Submitted to *Service Science*
manuscript

Personal Shopper Systems in Last-Mile Logistics

J. P. van der Gaast
Department of Management Science, Fudan University, P.R. China jpgaast@fudan.edu.cn

A. M. Arslan
Department of Management Science, Lancaster University, United Kingdom, a.arslan1@lancaster.ac.uk

This paper explores the logistics operations of instant grocery delivery services. Therefore, we introduce the Instant Delivery Problem (IDP) to replicate and examine two widely adopted strategies in the rapid delivery market: Personal Shopper Systems (PSS) and Inventory Owned Delivery Systems (IOD). In the PSS, couriers visit affiliated brick and mortar stores in the delivery area to pick up, purchase ordered products and deliver them to customers. Whereas in the IOD, couriers collect products from a single distribution center or so-called a dark-store, in which the platform manages the inventory. Even though a PSS strategy is asset-light due to the utilization of existing retailers in the area, maintaining a good level of on-time instant deliveries with the PSS is more complex than the IOD. This is because the PSS requires deciding which store to purchase ordered goods, and picking and shopping at stores needs to be considered in the real-time decision process. We propose a tailored rolling horizon framework that utilizes column generation to browse updated delivery plans for arriving customer orders. Computational studies both in real-life inspired settings and in case studies on selected urban areas show the PSS is a highly competitive strategy compared to the IOD, particularly when dealing with small-sized customer orders. We observe that the performance of the PSS is robust when varying the delivery service time frame. The case studies also suggest that the PSS becomes even more competitive in areas where the retail store density is high.

Key words: Instant Delivery, Personal Shopper Systems, Last-Mile Logistics, Optimization, Dark Stores

1. Introduction

Online shopping has become popular due to increased urbanization, rapid e-commerce adaptation, and changed customer expectations (Savelsbergh and Van Woensel 2016, Grand View Research 2019). Especially, instant delivery is increasingly becoming widespread since it integrates the convenience of online shopping and the immediacy of buying products from a brick and mortar store. As a result, many online platforms around the world have started to offer instant deliveries with shipment times as short

Table 1 Examples of instant delivery platforms varying inventory choices as of 2021

Platform	Speed	Region	Inventory-Source	Operating strategy
Amazon-PrimeNow	In two-hour	North America	Own at a DC	IOD
Instacart	In one-hour	North America	Affiliated stores	PSS
GrabMart	As soon as possible	Southeast Asia	Affiliated stores	PSS
Getir.com	As soon as possible	Europe, North America	Own at a DC	IOD
Meituan	As soon as possible	China	Affiliated stores	PSS
Ele.Me	As soon as possible	China	Affiliated stores	PSS
Gorillas	As soon as possible	Europe	Own at a DC	IOD
CornerShop	As soon as possible	South America	Affiliated stores	PSS
OneKiosk	As soon as possible	Africa	Affiliated stores	PSS

as an hour, making these platforms not only a strong competitor to offline shopping but also to traditional online shopping services.

Nevertheless, managing an economical and sustainable instant delivery service is challenging as these services share many of the same issues faced by traditional two-day and next-day delivery systems (Stroh et al. 2021). Those are tight order delivery deadlines, low order volumes, and high levels of order variability. This structure leads to inefficient delivery fleet utilization, and consequently, substantial operational expenses (Arslan 2019). In addition, there is a growing requirement for storage facilities to be in or nearby urban areas to sustain expedited delivery, which in turn increases capital investment expenses even further. These difficulties have led to the emergence of new business models reorganizing the fundamental design of instant delivery services.

A prominent strategy to tackle these issues is the *Personal Shopper System* (PSS). This system uses an asset-light last-mile strategy that utilizes the existing retail infrastructure and its assortments in a delivery area to provide an expedited and convenient shopping experience. That is, the PSS provides an online shopping platform in which customers browse and order desired products from nearby brick and mortar stores. When a customer places an order, the platform assigns the order to a courier (*shopper*), who is either hired by the platform or is an independent contractor. This courier then collects and purchases the requested products, similar to a regular customer, at one or more stores nearby. Afterward, the courier delivers to the customers' address within the given delivery deadline. Moreover, each courier can simultaneously work on multiple customer orders, increasing the number of orders that can be accepted and delivered within a short time frame and the efficiency of the platform.

Table 1 shows an overview of popular platforms using instant grocery deliveries from nearby stores. We observe that instant delivery platforms operate on two strategies that differ in the source of the inventory. While in *Inventory Owned Delivery (IOD)*, the platforms fulfill the customer orders solely from a distribution center (DC), or so-called a dark store managed by the same platform; in the *Personal Shopper System (PSS)*, the platform collaborates with affiliated stores and shoppers purchases goods from those stores.

A key advantage of the PSS is that retail stores are utilized as storage facilities. This allows the system to bypass distribution centers in the last-mile chain, which are reported to contribute up to 50% of the total logistics cost (Alicke and Losch 2015). Furthermore, opening distribution centers is often stated as the significant barrier for platform providers to scale up their services due to the hefty capital requirements for in or nearby urban areas (Harrington 2015).

As a last-mile logistic service, the PSS carries out two-core logistics processes: goods collection and order distribution. Nevertheless, the PSS's organization is fundamentally different from the IOD, which relies on the storage facilities owned and managed by the service provider, because of the following reasons;

- **Real-time store selection:** In the PSS, there is no need for fulfillment centers. However, due to the smaller store sizes and assortments, not every customer order can be fully fulfilled by a single store. Therefore, a courier might need to visit multiple stores in succession before an order is complete. This differs from other same-day or on-demand services which typically require a single pick-up for the order fulfillment (Ulmer et al. 2020b). As a result, the dispatching decisions are more complicated as the dispatcher has to decide which store to shop on top of which order is assigned to a courier, and what route should be chosen. This is further complicated since products might be in the assortment of multiple stores and new consolidation options can occur whenever new customer orders arrive.
- **In-store collection**: In the PSS, a courier's job expands as they now also collect and purchase customer orders at stores before delivery. Fulfilling goods of varying sizes at independent retailer stores requires a large-scale reorganization in supply chain operations and coordination. Furthermore, the provider operating with the PSS strategy should account for the pick-up duration in different store types since the order collection and distribution processes form a significant part of a courier's task. This differs from the traditional last-mile logistics which mainly concerns the distribution process and the transportation times between various stops.

In this paper, we introduce the *Instant Delivery Problem* (IDP) in which customers located in a delivery area receive ordered goods within a promised delivery service time frame to primarily compare the operational efficiencies of two instant delivery service strategies; (i) the Personal Shopper System (PSS), and (ii) the Inventory Owned Delivery (IOD) system. This problem is a generalization of instant delivery problems as it allows dispatchers to select pick-up stores in real-time and considers explicitly in-store collection duration. Within the IDP, we derive a granular operational model that represents the PSS- and IOD-specific processes and decisions accurately. Subsequently, we propose a large-scale rolling horizon optimization approach that is capable of handling real-life size instant delivery services efficiently. This solution approach effectively integrates column generation into a real-time sequential decision-making process. Using column generation allows the system to re-optimize the couriers' trips at the time of new order arrives. Finally, we examine the advantages of the PSS strategy compared to the IOD in varying settings and urban delivery environments through a comprehensive set of numerical experiments. We observe that the PSS strategy is a compelling alternative to the IOD when customers' order sizes are small. Furthermore, the results show that the performance of the IOD system is relatively sensitive to the location of the distribution center/dark store and delivery time promises, whereas the PSS is robust to delivery service time deviations. The experiments in the urban delivery environment underline the advantages of the PSS over the IOD, and they clearly demonstrate that our solution approach is viable for real-life applications.

The remainder of the paper is organized as follows. In Section 2, we present the relevant literature. Section 3 provides a detailed description of the problem. In Section 4, we introduce the rolling horizon optimization approach. Section 5 presents the numerical results of the computational analysis. A comprehensive discussion of the advantages of PSSs, limitations, and possible extensions of our approach is given in Section 6.

2. Related Literature

This paper focuses on the instant delivery problem (IDP), particularly the comparison of two strategies: the PSS with the IOD. In addition, it deals with organizing a large-scale on-demand service platform. Consequently, our literature review concentrates first on studies in last-mile logistics, and second on resource management optimization for online marketplaces.

2.1. Last-mile logistics

The IDP shares several characteristics with well-known problems in the dynamic pickup and delivery literature (Pillac et al. 2013, Ulmer 2020), and stochastic vehicle routing literature (Ritzinger et al. 2016). In these problems, stochastically arriving customers are served by a fleet of couriers, and these problems seek assignment and routing decisions to coordinate couriers as economical as possible. Most of these studies focus on meal delivery operations. In such a system, a restaurant aggregator platform coordinates couriers' trips between restaurants and customer locations to fulfill restaurant-specific orders in a limited time. Several operational aspects of the restaurant meal delivery problem such as courier-order assignment, and commitment strategies are recently studied by Reyes et al. (2018), Steever et al. (2019), Ulmer et al. (2020b). Nevertheless, the essential difference is that the IDP relaxes the coupling relationship between the order and stocking locations such that a product can be picked up at one of the stores selling the product. This feature enables consolidation possibilities in stores, particularly when in-store shopping time is a concave function of the number of products. One exception to the order-restaurant coupling in the meal delivery service is the study by Yildiz and Savelsbergh (2019), where the authors numerically examine the option of picking the meal order from another restaurant offering the same cuisine. However, this work analyzes the static problem. Furthermore, in meal delivery problems, couriers do not spend time finding and collecting products in restaurants as restaurants prepare orders to be picked by couriers.

The studies by Arslan et al. (2021), Schrotenboer et al. (2021) are the closest ones in terms of the setting of the IDP as these papers explore the idea of retail stores as storage facilities for expedited deliveries. Nevertheless, both of these papers, similar to the restaurant meal delivery problem, consider a setting in which customers specify the store where orders will be collected. This assumption is relaxed, as mentioned, when studying the PSS.

A special case of the IDP is when all customer orders are collected at a single distribution center. This problem is known as the same-day delivery problem (SDD). Note that this problem corresponds with the IOD in our description. Klapp et al. (2018), Voccia et al. (2019), Ulmer et al. (2019) study and examine the impact of dispatching, routing, order acceptance, and the preemptive distribution center return operations for the same-day delivery services, in which orders arrive throughout the service period, and, they should be delivered before the period ends. As orders are only collected at a single distribution center, store selection, which is essential in a PSS, is not part of the same-day delivery problem. Furthermore, order collection and preparation are handled in the distribution center; thus, the SDD does not consider these activities for courier planning. In a PSS, couriers are also responsible for order picking and packing within stores before delivering them to customers. In a similar vein, Ulmer (2017) explores the impact of delivery time promises as short as 60 minutes within the SDD framework.

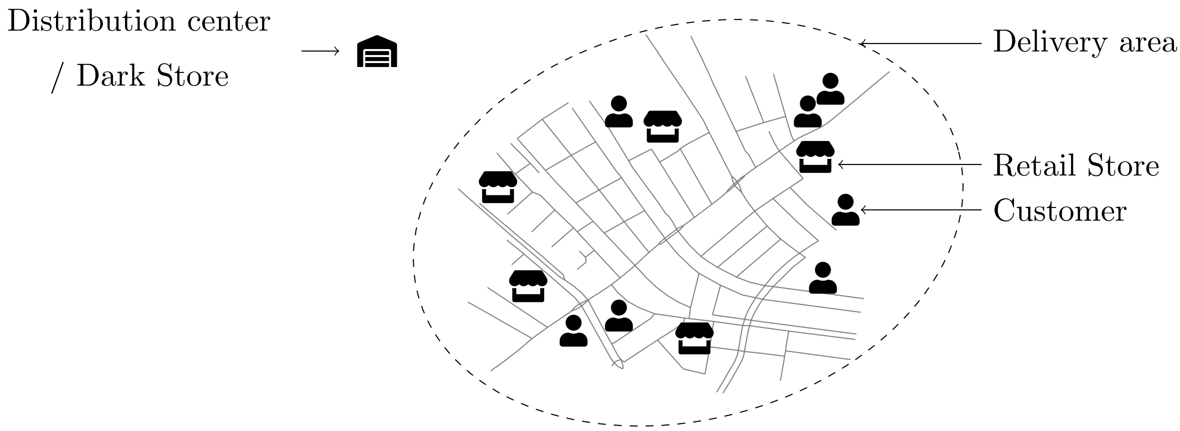
When orders are known in advance, i.e., the static variant of the IDP, it is worth mentioning the multi pick-up and delivery problem (MPDP). In the MPDP, a fleet of couriers serves customer orders requiring multiple pickups at different locations (Naccache et al. 2018, Aziez et al. 2020). Unlike the IDP, the pickup points are fixed per order; hence, the MPDP solely considers assignment and routing decisions. Furthermore, the picking/shopping time consolidation is not considered in these problems. Also, there is the location routing problem (LRP) (Drexl and Schneider 2015, Karaoglan et al. 2012), and the multi-depot vehicle routing problem (MDVRP), in which routing and pickup locations are chosen jointly that corresponds with the store selection feature of the IDP (Montoya-Torres et al. 2015). Nevertheless, the setting of these problems is static and does not model the in-store collection duration depending on the number of products.

2.2. Operations in large-scale on-demand service platforms

The IDP and decisions for the personal shopper systems can be analyzed under the operations in large-scale on-demand transportation and logistics platforms. This type of platform requires managing their physical resources effectively to satisfy continuously arriving, location dependent, and time-sensitive customers' requests (Taylor 2018). Hence, a growing number of recent papers study real-time decisions for resource management, optimization and coordination in the different domains of the on-demand services, such as ride-hailing (Bertsimas et al. 2019, Alonso-Mora et al. 2017, Wang and Yang 2019), item-sharing (Behrend and Meisel 2018), car-sharing (Boyacı et al. 2017), and bike-sharing (Pei et al. 2022).

We observe that surge pricing (Bai et al. 2019, Guda and Subramanian 2019), capacity planning and resizing (Benjaafar et al. 2021, Lyu et al. 2019), and online assignment and matching (Özkan and Ward 2020) are the most widely used operational strategies to

Figure 1 Overview of the instant delivery network



increase the efficiency of the resources. Our paper contributes a new operational strategy for the on-demand delivery services through real-time store selection, particularly in cases when customers are indifferent to the location of their requests' origin. With the PSS, the on-demand delivery platform can select shopping/pick-up locations to achieve better utilization of their resources.

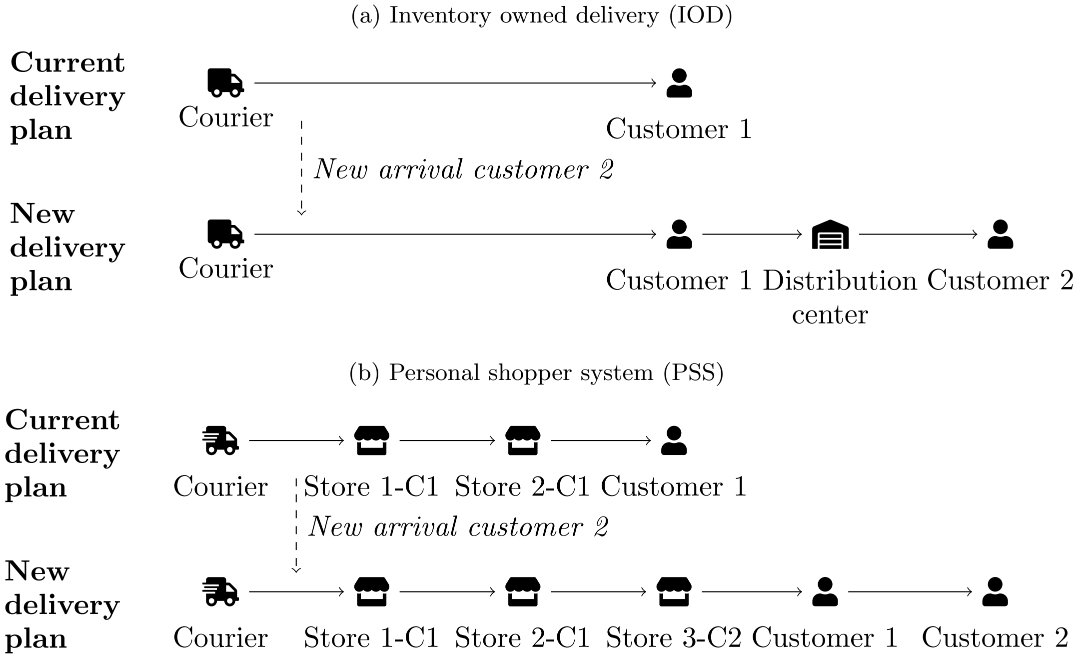
3. Problem Definition

In this section, we formally describe the *Instant Delivery Problem* (IDP). We first provide an overview of the problem in Section 3.1. Afterward, we present the mathematical notation in Section 3.2. Finally, in Section 3.3 we describe the dynamics of our model and the underlying *Sequential Decision Process* (SDP).

3.1. Overview

Inspired by the real-world platforms as listed in Table 1, we consider an instant delivery setting where customers located in a delivery area receive ordered goods within a promised delivery service time frame, e.g., an hour after they placed the order. An example of the corresponding network is shown in Figure 1. In an IOD, distribution centers or dark stores, which are typically located within or nearby the delivery area, contain the inventory that is used to fulfill all customer orders within the delivery area. Furthermore, they serve as both the starting and ending points for the couriers' trips. On the contrary, in a PSS, customer orders are fulfilled using retail stores' inventory located within the delivery area. A courier visits potentially multiple stores within the delivery area to fully pick up a customer order, afterward, they will deliver the collected order to the customer's address.

Figure 2 The difference in the responsiveness of different instant delivery systems in the rolling delivery plan



A crucial difference between an IOD and the proposed PSS is their responsiveness and, consequently, their overall operational efficiency. Particularly, the difference in the responsiveness of both systems can be observed when new customer orders are incorporated into the rolling delivery plans, see Figure 2. In the case of the IOD in Figure 2a, whenever a new customer order arrives, the current delivery plans for shipment cannot accommodate a new order without the courier first returning to the distribution center (DC) and picking up the new customer order from there. Moreover, dispatched couriers cannot easily return to the DC as they are bound to serve orders already loaded within their delivery windows. In particular for Figure 2, customer 2's order arrive after the courier departs at the DC. Therefore, the courier has to return to DC to pick up customer 2's order. On the other hand, in a PSS as shown in Figure 2b, new customer orders can be incorporated within the current plan and do not require the courier first to finish its current delivery plan. Due to the proximity of the stores, it might be possible to visit an additional/different store for the new customer order without violating any of the promised delivery service time frames. For example, in Figure 2b the courier has to visit two stores (Store 1 and Store 2 and where C1 denotes customer 1) before being able to deliver to customer 1. In case customer 2 is added to the current delivery plan, it might be feasible to pick up the products for this customer at another store (Store 3) before delivery to customer 1 in case customer 2 is located nearby customer 1. This will save additional travel time as well as increase the responsiveness of the system.

Nevertheless, the order pickup time and the average retrieving cost per order can be advantageous in the IOD due to larger economies of scale. However, when the promised delivery service times shrink to only a few hours, and orders appear dynamically within the service period, recalling the couriers to the DC can be regarded as a fixed cost which consequently becomes disadvantageous for the IOD. Therefore, by investigating the IDP, we can examine the fundamental trade-offs between the IOD and PSS given varying customer and store densities, delivery duration promises, and order pick times.

In the next subsection, the IDP will be described which will allow for the comparison of the IOD and PSS.

3.2. The Instant Delivery Problem

The IDP considers an instant delivery platform that consists of a set of retail stores/distribution centers (stocking facilities) that carry an assortment of products, and couriers fulfilling orders placed by customers in a defined delivery area. The relevant assumptions and notation for the IDP are as follows:

Service period. We denote the service period, i.e., the time when a customer can place orders, as $\mathcal{T}$, where $t = 0$ is the start of the service period and $t = T$ is the end of the service period.

Geography, Products, and Storage facilities. The IDP covers a predetermined delivery area. The platform sells an assortment of products, denoted by set $\mathcal{P}$, and each product can be purchased/collected at one or multiple storage facilities. Set $\mathcal{M}$ denotes all storage facilities that are affiliated with the platform. Each storage facility is associated with relevant attributes (from the point of the IDP) which are given by a vector $(v_m, \mathcal{P}_m)$, where $v_m \in \mathcal{V}^M$ is the storage facility location in the delivery area, $\mathcal{P}_m \subseteq \mathcal{P}$ is the set of products available at the facility. Reversely, $\mathcal{M}_p^S$ is the set of stores where product $p \in \mathcal{P}$ can be picked up. A storage facility can either be a store that is located within the delivery area ($m \in \mathcal{M}^S$) or a distribution center ($m \in \mathcal{M}^D$).

Customer Orders. Throughout the service period, the arrival times of customer orders follow a stochastic process. Let $\mathcal{O}$ denote all arriving orders during the service period, which is unknown to the platform at the start of the service period. Each order $o \in \mathcal{O}$ corresponds to an attribute vector $(B_o, a_o, v_o, d_o, s_o)$ where B_o is a basket of products, a subset of $\mathcal{P}$, to be delivered, $a_o \in \mathcal{T}$ is the order placement time,

Table 2 Notation for the IDP

Sets	
$\mathcal{T}$	The service period, $t = 0, \dots, T$
$\mathcal{P}$	The set of products, $p \in \mathcal{P}$
$\mathcal{M}$	The set of storage facilities, $\mathcal{M} = \mathcal{M}^S \cup \mathcal{M}^D$
$\mathcal{M}^S$	The set of stores, $m \in \mathcal{M}^S$
$\mathcal{M}^S_p$	The set of stores where product $p \in \mathcal{P}$ can be picked up
$\mathcal{M}^D$	The set of distribution centers, $m \in \mathcal{M}^D$
$\mathcal{K}$	The set of couriers, $k \in \mathcal{K}$
$\mathcal{O}$	The set of orders, $o \in \mathcal{O}$
$\mathcal{V}$	The set of vertices representing the location of storage facilities $\mathcal{V}^M$ and delivery addresses of orders, $\mathcal{V}^O$, $\mathcal{V} = \mathcal{V}^M \cup \mathcal{V}^O$
Storage facilities	
v_m	The location $v_m \in \mathcal{V}^M$ of storage facility $m \in \mathcal{M}$
$\mathcal{P}_m$	The set of products available at storage facility $m \in \mathcal{M}$
Orders	
B_o	The basket of products that are requested by order $o \in \mathcal{O}$
a_o	The arrival time of order $o \in \mathcal{O}$
v_o	The delivery address $v_o \in \mathcal{V}^O$ within the delivery area of order $o \in \mathcal{O}$
d_o	The delivery deadline of order $o \in \mathcal{O}$
s_o	The delivery time at the customer address of order $o \in \mathcal{O}$
Couriers	
q_k	The number of orders courier $k \in \mathcal{K}$ can serve simultaneously
w_{v_i,v_j}	The travel time between vertices, $v_i, v_j \in \mathcal{V}$
$f_m(n)$	The service duration of a visit to facility $m \in \mathcal{M}$ while retrieving $n \in \mathbb{Z}^+$ products
Rewards	
r	The reward for the platform for delivering an order on time
System state	
S_{a_o}	The system state when order $o \in \mathcal{O}$ arrives
$\mathcal{O}_{a_o}$	The set of active orders at decision epoch a_o
P_{a_o}	The list of products that still need to be collected for the active orders in $\mathcal{O}_{a_o}$
Θ_{a_o}	The current delivery plan at decision epoch a_o, where $\theta_k \in \Theta_{a_o}$ is the forward trip of courier $k \in \mathcal{K}$
J_h	The list of products to pick/drop at vertex $v_h \in \mathcal{V}$ in a forward trip
z_h	The departure time of the courier from vertex $v_h \in \mathcal{V}$ in a forward trip
Decisions	
$\mathcal{X}_{a_o}$	The set of actions (delivery plan modifications) that can accommodate arriving order o, $x \in \mathcal{X}_{a_o}$
$\Theta^x_{a_o}$	The new delivery plan obtained after action x is selected
Objective	
Π	The set of all Markovian policies, $\pi \in \Pi$
$\delta^\pi_{a_o}$	A function that specifies the selected action given system state is S_{a_o} following policy $\pi \in \Pi$

$v_o \in \mathcal{V}^O$ is the delivery address, $d_o > a_o$, is the (hard) order delivery deadline, and, $s_o \geq 0$ is the delivery time at the customer address.

In addition, we assume two selection modes in which customers can either select the store from which the product is collected or let the platform decide where the product is picked. In case under the *store selection* mode, basket B_o also contains the locations, $v_m \in \mathcal{V}^m$, where the courier should pick up the products. Otherwise, under the *product selection* mode, the platform is free to select a store in $\mathcal{M}_p^S$ to collect product $p \in B_o$ from.

Couriers. A fixed number of couriers, denoted by set $\mathcal{K}$, fulfill the customer orders during the service period. That is, a courier visits a store or multiple stores, purchases the products on behalf of customer o, and eventually delivers the basket of B_o to the customer address v_o. Otherwise, the courier visits the DC and picks up multiple orders at once before delivery. Each courier k travels at a constant speed and carries up to q_k orders at the same time. We denote the travel time between vertices as w_{v_i,v_j} where $v_i, v_j \in \mathcal{V}$. In addition, we assume that order splitting is not allowed, i.e., a single courier is fully responsible for completely fulfilling a particular customer order.

The duration of a visit by a courier at storage facility $m \in \mathcal{M}$ depends on the type and size of the facility and the number of products that can be collected. This duration is defined by the following function $f_m : \mathbb{Z}_+ \mapsto \mathbb{R}_+$ with the following properties:

— $f_m(0) = 0, \ \ m \in \mathcal{M}$,

— $f_m(n)$ is increasing and concave,

in which n is the number of products retrieved at facility m at a single visit given the orders that are currently assigned to the courier. In a PSS, this time also includes, picking, paying, and packing time done by the courier. The above properties indicate that collecting (and purchasing in case of a PSS) more products requires more time, but the marginal time per product decreases, which enables economies of scale.

Rewards. We assume each order $o \in \mathcal{O}$ brings an equal amount of reward of $r \geq 0$ to the platform. The objective of the IDP is to serve as many orders as possible with a fixed number of couriers. Therefore, we need to consider the following decisions at the arrival of each order:

1. Accept the arriving order to the platform's couriers or decline service,
2. Organize the courier's shopping and routing plans.

Note that orders can be rejected if it is not possible to construct a delivery plan such that the order is delivered on time.

In the following section, we formulate the dynamics of the IDP and describe the underlying the sequential decision problem.

Note that the IDP is described for the PSS operation strategy. Nevertheless, the IOD operation strategy is a special case of the PSS strategy as a single shop (distribution center/dark store) holding and providing all products for customer orders. As a result, in the IOD strategy, the platform does not require deciding where to collect the products. Finally, in the IOD, we do not consider the preemptive DC returns, i.e., the courier returns the depot only when she finishes all delivery tasks with her.

3.3. Dynamics of the IDP: The Route-based Sequential Decision Process

The IDP operates in a dynamic environment since information about customer orders is stochastically revealed during the service period. We assume that the platform has little a priori information available regarding the number and size of customer orders during the service period for setting the size of the courier fleet. Therefore, we present the IDP as a finite-horizon stochastic dynamic problem (Powell 2019). We utilize the modeling components based on a rolling horizon optimization and a route-based Markov Decision Process (Ulmer et al. 2020a, Arslan et al. 2021) as follows:

Decision epochs. In our model, a decision epoch a_o occurs when a customer places a new order o, after which the platform has to decide to accept the order and/or modify the current delivery plan. The total number of decision epochs during the service period depends on the order arrival distribution, and the process terminates when the service period ends, $a_o > T$.

System (decision) states. The state of the system at each decision epoch a_o can be represented by a tuple $S_{a_o} = (\mathcal{O}_{a_o}, P_{a_o}, \Theta_{a_o})$, where $\mathcal{O}_{a_o}$ is the set of active orders, accepted for service but not delivered by the time of a_o plus the order that just arrived, and P_{a_o} as the list of products still needed to be collected for these orders. That is, we can ignore all delivered and rejected orders at the decision epoch. $\Theta_{a_o} = \{\theta_1, \dots, \theta_{|K|}\}$ is the current delivery plan that contains the couriers' forward trips. A forward trip for courier $k \in \mathcal{K}$, $\theta_k = \{(v_0, J_0, z_0), \dots, (v_h, J_h, z_h)\}$, is the sequence of activities defined by the vertex v_h, list of products to pick/drop at the vertex is denoted by J_h, and the departure time, z_h, from the associated node.

Inspired by Yildiz and Savelsbergh (2019), we initially set the departure time for an activity to as soon as possible. That is, a courier does not strategically wait at any location if the forward trip consists of the following activities after the incumbent one. Therefore, the departure times at each vertex can be recursively determined using the following formula

$$z_j = z_{j-1} + w_{v_{j-1},v_j} + \begin{cases} f_{m(v_j)}\left(|J_j|\right), & \text{if } v_j \in \mathcal{V}^M, \\ s_{o(v_j)}, & \text{otherwise,} \end{cases} \tag{1}$$

for $j = 1, \ldots, h$, and where $m(v_j)$ denotes the storage facility at $v_j \in \mathcal{V}^M$ and $o(v_j)$ denotes the order o at customer address $v_j \in \mathcal{V}^O$. The intuition behind the formula is that the departure time at vertex v_j depends on the previous departure time, z_{j-1}, plus the travel time between the previous and current vertex and the time spent at vertex v_j which either consists of collecting products at a storage facility or dropping off a customer order. Note that (v_0, J_0, z_0) corresponds with the last completed activity in case the courier idles or the activity the courier is currently working on. Moreover, for a forward trip to be feasible all associated departure times for the assigned orders in θ_k should be $z_j \leq d_{o(v_j)}$, where $j = \left\{0, \ldots, h \mid v_j \in \mathcal{V}^M\right\}$. In conclusion, a forward courier trip provides a full description of the details to serve all assigned orders by courier $k \in \mathcal{K}$.

Actions and rewards. An action modifies the current delivery plan to accommodate new order o, collecting the associated products and dropping them off within the forward courier trips. This modification can (re)-determine the storage facilities to collect products, (re)-assign orders across couriers and (re)-sequence vertices in the couriers' forward trips. Let $x \in \mathcal{X}_{a_o}$ denote an action that can accommodate order o at epoch a_o which subsequently modifies the delivery plan into Θ_o^x. As long as $\mathcal{X}_{a_o} \neq \emptyset$, the order is accepted and a constant reward is received:

$$r_{a_o}(x) = \begin{cases} r, & \text{if } x \in \mathcal{X}_{a_o}, \\ 0, & \text{otherwise.} \end{cases}$$

If it is not possible to accommodate the order, i.e., $\mathcal{X}_{a_o} = \emptyset$, the delivery plan remains unchanged, $\Theta_{a_o}^x = \Theta_{a_o}$.

The aforementioned action is obtained by solving a combinatorial optimization problem, in which determining a feasible delivery plan that accommodates the new order in it is a $\mathcal{NP}$-complete problem. In Section 4, we formally define the associated optimization problem, called the *Courier Dispatch Problem*.

Transitions. A new decision epoch is triggered at a_{o+1} with the arrival of new order $o+1$. Through this transition, the system state changes as follows:

— The decision epoch becomes a_{o+1}.

— Set $\mathcal{O}_{a_o}$ is updated by the removal of completed deliveries and/or a rejected order. In addition, the new arriving order $o+1$ is added to $\mathcal{O}_{a_{o+1}}$.

— Courier trips in $\Theta^x_{a_o}$ are updated by the removal of visited vertices.

Objective function Let Π be the set of all Markovian deterministic policies and a policy π is a sequence of decision rules: $\pi = (\delta^\pi_{a_1}, \delta^\pi_{a_2}, \ldots, \delta^\pi_{a_{|\mathcal{O}|}})$, where each decision rule $\delta^\pi_{a_o} : \mathcal{X}_{a_o} \mapsto x$ is a function that specifies the chosen action given that the system state is S_{a_o} following policy π. Hence, in the IDP we seek a policy $\pi \in \Pi$ that maximizes the total expected reward for the platform:

$$\max_{\pi\in\Pi} \mathbb{E}\big[\sum_{o\in O} r_{a_o}(\delta^\pi_{a_o}(S_{a_o}))\big]. \tag{2}$$

4. Rolling Horizon Approach

In this section, we present the rolling horizon approach to analyze the IDP. Initially, we present the reason behind this approach and its advantages. Afterward, we describe the solution steps and the PSS- and IOD-specific processes.

4.1. Motivation

The IDP, formulated as a Markov Decision Problem, suffers from the *curse of dimensionality* due to the number of active orders and products to be collected at a decision epoch. Furthermore, the action space at each decision epoch requires browsing an exponentially growing number of options of (re)-assignment, (re)-routing, and (re)-store selection to accommodate the new order. At a given epoch and excluding any future order arrivals, the corresponding decision problem is a variant of the Multi-Depot Vehicle Routing Problem with Time Windows, which is a $\mathcal{NP}$-complete problem (Uit het Broek et al. 2021). Considering the dynamic nature of the IDP, it is near impossible to evaluate all feasible forward shopper trips for real-size problems in a limited time.

To overcome these challenges arising from the size of the action space, we introduce an optimization problem, named *Courier Dispatch Problem* or shortly CDP that determines feasible forward trips accommodating the newly arriving order at a decision epoch.

Table 3 Courier Dispatch Problem — Classification of activities at an epoch

	Sequencing	Store re-selection	Courier re-assignment
Committed activities	✓		
Partially-committed activities	✓	✓	
Uncommitted activities	✓	✓	✓

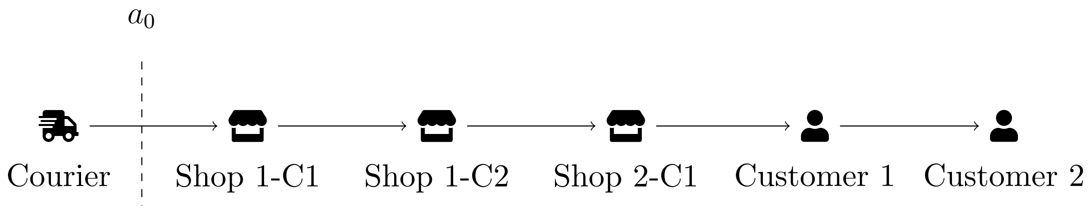
4.2. The Courier Dispatch Problem

The CDP at state S_{a_o} is concerned with joint store selection, assignment and routing decisions for updating the couriers' forward trips. The primary purpose of the CDP is to seek a new delivery plan in which the arriving order o can be collected and delivered without violating any of the delivery service time frames of the active orders at the given moment.

Since the CDP's main objective is to modify couriers' forward trips at a given epoch a_o, it is key to classify the activities in the ongoing forward trip into three groups: (i) *committed activities*, (ii) *partially-committed activities*, and (iii) *uncommitted activities*. On the one hand, if an activity is committed, it means that the courier is required to continue picking/dropping the preassigned products at the predetermined vertices with the only possibility of changing the visiting sequence; i.e., these activities remain attached to the courier after the new order arrival. A shopping activity is partially committed if the customer order is assigned to the courier, but the upcoming visit to the preassigned store (vertex) can be altered to another store. Note, that these activities correspond with orders for which the courier already picked one or more products. On the other hand, an uncommitted activity can be removed, reassigned or re-sequenced to another courier, which corresponds to orders for which the courier has not picked up any product of them yet. That is, some preassigned customers can be assigned to another courier, or pre-determined storage facilities for the products of the committed orders can be altered to new storage facilities, see Table 3 for the summary.

We consider that a forward trip, $\theta_k \in \Theta_{a_o}$, can be modified after the first upcoming activity following the decision epoch. In other words, the courier will not be en-routed before the first assigned activity is done. Figure 3 illustrates an example of classifying activities. At the pre-decision epoch of a_o, the ongoing forward trip instructs visiting Shop 1 and picking-up the products of customer 1 and customer 2, then moving to Shop 2 to pick up the remaining products for customer 1. The trip continues with visiting the delivery vertices of customers 1 and 2. Note that picking up the products

Figure 3 Illustration of committed activities (black, Shop 1-C1), partially committed activities (Shop 1-C2), and uncommitted activities (Shop 2-C1) of the forward trip at decision epoch a_o. While the courier is committed to serve customer 1 (C1), customer 2 (C2) can be assigned to another courier. Also, courier's visit to Shop 2 can be altered.



for customer 1 at Shop 1 is the first activity when new order o arrives. Therefore, the courier should serve customer 1, and therefore the courier commits all remaining pick-ups and delivery for customer 1. Nevertheless, at this epoch the courier has not picked up any product for customer 2 yet, which, therefore, are all uncommitted activities.

Formally, the CDP is defined over feasible forward trips $\Theta(k)$ for each courier $k \in \mathcal{K}$. Note that the courier forward trip for courier k is a sequence of activities $\theta_k = \{(v_0, L_0, z_0), \ldots, (v_h, L_h, z_h)\}$ where the (v_o, L_o, z_o) represents courier k first activity after a_o (courier k is either idling at the location or carrying out the assigned forward trip at decision epoch a_o). The forward courier trip is feasible if all activities in the trip satisfy the delivery deadlines without violating the courier's carrying capacity. Let τ_θ be the total time required for all activities in the forward trip θ; i.e., $\tau_\theta := z_h - z_0$.

Let $\Theta(o)$ be the collection of all feasible trips customer order o is served. Furthermore, let x_θ^k be the binary decision variable taking value of 1 if θ is chosen for the updated delivery plan for courier $k \in \mathcal{K}$. Then, the set partitioning formulation for the CDP is written as follows:

$$\min \quad \sum_{k \in \mathcal{K}} \sum_{\theta \in \Theta(k)} \tau_\theta x_\theta^k \tag{3a}$$

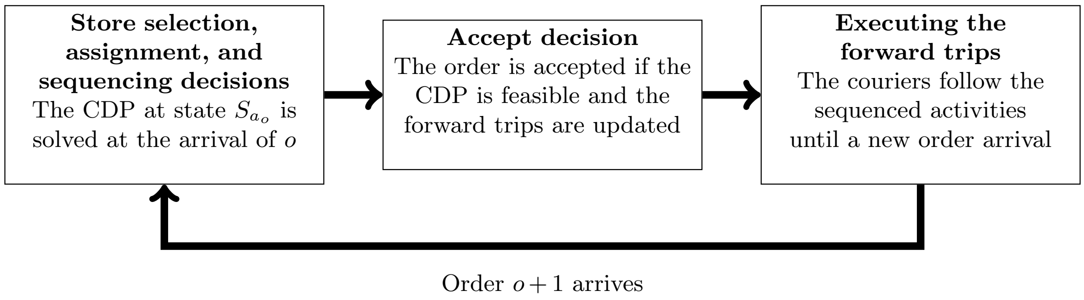
$$\text{s.t.} \quad \sum_{k \in K} \sum_{\theta \in \Theta(o) \cap \Theta(k)} x_\theta^k = 1, \qquad o \in \mathcal{O}_{a_o}, \tag{3b}$$

$$\sum_{\theta \in \Theta(k)} x_\theta^k \leq 1, \qquad k \in \mathcal{K}, \tag{3c}$$

$$x_\theta^k \in \{0, 1\}, \qquad k \in \mathcal{K}, \theta \in \Theta(k). \tag{3d}$$

The objective in (3a) is to minimize the total time required to fulfill orders in $\mathcal{O}_{a_o}$. Constraint (3b) guarantees that all active orders including a new one should be served and Constraint (3c) ensures that a courier can serve at most one forward trip at a time.

Figure 4 Instant delivery service operating mechanism



Since the number of possible forward trips $\Theta(k)$ for courier k at a given epoch can be extremely large, we use column generation to solve the CDP. Column generation was already proven itself to be very effective in solving these kinds of problems in a limited amount of time (Desaulniers et al. 2006). The column generation starts with a reduced limited set of possible forward trips and new columns (forward trips) are obtained by solving a pricing problem. For the CDP, the associated pricing problem is a resource constraint shortest path problem which is known to be $\mathcal{NP}$-Complete. We employ the column generation without progressing into branching, which enables a fast solution. To circumvent the use of a branch-and-price algorithm, we use a well known result of integer programming (Nemhauser and Wolsey (1988) page 389 Proposition 2.1) which implies: Given an UB and a LB with the associated duals, only the columns whose reduced costs are smaller than or equal to $\zeta = UB - LB$ can appear in the optimal integer solution of the master problem. We further describe the column generation and the pricing algorithm in Section EC.2.

4.3. Instant delivery service operating mechanism

In this section, we describe the complete operating mechanism of the IDP. The associated platform continuously coordinates the courier's forward trips by keeping the system state up-to-date at each new order arrival. When an order arrives, a new CDP is constructed and solved. If the CDP identifies at least one feasible set of forward trips that serve the incoming order, the order is accepted and the forward trips of the couriers are updated. The couriers follow the activities of the forward trips until the new order arrival. Figure 4 provides a summary of all the steps.

The instant delivery platform' mechanism slightly varies whether the operating strategy is a PSS or an IOD. The primary reason behind this distinction is that couriers act differently in these two systems due to the different job descriptions. While a courier in IOD performs solely delivery activities, the courier (shopper) in the PSS does the products collected in stores. Hence, the visiting time at a store in the PSS accounts for the number of products to be collected. The other crucial distinction between the two strategies is where the products are stored. In the IOD, couriers always return the DC to collect products picked and prepared by dedicated employees as a result the associated CDP is relatively less complex than the CDP in the PSS, in which store selection is mixed with assignment and routing decisions.

5. Numerical Results

To evaluate the PSS in an instant delivery setting, we conduct a series of computational experiments. We first describe how the instances are generated. Then, we provide details on the solutions and sensitivity in terms of the operational benefits of the PSS compared to an IOD. Afterward, we describe the performance of the PSS and IOD for several case studies.

5.1. Instance generation

We test the performance of an instant delivery service in an artificial environment. We consider a square-shaped 10 km × 10 km delivery area that consists of all stakeholders; storage facilities (a distribution center/dark store (DC) and retail stores), customers, and couriers. The retail stores and customers' delivery addresses are spread uniformly within the delivery area. The DC is initially located at the corner of the delivery area, see Figure 5.

We test the performance of the two instant delivery platforms in the given delivery area. The PSS collaborates with 30 retailer stores and fulfills the customer orders through these stores, while the IOD uses the DC to collect the products to fulfill the orders. Both platforms offer an assortment of 100 products. In the IOD, the DC contains all products and pickers prepare the basket of products for an order to be collected by the couriers. On the other hand, in the PSS, each product is available at a subset of stores. $|\mathcal{M}_p^S|$ indicates the number of stores where a product can be purchased. We assume that $|\mathcal{M}_p^S|$ is identical for each product p throughout the experiments for easier interpretation. Throughout the numerical experiments, we consider two scenarios, in which $|\mathcal{M}_p^S| = 5$ or $|\mathcal{M}_p^S| = 10$. This means any product p is available in 5 out of 30 or 10 out of 30 stores, respectively. Furthermore, it is assumed that products do not go out

Figure 5 **Illustration of the squared-shaped delivery area**

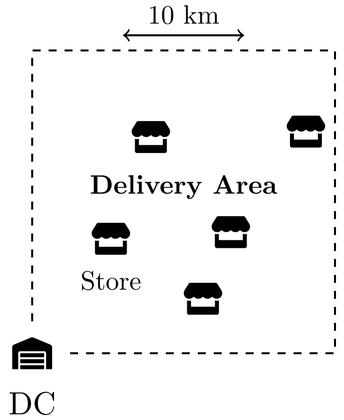


Table 4 **Parameters for the artificial environment**

$d_o - a_o,\ o \in \mathcal{O}$	90 minutes
Fleet size, $\lvert\mathcal{K}\rvert$	3 couriers
Courier capacity, q_k	2 orders
Fixed store visit duration	UNIF(4,8) minutes
A product collection at a store	UNIF(1,3) minutes
Fixed depot visit time	8 minutes
Order arrival rate	0.2 order/min
Number of stores, $\lvert\mathcal{M}_p^S\rvert$	5 or 10 stores

of stock during the service period and couriers in the PSS have up-to-date knowledge of where each product is located.

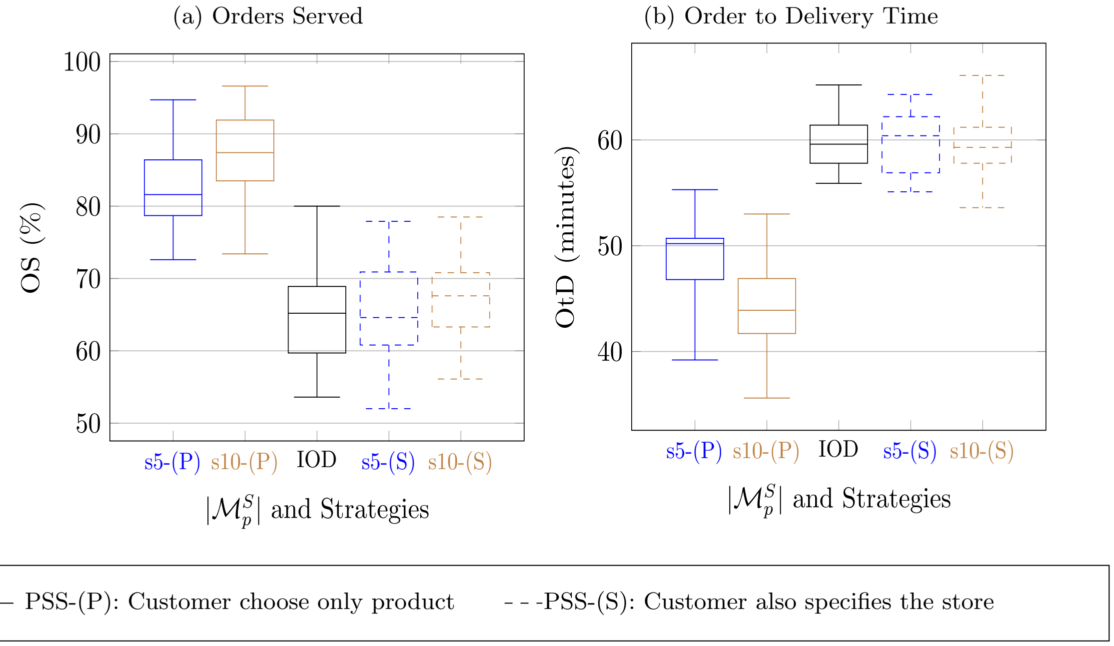
A customer order consists of up to three products, chosen randomly among the assortment of 100 products. We consider that the PSS may adopt two possible selection modes: (i) product selection (P) and (ii) store selection (S). In the former option, a customer only specifies products in the order, in the latter one, the customer also specifies a pick-up store for each product. We distinguish these two options under the names *PSS-(P)* and *PSS-(S)*. In the case of PSS-(S), we choose the customer's store specification randomly among the stores selling the products in the order's basket.

Orders arrive following a homogeneous Poisson process throughout the service period of twelve hours. We consider an arrival rate of 0.2 orders/min (one order arrival for every five minutes on average). A customer order should be fulfilled within a given lead time of 90 minutes; i.e., $d_o = a_o + 90$.

Three homogeneous couriers with a constant speed of 30 km/hour are responsible for shopping at stores in the PSS, or picking up orders at the DC in the IOD, and delivering the orders. Distances between storage locations and customers are obtained by calculating the Euclidean distance. Each courier is identical in terms of the product carrying capacity of 2 orders (up to 6 products).

A courier incurs a fixed visit time and variable pickup time depending on the number of products that will be collected at a store. The fixed store visit and variable duration per product collection are randomly drawn from the uniform distribution, UNIF(4,8) minutes and UNIF(1,3) minutes, respectively. (In other words, the duration of a store visit is an affine function with the constant term is drawn UNIF(4,8) and variable term

Figure 6 **PSS-(P) vs IOD vs PSS-(S), lead-time: 90 minutes, 3 couriers, order size: one product**



drawn UNIF(1,3) for each store, which is concave and increasing). In the IOD, a courier only spends a fixed 8 minutes of visit time at the DC regardless of the number of orders that are collected. Table 4 summarizes the parameters set for the base-case experiments.

5.2. Base-case results

Figure 6 presents the results averaged over 30 random replications. The figure shows the performance of the two PSS selection modes, PSS-(P) and PSS-(S), and the IOD. Figure 6 further shows the results if a product can be picked up at 5 or 10 stores in the PSS, represented by s5 and s10.

We report the following performance indicators for the instant delivery platforms: (i) *% Orders Served (OS)*: the percentage of arriving orders served by the platform; (ii) *Order to Delivery time (OtD)*: the average time in minutes between when the order arrives and its delivery time.

In Figure 6, we see that PSS-(P) outperforms the IOD in terms of the percentage of served orders. The percentage of orders served with PSS-(P) is 87.4% and 81.6% (median value) depending on whether the products are available in ten or five stores. On the other hand, with the IOD strategy the platform can serve only 65.2% of the arriving orders.

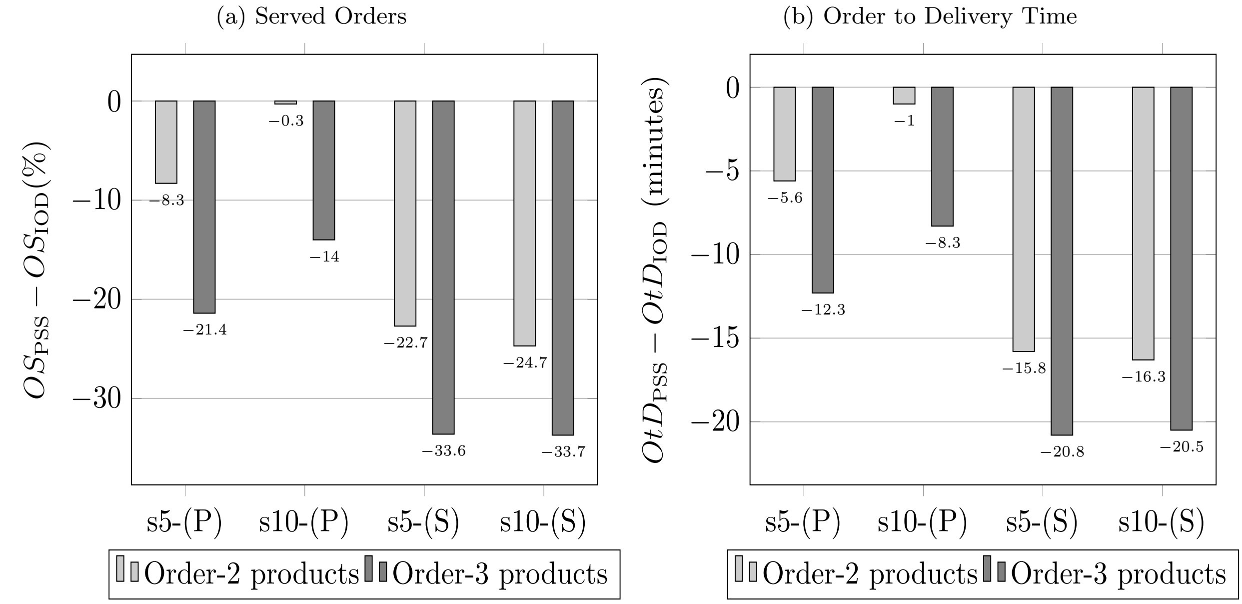
We also see that the store selection mode in the PSS has a considerable impact on operational efficiency. The percentage of orders (OS) served significantly decreases when the PSS platform chooses to offer the PSS-(S) option over the PSS-(P); i.e., customers specify the stores for product pick-ups. That is, the OS decreases to 67.6% and 64.6% for the cases where products can be picked up at ten or five stores, respectively. This result highlights the additional efficiency that the PSS-(P) obtains by jointly deciding on routing and store selection, which provides additional consolidation benefits at the picking time at stores. Moreover, it is worth highlighting that the performance of PSS-(S) is comparable to the IOD, as the OS is 65.2% for the IOD.

It is also seen that the OtD decreases considerably with the PSS-(P) strategy, see Figure 6b. When the PSS can choose pick-up stores (i.e., PSS-(P)), the customers receive their orders in an average of 43.9 and 50.2 minutes in case of a product is available at ten or five stores. Nevertheless, for the IOD, the OtD is 59.6 minutes and for PSS-(S) OtD for products that are available in ten or five stores are 59.4 and 60.4 minutes. These results show again the benefits of store selection for the PSS.

Figure 7a and Figure 7b present the results in which orders contain two or three products instead of one, i.e., an order requiring shopping at most at two or three stores. The figures show the absolute difference in order served and OtD between the PSS and the IOD. OS_{PSS} and OS_{IOD} are defined as the percentage of orders served in case of the PSS or the IOD. First, we see that the PSS fails to achieve the same service level as the IOD when the order size increases to two products per order. In almost all settings, except s10-(P) with order size two, the OS and the OtD deviate significantly from the IOD. Second, we see that the bigger the order size, the number of orders that all four PSS configurations can serve decreases quicker than the IOD. That is, when a product is available at ten stores (the column indicated by s10-(P)), the IOD serves 0.3% or 14% more orders in comparison to the PSS-(P) strategy for order sizes 2 and 3, respectively. Bigger order sizes hinder the PSS in two ways: (i) a personal shopper visits more stores per order, and hence driving and shopping takes longer per order; (ii) the likelihood of shopping at a single store for multiple orders shrinks significantly and consequently, the PSS utilizes fewer consolidation opportunities.

In Figure 7a, we continue to see the advantage of the PSS-(P) strategy over PSS-(S). Similarly, we observe that it is beneficial for products to be widely available in many stores for the PSS strategy. In other words, while the PSS strategy almost performs equivalent to the IOD when a product is available at 10 out of 30 stores and the order size is two, the OS decreases noticeably, and the product availability is lower.

Figure 7 PSS-(P) vs IOD vs PSS-(S), L:90 minutes, 3 couriers



5.3. Sensitivity to the problem characteristics

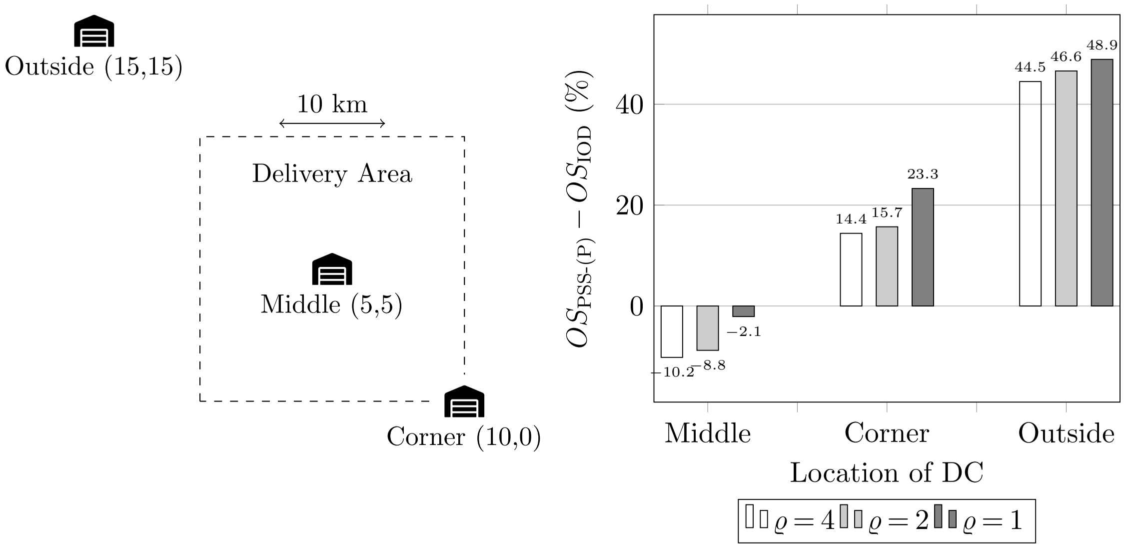
In this section, we provide insights regarding the sensitivity to problem characteristics, namely, the delivery lead time, the location of the depot, and the shopping time at stores.

5.3.1. The impact of pickup time in the DC and DC's location. This section explores the impacts of two inter-related parameters in the IOD: (i) the location of the DC, and (ii) the fixed visiting time at the DC. It should be noted that both parameters have a strong influence on the performance of IOD's outcome. At the same time, the DC's location and/or efficiency at the DC pickup are strongly associated with the investment decision. For example, the location of the DC nearby the delivery area's centroid or outside the delivery area can vary the amount of investment significantly. Therefore, we concentrate solely on reporting the operational impacts of these parameters.

Figure 8a presents three settings where the DC can be located (i) *middle*, (ii) *corner*, and (iii) *outside* the delivery region. Figure 8b shows the results of the difference in the OS between the PSS-(P) and the variations of IOD, derived from the locations of DC and the ratio of visiting time at a store and the DC; $\varrho = \frac{\text{Pick up time at a store}}{\text{Pick-up time at DC}}$. That is, the visit time of a store can take four or two times longer than visiting the DC on average, or it takes an equal amount of time for a single product pick-up as in the base-case experiments.

Figure 8 Impact of DC location, L:90 minutes, 3 couriers

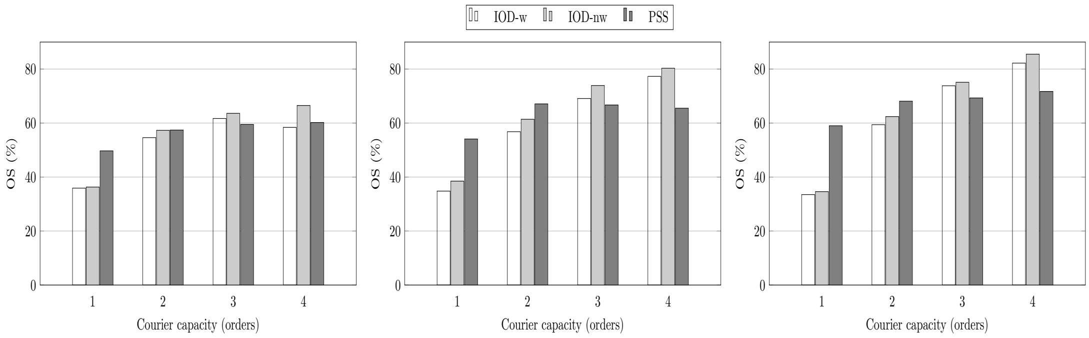
(a) Three possible DC locations (b) Orders Served (OS)



We see that both the location of the DC and visiting time at the DC have a substantial influence on the performance of the IOD. It is worth noticing that the IOD outperforms the PSS-(P) if the DC is located in the middle of the delivery area. In this case, the additional percentage of orders served with IOD increases by 10% if the DC visit time is four times is less than a store visit time in the PSS-(P) strategy. The advantage of the centrally located DC in the IOD, however, almost disappears if the visiting time at the DC is equivalent to a store visit. In case the single DC is located at the border or outside the delivery area, the strategy PSS-(P) performs considerably better than the IOD.

Another takeaway from Figure 8 is that the impact of visiting time of the DC changes depending on the location of the DC. For the settings in which the DC is at the middle or the corner, decreasing the depot visit time brings considerable benefits to the IOD (see the cases where the pickup time at the DC is one or two times of the store visit). Nevertheless, when the DC is located far away, the marginal benefit of fast DC is visited is negligible.

5.3.2. The impact of delivery lead time and the depot waiting strategy and courier capacity. It is crucial for service providers to balance delivery speed with the service quality for a given fleet size. Also, in this section, we investigate the sensitivity of PSS and IOD operating strategies against the courier capacities and waiting rules at the DC. We choose a setting in which the DC is located at the center of the delivery



(a) Lead time=60 minutes (b) Lead time=90 minutes (c) Lead time=120 minutes

Figure 9 Impact of lead-time, courier capacity and waiting strategy at the DC

area and other parameters are given by Table 4 as otherwise mentioned. We vary the capacity of couriers between one and four orders. We introduce a set of waiting rules for the couriers operating under the IOD strategy. In these new rules, a courier will depart from the depot if one of the following conditions occurs:

- The last moment that the delivery deadline of orders assigned to the courier is not violated,
- The courier's capacity is full after an order is assigned to her

In addition, all couriers waiting at the DC to serve their assigned orders will depart immediately when any arriving order is rejected to be served. To prevent confusion, we name the (default) no waiting strategy *IOD-nw* and the waiting strategy with *IOD-w*. Finally, in Section EC.3 we provide an overview of the performance of our solution approach described in Section 4 for the PSS instances.

Figure 9 presents the ratio of orders served to show the impact of the lead-time varying between 60, 90, and 120 minutes for different courier capacities. Firstly, we can look at the isolated impact of lead-time for the base-case parameters (courier capacity equals two). We see that the ratio of served orders is positively correlated with the lead-time length regardless of the delivery strategy. That is because an instant delivery service can serve more orders when the lead time is longer due to the additional flexibility for travel and shopping/picking per order. When the courier capacity equals two orders, the PSS performs better than the IOD strategy. Another interesting result is the performance of the IOD-w and IOD-nw when courier capacity equals one order. We see that increasing lead times does not improve the served orders ratio. This result indicates that the IOD is bound by the platform capacity and increasing flexibility through longer lead-time does not contribute to the additional performance. Nevertheless, the performance of the PSS in the same scenario presents its resilience in the more restrictive delivery conditions.

Figure 9 also presents the impact of courier capacity and waiting strategies. We see that when the capacity of couriers increases that more orders can be delivered. We observe that the positively correlated pattern between courier capacity and orders served is robust even when the lead-time and/or waiting strategy changes. In addition, the figure shows that the waiting strategy at the DC has no significant impact on the IOD strategy. That is, the default no-waiting at DC performs slightly better than the IOD-w. When we look at the difference between the PSS and the IOD while the courier capacity increases, we observe that the served orders become roughly equal in case the courier capacity equals three orders. That is, the IOD strategy outperforms the PSS if the couriers can serve more than three orders simultaneously.

Figure 9 shows the relationship between the courier capacity and the lead-time. We see that both larger courier capacity and delivery lead-time individually increase the served orders regardless of the operating strategy. Nevertheless, the dual impact of these two parameters is more pronounced for the IOD than on the PSS. Particularly, we observe this effect when the lead time equals 120 minutes and the driver capacity equals four orders. The ratio of served orders with the IOD employed overtakes the case in which the PSS is used. This is a crucial indicator for the platforms as the PSS may not suitable for the settings in which there are longer lead-times and large courier capacities.

Figure 10 Ratio of trip updates per accepted order in the PSS given different courier capacities

In Figure 10, we consider the PSS's ability of rerouting (changing the visited store or collected orders) while accepting new orders. Note that in the PSS, couriers' trips do not have a given start and ending point. Therefore, we define the following rules to consider whether a trip is rerouted or not. A courier's forward trip is considered updated if the sequence of visiting locations until the address of the last customer in the ongoing trip changes after the order acceptance. This trip update may be because of visiting a different store or the visiting order of the nodes has changed. Furthermore, an order acceptance may update multiple trips simultaneously. However, we do not count the case as a trip reroutes if the accepted order's pick-ups and delivery visits are sequenced after the last customer visit. Lastly, we normalize the total trip updates by dividing them by the number of accepted orders during the delivery period.

Figure 10 presents the ratio of trip updates per accepted order in the PSS for varying lead times and courier capacities. Firstly, we see that on average the trip updates per accepted customer order lie between 0.5 and 0.9. This means that roughly every three out of four accepted orders make a change in the ongoing delivery plan. This result shows that real-time trip optimization is key to keeping resource usage effective. Furthermore, we see that the proportion of trip updates has a negative correlation with the lead-time length; i.e., the trip updates more often when the lead-time is short. We believe this is a result of restrictive delivery commitments, such as short lead times, and requiring more trip updates per accepted order as the order acceptance is only possible with the rerouting of the current trips. We observe an opposite pattern for the impact of the couriers' capacity on the number of trip updates. When the lead-time is 90 or 120 minutes, the trip updated per accepted order increases along with the capacity of couriers. As expected, more space in a courier trip allows utilizing the consolidation impact more. Therefore, when the larger capacity is combined with the longer lead-time, the ratio of updates per accepted order increases.

5.4. Case Studies

To study the economic viability of PSSs, we develop case studies using selected urban areas in Asia, Europe, and North America. Instead of concentrating only on the operational metrics such as served orders, we also compute the average workforce cost per served order as an additional performance metric.

As a test-bed, we choose three cities: Rotterdam in Europe, Atlanta in North America, and Singapore in South East Asia. The road networks of these cities are extracted from

Figure 11 **Delivery areas for case cities, blue dots represent supermarkets or convenience stores**

(a) Rotterdam (b) Atlanta (c) Singapore

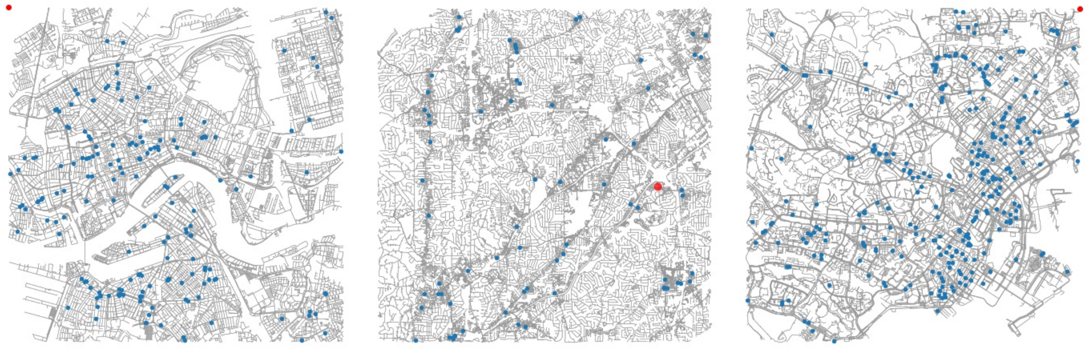


Table 5 **Personal shopping system vs depot-fulfillment: Order with single products, Lead-time: 90 minutes**

Cities	#Stores (Product Availability)	Delivery Area	Strategies	% Served	Cost per served order ($)
Rotterdam	179(20)	4 km × 4 km	PSS-(P)	83.5	5.4
			IOD	51.6	10.7
Atlanta	73(10)	8 km × 8 km	PSS-(P)	60.1	7.3
			IOD	54.1	9.9
Singapore	316(40)	4 km × 4 km	PSS-(P)	92.0	4.9
			IOD	47.7	11.9

the Open Street Map (OpenStreetMap contributors 2017). For each city, the delivery area is a square-shaped region with a known edge length of 4 or 8 kilometers centered around the town's centroid, (see Figure 11 for the delivery areas). We consider all stores within the delivery classified as 'supermarket' or 'convenience' with their exact location. The depots are located near the large grocery delivery providers' warehouses, such as in Rotterdam, the depot is located nearby the neighborhood of Berkel en Rodenrijs, in Atlanta, the depot location and the delivery region is inspired by the study by Stroh et al. (2021), and in Singapore, the depot is located within the Singpost Regional E-commerce Logistics Hub. We use the base-case parameters to compute the times of visiting DC and shopping at a store. Furthermore, in the IOD strategy, the DC employees one picker to prepare the orders for couriers. Couriers (and pickers in the IOD) are got paid 17(16) USD per hour.

Table 5 shows the operational and economical performances of the PSS-(P) and the IOD in three cities. Both PSS-(P) and IOD have a fleet of 3 couriers working 12 hours, and in the IOD one picker in the DC. The total workforce expense of a service period is computed by the number of hired couriers and pickers. We further assume no penalty associated with unserved order.

The results show that the PSS consistently serves more orders with an equivalent number of couriers than the IOD does for all three cities. We further observe that the location of the DC is the most crucial factor for the performance of the IOD as the difference between the PSS and IOD are the smallest in Atlanta, where the DC is closer to the center of the area.

The workforce cost per served order extends the competitiveness of the PSS strategy over the IOD. Table 5 presents that the amount of expenses for serving an order is substantially lower with the PSS strategy than with the IOD. That is the difference in cost of serving an order becomes as big as seven USD in Singapore. Regarding the characteristics of cities, it is possible to argue that areas with denser retail store presence are more advantageous for the PSS.

6. Conclusion

This paper explores instant delivery services in the last-mile logistics by introducing the instant delivery problem (IDP). The IDP effectively represents two last-mile strategies, namely, the personal shopper system (PSS) and, inventory owned delivery (IOD), which differ in choice in the storage facilities. In the PSS strategy, a delivery platform uses the retail stores' assortments in the delivery area without holding or owning inventory to fulfill on-demand customer orders. In the IOD strategy, the platform operates a distribution center along with the delivery organization and uses the products in the DC to fulfill customer orders. We devise a solution approach to the IDP that combines the sequential decision-making framework and column generation. This approach relies on maintaining forward courier trips as an active plan and reusing the ongoing forward trips to generate a new plan for the re-optimization step.

Computational results show that the personal shopper system (PSS) is a compelling alternative for instant delivery services. We specifically explore the service setting in which the PSS platform chooses the stores where the order will be picked/purchased. The results present that the performance of the PSS is sufficient or even superior compared to the DC based IOD under different settings of lead-time, the location of DC, etc. Nevertheless, the competitiveness of the PSS disappears when the order size (the number of store visits) increases or when the couriers have a large capacity to serve multiple customers simultaneously. We also study and report that the store selection mode plays a significant role in shopping/picking consolidation in the PSS systems.

Furthermore, the experiments for the case studies show that our solution approach is suitable for real-life settings.

As personal shopper services and expedited delivery demand are surging across the world, more inspiring applications in this field unlock opportunities for future research. The natural extension of the study is to integrate future information into the decision-making process. Furthermore, assessing the benefits of a few operational strategies such as transfer between couriers, allowing multiple couriers to serve single orders consisting of multiple products, workforce planning, etc. could be potential directions. Another avenue for further research is to develop analytical approximation schemes for personal shopper systems to answer more managerial questions. It is worth noting that the IDP and proposed approach is generic to address other on-demand and dynamic problems as well. Furthermore, it is possible to utilize the instant delivery problem and introduced an operating strategy of product selection in the restaurant meal delivery setting as enhanced service.

Acknowledgments

References

Alicke K, Losch M (2015) Lean and mean: How does your supply chain shape up? URL `http://www.mckinsey.com/~/media/mckinsey`.

Alonso-Mora J, Samaranayake S, Wallar A, Frazzoli E, Rus D (2017) On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences* 114(3):462–467, URL `http://dx.doi.org/10.1073/pnas.1611675114`.

Arslan A (2019) *Operational Strategies for On-demand Delivery Services*. Ph.D. thesis, Erasmus University Rotterdam.

Arslan AM, Agatz N, Klapp MA (2021) Operational strategies for on-demand personal shopper services. *Transportation Research Part C: Emerging Technologies* 130:103320, URL `http://dx.doi.org/10.1016/j.trc.2021.103320`.

Aziez I, Côté JF, Coelho LC (2020) Exact algorithms for the multi-pickup and delivery problem with time windows. *European Journal of Operational Research* 284(3):906–919, URL `http://dx.doi.org/10.1016/j.ejor.2020.01.040`.

Bai J, So KC, Tang CS, 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, URL `http://dx.doi.org/10.1287/msom.2018.0707`.

Behrend M, Meisel F (2018) The integration of item-sharing and crowdshipping: Can collaborative consumption be pushed by delivering through the crowd? *Transportation Research Part B: Methodological* 111:227–243, URL `http://dx.doi.org/10.1016/j.trb.2018.02.017`.

Benjaafar S, Wu S, Liu H, Gunnarsson EB (2021) Dimensioning on-demand vehicle sharing systems. *Management Science* URL `http://dx.doi.org/10.1287/mnsc.2021.3957`, ahead of print.

Bertsimas D, Jaillet P, Martin S (2019) Online vehicle routing: The edge of optimization in large-scale applications. *Operations Research* 67(1):143–162, URL `http://dx.doi.org/10.1287/opre.2018.1763`.

Boyacı B, Zografos KG, Geroliminis N (2017) An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations. *Transportation Research Part B: Methodological* 95:214–237, URL `http://dx.doi.org/10.1016/j.trb.2016.10.007`.

Desaulniers G, Desrosiers J, Solomon MM (2006) *Column generation*, volume 5 (Springer Science & Business Media).

Drexl M, Schneider M (2015) A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research* 241(2):283–308, URL `http://dx.doi.org/10.1016/j.ejor.2014.08.030`.

Grand View Research (2019) Same day delivery market size, share & trends analysis report. URL `https://www.grandviewresearch.com/industry-analysis/same-day-delivery-market`.

Guda H, Subramanian U (2019) Your Uber is arriving: Managing on-demand workers through surge pricing, forecast communication, and worker incentives. *Management Science* 65(5):1995–2014, URL `http://dx.doi.org/10.1287/mnsc.2018.3050`.

Harrington D (2015) From first mile to last mile global industrial & logistics trends. Collier International.

Karaoglan I, Altiparmak F, Kara I, Dengiz B (2012) The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach. *Omega* 40(4):465–477, URL `http://dx.doi.org/10.1016/j.omega.2011.09.002`.

Klapp MA, Erera AL, Toriello A (2018) The dynamic dispatch waves problem for same-day delivery. *European Journal of Operational Research* 271(2):519–534, URL `http://dx.doi.org/10.1016/j.ejor.2018.05.032`.

Lyu G, Cheung WC, Teo CP, Wang H (2019) Multi-objective online ride-matching. *SSRN Electronic Journal* URL `http://dx.doi.org/10.2139/ssrn.3356823`.

Montoya-Torres JR, Franco JL, Isaza SN, Jiménez HF, Herazo-Padilla N (2015) A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering* 79:115–129, URL `http://dx.doi.org/10.1016/j.cie.2014.10.029`.

Naccache S, Côté JF, Coelho LC (2018) The multi-pickup and delivery problem with time windows. *European Journal of Operational Research* 269(1):353–362, URL `http://dx.doi.org/10.1016/j.ejor.2018.01.035`.

Nemhauser GL, Wolsey LA (1988) *Integer programming and combinatorial optimization*, volume 191 (Springer).

OpenStreetMap contributors (2017) Planet dump retrieved from https://planet.osm.org. `https://www.openstreetmap.org`.

Özkan E, Ward AR (2020) Dynamic matching for real-time ride sharing. *Stochastic Systems* 10(1):29–70, URL `http://dx.doi.org/10.1287/stsy.2019.0037`.

Pei Z, Dai X, Yu T, Zhao L, He QC (2022) Dynamic Rebalancing Strategy in Free-Float Bicycle Sharing Systems: Orbit Queues and Two-Sided Matching. *Service Science* 14(1):35–59, URL `http://dx.doi.org/10.1287/serv.2021.0287`.

Pillac V, Gendreau M, Guéret C, Medaglia AL (2013) A review of dynamic vehicle routing problems. *European Journal of Operational Research* 225(1):1–11, URL `http://dx.doi.org/10.1016/j.ejor.2012.08.015`.

Powell WB (2019) A unified framework for stochastic optimization. *European Journal of Operational Research* 275(3):795–821.

Reyes D, Erera A, Savelsbergh M, Sahasrabudhe S, O'Neil R (2018) The meal delivery routing problem. Optimization Online.

Ritzinger U, Puchinger J, Hartl RF (2016) A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research* 54(1):215–231, URL `http://dx.doi.org/10.1080/00207543.2015.1043403`.

Savelsbergh M, Van Woensel T (2016) 50th anniversary invited article-city logistics: Challenges and opportunities. *Transportation Science* 50(2):579–590, URL `http://dx.doi.org/10.1287/trsc.2016.0675`.

Schrotenboer AH, Broek MAJ, Buijs P, Ulmer MW (2021) Fighting the e-commerce giants: Efficient routing and effective consolidation for local delivery platforms, arXiv preprint arXiv:2108.12608.

Steever Z, Karwan M, Murray C (2019) Dynamic courier routing for a food delivery service. *Computers & Operations Research* 107:173–188, URL `http://dx.doi.org/10.1016/j.cor.2019.03.008`.

Stroh AM, Erera AL, Toriello A (2021) Tactical design of same-day delivery systems. *Management Science* URL `http://dx.doi.org/10.1287/mnsc.2021.4041`, ahead of print.

Taylor TA (2018) On-demand service platforms. *Manufacturing & Service Operations Management* 20(4):704–720, URL `http://dx.doi.org/10.1287/msom.2017.0678`.

Uit het Broek MAJ, Schrotenboer AH, Jargalsaikhan B, Roodbergen KJ, Coelho LC (2021) Asymmetric multidepot vehicle routing problems: Valid inequalities and a branch-and-cut algorithm. *Operations Research* 69(2):380–409, URL `http://dx.doi.org/10.1287/opre.2020.2033`.

Ulmer MW (2017) Delivery deadlines in same-day delivery. *Logistics Research* 10(3):1–15.

Ulmer MW (2020) Dynamic pricing and routing for same-day delivery. *Transportation Science* 54(4):1016–1033, URL http://dx.doi.org/10.1287/trsc.2019.0958.

Ulmer MW, Goodson JC, Mattfeld DC, Thomas BW (2020a) On modeling stochastic dynamic vehicle routing problems. *EURO Journal on Transportation and Logistics* 100008, URL http://dx.doi.org/10.1016/j.ejtl.2020.100008.

Ulmer MW, Thomas BW, Campbell AM, Woyak N (2020b) The restaurant meal delivery problem: dynamic pickup and delivery with deadlines and random ready times. *Transportation Science* URL http://dx.doi.org/10.1287/trsc.2020.1000.

Ulmer MW, Thomas BW, Mattfeld DC (2019) Preemptive depot returns for dynamic same-day delivery. *EURO journal on Transportation and Logistics* 8(4):327–361, URL http://dx.doi.org/10.1007/s13676-018-0124-0.

Voccia SA, Campbell AM, Thomas BW (2019) The same-day delivery problem for online purchases. *Transportation Science* 53(1):167–184, URL http://dx.doi.org/10.1287/trsc.2016.0732.

Wang H, Yang H (2019) Ridesourcing systems: A framework and review. *Transportation Research Part B: Methodological* 129:122–155, URL http://dx.doi.org/10.1016/j.trb.2019.07.009.

Yildiz B, Savelsbergh M (2019) Provably high-quality solutions for the meal delivery routing problem. *Transportation Science* 53(5):1372–1388, URL http://dx.doi.org/10.1287/trsc.2018.0887.

Courier Dispatch Problem

EC.1. The Network Flow Formulation for the CDP.

The *Courier Dispatch Problem* (CDP) is a static optimization problem that attempts to find a new delivery plan to accommodate the arriving order o at decision epoch a_o. The state of the system at the new order arrival, $S_{a_o} = (\mathcal{O}_{a_o}, P_{a_o}, \Theta_{a_o})$, consists of O_{a_o}, the information of the active orders accepted in previous epochs but are not served by a_o yet, P_{a_o}, the list of products that need to be collected for the active orders, and Θ_{a_o}, the current delivery plan for couriers. Since the CDP is defined the same for every decision epoch, without loss of generality we omit subscript a_o from all parameters and variables from here on.

The CDP is presented on an activity graph $G = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \mathcal{V}^M \cup \mathcal{V}^O \cup \xi$ is the set of vertices representing the storage- and customer-locations, and an additional fictitious terminal vertex and $\mathcal{A}$ defines the arc set. Since couriers are not rerouted when traveling between vertices, $v_k \in \mathcal{V}^{\mathcal{K}} \subseteq \mathcal{V}$ refers to the location of the courier in the current delivery plan, from where courier k's forward trip can be altered. That is, courier k's immediate activities at the current epoch are frozen against the possible updates before she reaches her corresponding courier node v_k. Finally, the fictitious terminal vertex ensures that every courier trip has a given ending location.

Let $\mathcal{O}$ be the set of not yet delivered orders, P be the uncollected products of orders in $\mathcal{O}$ and $P_{\hat{o}}$ be the uncollected products of order $\hat{o} \in \mathcal{O}$. For each uncollected product $p \in P_{\hat{o}}$ of an order $\hat{o}$, we define a unique set of vertices, denoted by $\mathcal{V}^p$ that represents the storage locations in which product $p \in P_{\hat{o}}$ is available and can be picked. Consequently, we define a set $\mathcal{V}^P = \bigcup_{\hat{o} \in \mathcal{O}} \bigcup_{p \in P_{\hat{o}}} \mathcal{V}^p$, representing all unique vertices associated with each possible location where a product can be collected for an order in $\mathcal{O}$. Furthermore, the set $\mathcal{V}^{\mathcal{O}}$ consists of vertices associated with the customer locations of each undelivered order $\hat{o} \in \mathcal{O}$. Let mapping $l : \mathcal{O} \mapsto \mathcal{V}^{\mathcal{O}}$ link an order $\hat{o}$ to its associated node and the reverse function $l^{-1} : \mathcal{V}^{\mathcal{O}} \mapsto \mathcal{O}$ links the customer location to the order. Furthermore, we denote $z^k_{v_k}$ as the earliest moment of time that courier k can depart from the vertex v_k.

Note that we account for the shopping time through arcs, and therefore $t(i, j)$ denotes shopping time if node i and/or j are an element of $\mathcal{V}^P$. For the case of a fixed visiting duration per store and a constant pick-up duration per product, we can write $t(i,j)$ as follows:

$$t(i,j) := \begin{cases} w_{ij}, & \text{if } i \overset{\text{L}}{\neq} j, \\ \triangle f_m(0), & \text{if } j = v_m \wedge i \neq v_m, \\ \triangle f_m(n), & \text{if } j = v_m \wedge i = v_m, \end{cases} \tag{EC.1}$$

where $\triangle f_m(n) := f_m(n) - f_m(n-1), n \geq 1$, represent the incremental shopping time for additional task picking (see Section 3.2). Notice that $\triangle f_m(1)$ considers the fixed cost for visiting store m. Also, v_m denotes the node associated with store m.

Let x_{ij}^k be the binary variable if courier $k \in \mathcal{K}$ traverses on arc $(i,j) \in \mathcal{A}$. Let z_i^k be the departure time of courier k from vertex i and let $\hat{q}_i^k$ be the load size in the number of product of the courier k before departing at node i. Note that we changed our load and capacity definition to a more general case of number of products instead of number of orders. Auxiliary variables need to be introduced to model the number of orders the courier is currently working on. Then, the mixed integer programming formulation of the CDP is written as follows:

$$\min \quad \sum_{k \in \mathcal{K}} \sum_{i,j \in \mathcal{A}} t(i,j) x_{ij}^k \tag{EC.2}$$

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}^p} x_{ij}^k = 1 \qquad p \in P_{\hat{o}}, \ \hat{o} \in \mathcal{O} \tag{EC.3}$$

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}} x_{ij}^k = 1 \qquad j \in \mathcal{V}^{\mathcal{O}} \tag{EC.4}$$

$$\sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{V}} x_{ij}^k = \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{V}} x_{ji}^k \qquad j \in \mathcal{V}^{\mathcal{O}} \cup \mathcal{V}^{\mathcal{P}} \tag{EC.5}$$

$$\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}^{p_j}} x_{ij}^k - \sum_{i \in \mathcal{V}} \sum_{l \in \mathcal{V}^{p_l}} x_{il}^k = 0 \qquad p_j, p_l \in P_{\hat{o}}, \ \hat{o} \in \mathcal{O}, \ k \in \mathcal{K} \tag{EC.6}$$

$$\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}^p} x_{ij}^k - \sum_{i \in \mathcal{V}} x_{i,l(\hat{o})}^k = 0 \qquad p \in P_{\hat{o}}, \ \hat{o} \in \mathcal{O}, \ k \in \mathcal{K} \tag{EC.7}$$

$$\sum_{j \in \mathcal{V}} x_{v_k j}^k = 1 \qquad k \in \mathcal{K} \tag{EC.8}$$

$$\sum_{i \in \mathcal{V}} x_{i\xi}^k = 1 \qquad k \in \mathcal{K} \tag{EC.9}$$

$$z_j^k \geq z_i^k + t(i,j) - M(1 - x_{ij}^k) \qquad (i,j) \in \mathcal{A}, \ k \in \mathcal{K} \tag{EC.10}$$

$$z^k_{l(\hat{o})} \leq d_{\hat{o}} + M(1 - \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}} x^k_{i,l(\hat{o})}) \qquad \hat{o}\in\mathcal{O} \qquad \text{(EC.11)}$$

$$\hat{q}^k_j = \hat{q}^k_i + \begin{cases} x^k_{ij}, & \text{if } i\in\mathcal{V}^P \\ -|P_{l^{-1}(i)}|x^k_{ij} & \text{if } i\in\mathcal{V}^{\mathcal{O}} \end{cases} \qquad (i,j)\in\mathcal{A},\ k\in\mathcal{K} \qquad \text{(EC.12)}$$

$$\hat{q}^k_i \leq \hat{q}_k \qquad i\in\mathcal{V},\ k\in\mathcal{K} \qquad \text{(EC.13)}$$

$$x^k_{ij}\in\{0,1\}, q^k_i\in\mathbb{Z}_+, z^k_i \geq 0 \qquad (i,j)\in\mathcal{A},\ k\in\mathcal{K} \qquad \text{(EC.14)}$$

The objective function in (EC.2) minimizes the total travel and picking (and shopping) time of shoppers. The travel time on arc i,j is given in a functional form as it also includes non-linear shopping time at a store. We refer Arslan et al. (2021) on how to linearize this function.

Constraints (EC.3) ensures that each product should be collected only from one of the stores selling the product. Constraints (EC.4) guarantee that the courier drops off the products at each order's address. Constraints (EC.5) ensure the flow conversation. Constraints (EC.6) and (EC.7) ensure that the same courier collects all products for the same order and eventually drops off them at the delivery address of the order. Constraints (EC.8) and (EC.9) ensure that a trip for each courier starts at the current location and ends at the terminal node. Constraint set (EC.10) and (EC.11) guarantee to acknowledge travel times between activities and orders are delivered before deadlines. Note that M denotes a big number and $d_{\hat{o}}$ the delivery deadline of order $\hat{o}$. Recall that $d_{\hat{o}}$ is the delivery deadline of order $\hat{o}$. The last set of constraints (EC.12) and (EC.13) ensures the maximum number of products that a courier can serve simultaneously is not violated.

EC.2. Solution Approach to the CDP

The forward courier trips for the CDP associated with state S_{a_o} that is of an exponential size and enumerating all of them may not be practical. Therefore, we devise a generation method, which starts to solve a restricted master problem (RMP) with a restricted set of forward trips, denoted by $\Theta^{'}_{a_o} \subset \Theta_{a_o}$. The RMP's solution is optimal if there is not a forward trip in $\Theta_{a_o}\backslash\Theta^{'}_{a_o}$ with negative reduced cost. To identify whether a forward trip with negative reduced cost in $\Theta_{a_o}\backslash\Theta^{'}_{a_o}$, we solve the *pricing problem.* We add those forward trips into the restricted forward trip set and resolve the RMP. This procedure is continued until no forward trip with negative reduced cost remains.

The associated pricing problem of the CDP is the resource constraint shortest path problem (RCSRP). Let μ_o be the dual variable associated with Constraints (3b) and κ_k be the dual variable associated with Constraints (3c) obtained by solving the linear relaxation of the RMP. Then, the reduced cost of the forward trip θ for courier k, denoted by π_θ^c is:

$$\tau_\theta^c = \tau_\theta - \sum_{o \in \mathcal{O}_{a_o}(o)} \mu_o - \kappa_k.$$

The pricing problem is to find the forward trip with a minimum τ_θ for each courier $k \in \mathcal{K}$.

We adopt a simple enumeration algorithm to identify/generate forward trips with negative reduced costs in a greedy fashion. That is, for each courier $k \in \mathcal{K}$, we initially identify the first activity of the ongoing trip and mark it as the starting vertex and remain the committed activities. The following path of the courier is extended by adding new activities based on the cheapest insertion heuristic and considering the preceding relationship of the pick-ups and drop-offs and delivery deadlines of the orders. After finding at most $|\mathcal{K}| \times |\mathcal{O}_{a_o}|$ forward trips, we solve the RMP again and continue adding columns until no forward trips with negative reduced cost can be found by the pricing problem. Solving both linear relaxation of the RMP and the RMP will provide a lower- and upper-bound for the CDP. Finally, the optimal solution can be obtained by running the pricing problem once again and adding any forward trip for which the reduced cost are smaller than or equal to the difference between the lower- and upperbound. Solving the RMP on the this set of columns will provide the optimal integer solution for the CDP.

EC.3. Performance Statistics for the Rolling Horizon Approach

In this section, we analyze the performance of the rolling horizon approach used to solve the CDP as described in Section 4.2. For only the PSS instances studied in Section 5.3.2, Table EC.1 shows the performance statistics for varying courier capacities and different order due times. Column $\Delta(\%)$ shows the average and standard deviation of the percentile difference between the upper and lower bounds for the CDP obtained after no more forward trips with negative reduced cost can be found by the pricing algorithm. We can conclude that our solution approach can find bounds that are almost equal to each other. As such the percentile difference between the upper bound and final (integer) solution found, as reported in column Δ^f (%) is again small. Only when the capacity is small and the order due times are long, these gaps become larger as in this case there will be more non-committed orders that can still be swapped between different courier forward trips. The last two columns, # of columns and Time (sec.) present the average and standard deviation of the number of columns generated per decision epoch and the computational time for generating the columns and solving the model on an Intel Core i7 with 16 GB of RAM. From here we can see that minimal effort is required to resolve the CDP at each epoch. This is due that our approach can reuse the current forward trip from the previous epoch which allows us to quickly find the new optimal solution when a new order arrives.

Table EC.1 Performance statistics for the CDP for PSS instances.

Capacity (#)	Due mean (min.)	Δ (%)		Δ^f (%)		# of columns		Time (sec.)	
1	60	0.01	(±0.02)	0.01	(±0.01)	2.08	(±0.29)	0.00	(±0.00)
2	60	0.19	(±0.12)	0.01	(±0.02)	2.52	(±0.55)	0.00	(±0.00)
3	60	0.15	(±0.03)	0.02	(±0.03)	2.43	(±0.27)	0.00	(±0.00)
4	60	0.11	(±0.07)	0.01	(±0.01)	2.75	(±0.67)	0.00	(±0.00)
1	90	0.12	(±0.04)	0.10	(±0.10)	11.83	(±1.56)	0.01	(±0.00)
2	90	0.20	(±0.11)	0.03	(±0.05)	4.82	(±0.52)	0.01	(±0.00)
3	90	0.21	(±0.07)	0.05	(±0.07)	5.12	(±0.89)	0.01	(±0.00)
4	90	0.25	(±0.13)	0.04	(±0.05)	5.41	(±0.95)	0.01	(±0.00)
1	120	0.26	(±0.11)	0.17	(±0.09)	28.79	(±3.93)	0.05	(±0.01)
2	120	0.22	(±0.09)	0.04	(±0.02)	6.76	(±1.27)	0.01	(±0.00)
3	120	0.25	(±0.11)	0.07	(±0.05)	6.68	(±1.71)	0.01	(±0.00)
4	120	0.23	(±0.10)	0.03	(±0.04)	6.78	(±1.25)	0.01	(±0.00)