas are merged as an area L. In the third iteration, the largest $\Delta_{a,a'}$ value is 1.8% for the merge of areas A and B; then the two areas are merged as an area M. In the fourth iteration, the largest $\Delta_{a,a'}$ value is 1.4% and lower than the predetermined threshold 1.5%. Thus the calculation process terminates. The final version of the territory planning is shown in Figure 6.

Table 7: Iterations for merging areas in the strategic decision of PSP's territory planning

Two adjacent areas	$f(\mathcal{R}_a)+f(\mathcal{R}_{a'})$	$f(\mathcal{R}_a \cup \mathcal{R}_{a'})$	$\Delta_{a,a'}$	Two adjacent areas	$f(\mathcal{R}_a)+f(\mathcal{R}_{a'})$	$f(\mathcal{R}_a \cup \mathcal{R}_{a'})$	$\Delta_{a,a'}$	Two adjacent areas	$f(\mathcal{R}_a)+f(\mathcal{R}_{a'})$	$f(\mathcal{R}_a \cup \mathcal{R}_{a'})$	$\Delta_{a,a'}$
Iteration 1											
AB	1460	1487	1.8%	AC	1969	1984	0.7%	BC	2114	2131	0.7%
BD	1313	1328	1.1%	CE	1754	1754	0.0%	CF	2511	2527	0.6%
DE	**930**	**958**	**3.0%**	DG	1183	1187	0.2%	EF	1586	1599	0.8%
EG	1083	1108	2.3%	FG	1906	1926	1.0%	FJ	1543	1543	0.0%
GI	1608	1625	1.0%	GJ	997	998	0.0%	HI	1457	1497	2.7%
IJ	1293	1315	1.7%								
Iteration 2											
AB	1460	1487	1.8%	AC	1969	1984	0.7%	BC	2114	2131	0.7%
CF	2511	2527	0.6%	FG	1906	1926	1.0%	FJ	1543	1543	0.0%
GI	1608	1625	1.0%	GJ	997	998	0.0%	**HI**	**1457**	**1497**	**2.7%**
IJ	1293	1315	1.7%	BK	1698	1712	0.8%	CK	2234	2234	0.0%
KF	2110	2117	0.3%								
Iteration 3											
AB	**1460**	**1487**	**1.8%**	AC	1969	1984	0.7%	BC	2114	2131	0.7%
CF	2511	2527	0.6%	FG	1906	1926	1.0%	FJ	1543	1543	0.0%

GJ	997	998	0.0%	BK	1698	1712	0.8%	CK	2234	2234	0.0%
KF	2110	2117	0.3%	GL	2055	2073	0.8%	LJ	1664	1688	1.4%
Iteration 4											
CF	2511	2527	0.6%	FG	1906	1926	1.0%	FJ	1543	1543	0.0%
GJ	997	998	0.0%	CK	2234	2234	0.0%	KF	2110	2117	0.3%
GL	2055	2073	0.8%	LJ	1664	1688	1.4%	MC	2693	2697	0.1%
MK	2278	2288	0.4%								

Notes: $f(\mathcal{R}_a) + f(\mathcal{R}_{a'})$ denotes the sum of the individual profits of two areas before merging; $f(\mathcal{R}_a \cup \mathcal{R}_{a'})$ denotes the profit if the two areas are merged; $\Delta_{a,a'}$ is the percentage of the profit increment for merging the two areas.

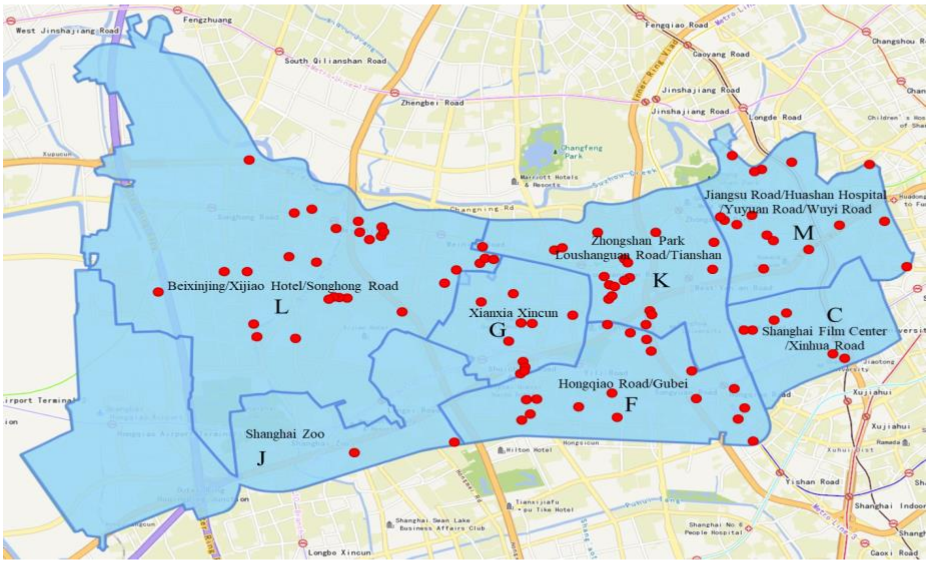


Figure 6: Final version of the PSP's territory planning after merging areas

6. Conclusions

We have proposed a comprehensive decision tool for the PSPs based on the use of MIP models. This decision tool can be useful for improving the operations management level for the PSP operators. The main contributions of this study are summarized as follows:

(1) For this newly emerging business mode, this paper is probably the first work that investigates the operations management and optimization problems. A comprehensive set of MIP models are provided for supporting the PSP operators' operational level decisions on order assignment and shopper routing under various context.

(2) Some practical algorithms based on dynamic programming as well as some criteria for accelerating the computational speed were designed in order to facilitate the application of the above proposed methodology to realistic large-scale instances. Our proposed simple but practical algorithm can solve the basic case with up to 1,000 orders and 1,050 shoppers (with about 10^7 variables and 10^7 constraints in MIP models) in half a minute.

(3) Some industrial implications are also obtained by some comparative experiments between our proposed model and some decision rules. The experimental results show that our proposed model can increase the platform's profit by about 14%. In addition, this study demonstrates the influence of the percentage of shoppers with ongoing tasks on the final platform's profit; the larger is the percentage, the

more significant is the relative advantage of our proposed model.

(4) Besides proposing models for supporting operational level assignment and routing decisions, this study also illustrates the possible application of these MIP models in some strategic level decisions such as the PSP mode adoption and the PSP's territory planning. A realistic case on Changning district in Shanghai was also used to validate the effectiveness of the proposed models and algorithms.

However, this study also contains limitations which could set the ground of further studies. For example, dynamic pricing has not been taken into account, and a PSP could set different unit fees with respect to travel distance for customers with different priorities when determining delivery fees for customers. Future studies could also consider more complex factors such as the explicit relationship between delivery fees and customers' perceived welfare. In addition, the probability data on the future demand arrivals should also be taken into account for some tactical or strategical level decisions in the PSPs, which can further improve the utility of the proposed methodology for this new business mode.

Acknowledgement

The authors would like to thank the editor and three anonymous reviewers for their constructive suggestions on improving this paper. This research is supported by the National Natural Science Foundation of China (Grant numbers 72025103 and 71831008).

References

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

Aziez, I., Côté, J.-F., Coelho, L.C. (2020) Exact algorithms for the multi-pickup and delivery problem with time windows. *European Journal of Operational Research* 284(3): 906–919.

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

Benjaafar, S., Ding, J.-Y., Kong, G., Taylor, T. (2022) Labor welfare in on-demand service platforms. *Manufacturing & Service Operations Management* 24(1): 110–124.

Bertsimas, D., Kallus, N. (2020) From predictive to prescriptive analytics. *Management Science* 66 (3): 1025–1044.

Boysen, N., Briskorn, D., Schwerdfeger, S. (2019) Matching supply and demand in a sharing economy: Classification, computational complexity, and application. *European Journal of Operational Research* 278(2): 578–595.

Carlsson, J.G. (2012) Dividing a territory among several vehicles. *INFORMS Journal on Computing* 24(4): 565–577.

Carlsson, J.G. Devulapalli, R. (2013) Dividing a territory among several facilities. *INFORMS Journal on Computing* 25(4): 730–742.

Dayarian, I., Savelsbergh, M.W.P. (2020) Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *Production and Operations Management* 29(9): 2153–2174.

Deng, Q., Fang, X., Lim, Y.F. (2021) Urban consolidation center or peer-to-peer platform? The solution to urban last-mile delivery. *Production and Operations Management* 30(4): 997–1013.

Fatehi, S., Wagner, M.R. (2021) Crowdsourcing last-mile deliveries. *Manufacturing & Service Operations Management*. In press, doi: 10.1287/msom.2021.0973.

He, F., Yang, J., Li, M. (2018) Vehicle scheduling under stochastic trip times: an approximate dynamic programming approach. *Transportation Research Part C* 96: 144–159.

Hong, J., Lee, M., Cheong, T., Lee, H.C. (2019) Routing for an on-demand logistics service. *Transportation Research Part C* 103: 328–351.

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

Naccache, S., Côté, J.-F., Coelho, L.C. (2018) The multi-pickup and delivery problem with time windows. *European Journal of Operational Research* 269(1): 353–362.

Najmi, A., Rey, D., Rashidi, T.H. (2017) Novel dynamic formulations for real-time ride- sharing systems. *Transportation Research Part E* 108: 122–140.

Novoa, C., Storer, R. (2009) An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research* 196(2): 509–515.

Özkan, E. (2020) Joint pricing and matching in ride-sharing systems. *European Journal of Operational Research* 287(3): 1149–1160.

Qi, W., Li, L., Liu, S., Shen, Z.-J. M. (2018) Shared mobility for last-mile delivery: Design, operational prescriptions, and environmental impact. *Manufacturing & Service Operations Management* 20(4): 737–751.

Sun, P., Veelenturf, L.P., Hewitt, M., Van Woensel, T. (2018) The time-dependent pickup and delivery problem with time windows. *Transportation Research Part B* 116: 1–24.

Sun, P., Veelenturf, L.P., Hewitt, M., Van Woensel, T. (2020) Adaptive large neighborhood search for the time-dependent profitable pickup and delivery problem with time windows. *Transportation Research Part E* 138: 101942.

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

Taylor, T. (2018) On-demand service platforms. *Manufacturing & Service Operations Management* 20(4): 704–720.

Ulmer, M.W., Goodson, J.C., Mattfeld, D.C., Hennig, M. (2019) Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science* 53(1): 185–202.

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

Wang, H. (2019) Routing and scheduling for a last-mile transportation system. *Transportation Science* 53(1):131–147.

Wang, Y., Lei, L., Zhang, D., Lee, L.H. (2020a) Towards delivery-as-a-service: Effective neighborhood search strategies for integrated delivery optimization of E-commerce and static O2O parcels. *Transportation Research Part B* 139: 38–63.

Wang, J., Kang, L., Liu, Y. (2020b) Optimal scheduling for electric bus fleets based on dynamic programming approach by considering battery capacity fade. *Renewable and Sustainable Energy Reviews* 130: 109978.

Yildiz, B. (2021) Express package routing problem with occasional couriers. *Transportation Research Part C* 123: 102994.

Yildiz, B., Savelsbergh, M. (2019a) Provably high-quality solutions for the meal delivery routing problem. *Transportation Science* 53(5): 1372–1388.

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

Yıldırım, Ş., Yıldız, B. (2021) Electric bus fleet composition and scheduling. *Transportation Research Part C* 129: 103197.