The adoption of self-driving delivery robots in last mile logistics

Abstract: Covid-19, the global pandemic, has taught us the importance of contactless delivery service and robotic automation. Using self-driving delivery robots can provide flexibility for on-time deliveries and help better protect both driver and customers by minimizing contact. To this end, this paper introduces a new vehicle routing problem with time windows and delivery robots (VRPTWDR). With the help of delivery robots, considerable operational time savings can be achieved by dispatching robots to serve nearby customers while a driver is also serving a customer. We provide a mathematical model for the VRPTWDR and investigate the challenges and benefits of using delivery robots as assistants for city logistics. A two-stage matheurisitic algorithm is developed to solve medium scale VRPTWDR instances. Finally, results of computational experiments demonstrate the value of self-driving delivery robots in urban areas by highlighting operational limitations on route planning.

Keywords: City logistics; Vehicle routing problem; Self-driving delivery robot; contactless delivery; Matheuristic algorithm

1. Introduction

The global e-commerce market has greatly changed the way businesses operate. The total e-commerce revenue is expected to grow to EUR 2,568.8 billion by 2023 (Striapunina, 2019), which has inevitably led to increased volume of global parcel delivery market. As the final step of transportation, last mile delivery has been a key success factor to achieve high customer satisfaction and increased market share for Logistics Service Providers (LSPs) around the world.

Due to increased traffic congestion, parking area limitations and environmental regulations, last mile delivery in populated urban areas is facing enormous difficulties (Akeb et al., 2018). As a result, new last mile delivery solutions have emerged in the last decade. As studied by Melo (2017), combining electric cargo bikes with delivery vans leads to a better traffic performance. In another application, Freight Traffic Control 2050 project team (El Hachemi et al., 2013) identified the potential benefits of portering in central London to alleviate congestion and associated negative environmental impact.

The evolution of automation technology during the last decade has seen great progress and opened the door for numerous innovative business applications in last mile logistics. More specifically, automated goods delivery is forecasted to provide a reasonable answer for up to 80 percent of all Business-to-Customers (B2C) deliveries (Grolms, 2019). There are already real-life experiments where unmanned aerial vehicles (UAVs) are being used by pioneering companies, including DHL (DHL, 2016), SF Express (Shields, 2018), Amazon (Vincent and Gartenberg, 2019), Google (BBCNEWS, 2019), and UPS (McFarland, 2019). However, besides its lack of cost-competitiveness, low-carrying capability, and flying range and legislation restrictions in urban areas, UAVs cannot perform deliveries in certain environments. Another application of last mile delivery service is the autonomous guided vehicle (AGV) with lockers that can deliver parcels without any human intervention. Customers are notified of the exact arrival time and asked to pick up a parcel from a specified locker mounted on a vehicle (Bouton, et al., 2017). In the application of AGV delivery, security is the biggest concern, especially in complex traffic environments in urban areas. So, efforts on seeking alternative services for the last mile delivery in urban areas continue. The self-driving delivery robot is a promising kind of autonomous delivery mode, which can cover limited areas. Before the Covid-19 pandemic, the sight of robots delivering customers' parcels would have seemed futuristic. However, Starship Technologies, the San Francisco-based firm, is currently running a delivery robot service in the north of London. Earlier in 2020, the company has also launched this new delivery system in six new cities, including a grocery delivery service in Washington, D.C. In **Fig. 1**, we present three different types of delivery robots that are available in the market.

Starship robot

(Source: starship.xyz)

ANYmal

(Source: anybotics.com)

FedEx SameDay Bot

(Source: van.fedex.com)

Fig. 1. Pictures of three different delivery robots.

The features of these robots are best suited for the last mile delivery in the context of city logistics. For example, weighing no more than 45 kg, the Starship robot can travel at pedestrian speed and carry a payload of less than 2.6 kg to customers within a 4-mile radius (Kottasova, 2016). In another example, besides up to 10 kg payload capacity, ANYmal can climb over challenging terrains such as curbs, stairs (up to 45 degrees), and other obstacles on the ground.

Moreover, ANYmal travels at a speed of 3.6 km/h with batteries for more than two hours of autonomy and can ring the doorbell with its foot and drop the package in front of the door (Hutter et al., 2017). As a last example, designed for B2C door-to-door deliveries, the FedEx SameDay Bot has a top speed of 16 km/h with a three-mile radius. The self-driving robot delivers parcels to doorsteps of customers at relatively slow speeds using sidewalks make sure their security implications are not a big concern (Kottasova, 2016). Therefore, delivery robots powered electronically can provide a cheaper, safer, and greener solution to current unsustainable last mile challenge.

Moreover, as robots may be a slow option to drive all the way from a distribution hub, it can be a good substitute of bike couriers, being used for instant deliveries in urban areas where delivery vans are inefficient. A specific type of delivery van could be used to ferry robots from neighborhood to neighborhood where it is fastest to make deliveries. Then, robots take over for the final step of a delivery. For example, Mercedes-Benz is partnering with Starship Technologies on a futuristic electric van that features delivery robots (McFarland, 2016). The electric van is designed as a "mothership" system, which holds several ground robots. The idea is that the mothership parks in a neighborhood and deploys robots to make nearby deliveries. After the robots have made their deliveries, they return to the van, drive up a ramp and then be driven to another neighborhood. Given certain tasks that require a human touch, Starship Technologies still plans to use human driver in a vehicle. Rather than the traditional model of a truck completing one delivery at a time, a delivery van can complete multiple deliveries at a time with the help of delivery robots. Following the same idea, this paper aims at investigating the related routing problem, named as Vehicle Routing Problem with Time Windows and Delivery Robots (VRPTWDR).

Our contributions are as follows. First, we study a promising and futuristic routing problem and investigate where and how it can be utilized. Second, we introduce a mixed-integer linear programming model for the VRPTWDR. The proposed model can be used to obtain good quality solutions for small-sized instances for benchmarking the solution quality of advanced solution methods in further studies. Third, given that the delivery robot is a novel technology, this study examines several features including customer geographic distribution, widths of customers' time windows, and robot's moving speed and service coverage radius. Finally, a two-stage matheuristic algorithm is proposed to solve medium-sized instances.

The rest of this paper is organized as follows. Section 2 briefly reviews the related literature. Section 3 defines the problem with mathematical formulations whereas section 4 discusses the important design features of the VRPTWDR. A matheuristic algorithm is presented in section

5. Computational experiments and related analyses are provided in Section 6. And finally, section 7 provides the summary of this study and lists possible future research directions.

2. Literature review

The VRPTWDR is a generalization of classical VRP in which a set of vehicle routes are determined to serve a given set of customers. A summary of various types of VRPs, their corresponding mathematical models, and solution methodologies can be found in Golden et al. (2008) and Braekers et al. (2015). Some variants of VRP share common features with the VRPTWDR (e.g., time windows (Cordeau et al., 2000), two types of vehicles (Wang and Sheu, 2019), and two echelon routes (Perboli et al., 2011)). However, the VRPTWDR differs from these studies in that we focus on newest technological features.

The two-echelon VRP (2E-VRP) considers two types of routes with two set of fleets, but customers in each echelon are known in advance (Perboli et al., 2011). In the VRPTWDR, customers are not distinguished for different kinds of routes in advance. The problem is to determine which customers are visited by vehicles and which nearby customers are visited by robots dispatched from a customer visited by a driver.

The truck and trailer problem (TTRP) also requires routing two types of vehicles as there are customers with accessibility constraints who must be served by a vehicle, and others who can be served either by a truck or by a complete vehicle (a truck pulling a trailer) (see, e.g., Villegas et al., 2013; Rothenbächer et al., 2018). Nevertheless, delivery robots travel to and return from their targeted customers independently, while a trailer can only be moved by connecting it to a truck.

In Nguyễn et al. (2018), a two-level clustered TSPTW was addressed for a single route that consists of a set of intra-cluster walking routes. Given predefined clusters, the proposed model determines the visit sequence of each cluster, as well as that within each cluster. However, the clustering influences generated routes greatly. In addition, multiple robots can conduct more efficient deliveries than one driver.

As the evolution of UAV technology develops quickly, research on routing problem for the vehicle and UAV tandem has increased in recent years and made significant progress. There are two types of application patterns of vehicle-UAV delivery discussed in the literature: flying sidekick and mothership systems. The flying sidekick system uses the dispatch-move-collect tactic, which means that vehicles move to another location after dispatching UAV(s) and UAVs are collected by the same or another vehicle from a different place. In the mothership system, dispatch-wait-collect tactics are usually applied, where a vehicle dispatches UAV(s) at a location and waits at the same location to collect them. Murray and Chu (2015) first introduced a flying sidekick TSP for the last mile delivery scenario in which a UAV works in collaboration with a traditional delivery truck. Later, more practical constraints were incorporated, such as payload-dependent flight and duration and restricted flying areas (Jeong et al., 2019). In another study, Luo et al. (2017) investigated a cooperated routing problem for the ground vehicle with UAVs, which extended the routing problem of the UAV and vehicle tandem to multiple vehicle situations. Wang and Sheu (2019) extended the studied routing problem to more generic scenarios, including multiple UAVs with parallel flying and landing at known stations. Meanwhile, research on efficient heuristic algorithms for the routing problems in the flying sidekick system were studied by several researchers (see, e.g., Sacramento et al., 2019; Schermer et al., 2019).

The existed studies on mothership systems generally assume that all customers are served by the delivery assistant (e.g., drones or robots). Some of them incorporate stations for dispaching and retrieving robots. For example, Boysen et al. (2018) investigated a truck-based robot delivery concept: all customers are served by robots, and a truck loads the shipments for customers at a central depot where the goods to be shipped are stored as well as several robots with a fixed reserved capacity. When a drop-off point is reached, robots are launched to autonomously deliver goods to customers. Unlike other tandem delivery concept, the truck takes no responsibility of collecting robots. Instead, a set of decentralized robot depots (stations) is introduced to retrieve robot and for trucks to replenish robots. Only a single truck is considered, and no shortage of robots are assumed in the study. In the study of Karak and Abdelghany (2019), a specific type of mothership system is investigated, which allows a “swarm” dispatching approach at stations and considers integrated pick-up and delivery. Vehicles transport UAVs and shipments among stations and UAVs can only be dispatched and collected at a station.

Other studies assume that the mothership can launch and retrieve the assistant anywhere in a continuous space. Chang and Lee (2018) considered a specific type of mothership system. They proposed an approach to solve the routing problem arising in a type of mothership system with dispatch-wait-collect tactics. First, nearby delivery locations within the UAV’s service range are clustered using the K-means clustering technique. Then, a truck’s delivery route among the centers of clusters is set up to minimize its traveling time using a TSP algorithm. Finally, a nonlinear programming model is applied to find shift-weights that move the centers of clusters to make for wider UAV-delivery areas and shorter truck-routes after the initial K-means clustering and TSP solution. Poikonen and Golden (2020) studied a truck-and-drone routing problem named as the mothership and drone routing problem (MDRP) which contains one mothership truck and one drone. In this study, the mothership can be a large ship or airplane with the ability to move in Euclidean space and launch or retrieve a drone at any location. However, finding an optimal parking site in a Euclidean plane with continuous variable may not be feasible in populated urban areas.

Overall, time window constraints are excluded in most truck-drone routing problems and most studies considered only a single truck route. The differences between the ground robot and the UAV in terms of features in their applications lead to distinct routing problems. The UAV travels at a much higher speed than robots, while the number of ground robots that fit in a single van (e.g., up to eight for Mercedes-Benz electronic van) is much higher than UAVs (only one UAV is usually assumed to be with a single van). As a result, the UAV is more suitable in rural areas and delivery robots have distinct advantages in last mile delivery within urban areas. On the other hand, as an emerging application, different concepts lead to different routing problems. Compared to research efforts that are underway to improve operational aspects to enable the delivery with UAVs, less attention has been given on the operational challenges associated with leveraging delivery robot technology. In the application of self-driving robots as a last mile delivery assistant, a hybrid type of delivery system combining the features of sidekick and mothership can be formed. There is no need to build stations or docks for dispatching robots, the delivery van parks at a customer's parking site and dispatches robots. A driver serves a customer at the parking site while robots can visit other nearby customers. Considering their distance-limited radius, the delivery van needs to dispatch and collect robot(s) at the same location. Therefore, the VRPTWDR differs from the operational perspective studied in published literature.

Given the complexity of routing problems, exact algorithms show their shortcomings on medium- and large-sized instances. As a result, many researchers developed various heuristic algorithms to solve routing problems arising in practice. The heuristic algorithms made by the interoperation of metaheuristics and mathematical programming techniques are named as matheuristic algorithms (Boschetti et al., 2010). The matheuristic algorithm is based on multiple calls of a MILP model, which solves an initial solution of the problem to optimality. Even though there are different types of implementations with matheuristic algorithms, using mathematical solvers as a part of the solution methodology can bring several advantages. First, matheuristic algorithm is a framework for the design of mathematically sound heuristics. Second, a simplification of the mathematical model can be solved more efficiently and does not deteriorate solution times. And finally, these algorithms can reduce the need of parameters used in metaheuristics algorithms. Interested readers are referred to the existing literature (see, e.g., Ghiami et al., 2019; Keskin and and Çatay 2018; Villegas et al., 2013).

3. Problem description

The VRPTWDR is defined on a graph $G=(V, A)$, where V is the set of nodes and $A=\{(i, j) | i, j \in V, i \neq j\}$ is the set of arcs. The set of nodes $V=\{0\} \cup C$ contains the depot node (0) and the customer set $C=\{1, 2\ldots, n\}$. Each customer i has a predefined hard time window $[l_i, u_i]$ for delivery service. The delivery outside the time window is not allowed by the customer. Each customer i has a known demand q_i that must be satisfied either by a vehicle (driver) or a robot (if possible). Whether a customer accepts a robot service is known in advance, which is indicated by a binary parameter f_i^d: equal to 1 if customer i can be served by a robot, and 0 otherwise. There are several situations when a customer cannot be visited by a robot, such as technological inaccessibility of its location, its preference, and capacity/radius limitation of the robot.

In our problem setting, a set of homogeneous vehicles $K=\{1, 2, \ldots, k\}$ and a set of homogeneous delivery robots $DR=\{1, 2,\ldots, m\}$ are initially located at the depot. The capacity of a vehicle is denoted by Q. The distance from customer i to j is represented by d_{ij}. We denote traveling speeds on arc (i, j) as vel_{ij}^v and vel_{ij}^d for vehicle and robot, respectively. Hence, the traveling times on arc (i, j) are calculated as $t_{ij}^v=d_{ij}/vel_{ij}^v$ and $t_{ij}^d=d_{ij}/vel_{ij}^d$, respectively. And finally, s_i^v and s_i^d are used to represent service times for vehicle and robot, respectively.

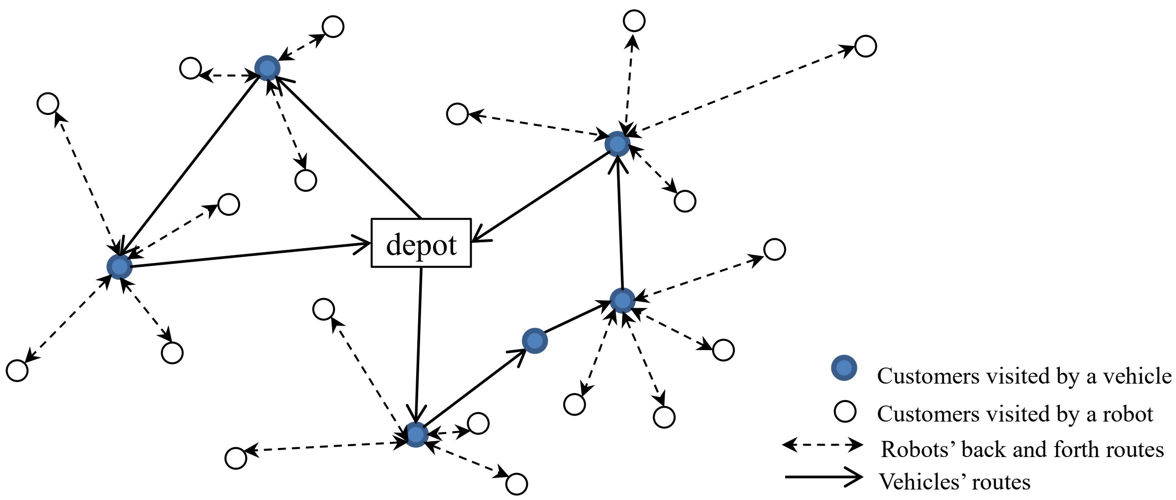

Fig. 2. An illustration of a feasible solution with 25 customers, two vehicles and four robots available in each vehicle.

As illustrated in **Fig. 2**, a feasible solution to the VRPTWDR needs to involve a set of integrated decisions, including how many vehicles are used and their routes, which customers to be visited by vehicle(s) and where and when to dispatch robots to visit which nearby customers. It can be easily adapted to a scenario where an unmanned van is used and all customers are served by robots through the setting possible parking locations as dummy customers. Moreover, the VRPTWDR is more complicated than the standard VRPTW, as there are two types of routes and both need to satisfy related capacity and time window constraints. The VRPTWDR is still a more general problem than that of VRPTW. For example, if the number of available robots in a van is set to zero, the problem at hand turns to be a standard VRPTW.

In addition to the above, the following assumptions are considered to specify the operational scenarios of vehicle-robot delivery system: (*i*) multiple self-driving robots are mounted on a single vehicle; (*ii*) some customers can only be served by a vehicle (driver), while others can be served by a vehicle or a robot; (*iii*) vehicles dispatch robots at parking places and wait for them back at the same place before moving to the next customer; (*iv*) the robot serves one customer per sortie, while multiple robots can be dispatched simultaneously to perform a swarm of deliveries; (*v*) the robot has a maximum radius, denoted by r^d; (*vi*) each robot can only be sent once from a certain parking place; and finally, (*vii*) en-route battery charging and battery replacement are assumed.

Therefore, the decision variables are defined as follows. Binary variables y_{ij}^{dk} equal 1 if robot d installed in vehicle k is dispatched at customer i to serve customer j, and 0 otherwise. Binary variables x_{ij}^k equal to 1 if vehicle k travels from node i to node j, and 0 otherwise. Continuous variables p_{ij}^k show the payload of vehicle k when it travels from node i to node j. Continuous variables a_i represent the arriving time of the visiting resource at customer i, either a vehicle or a robot. Moreover, continuous variables b_i represent the service start time of customer i. And finally, w_i show the waiting time of a vehicle at customer i's parking site from its service start time.

The problem at hand is modeled in the form of a MILP formulation as presented below ([P1]).

[P1]
$$\text{minimize} \quad = \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} x_{ij}^k \cdot t_{ij}^v + \sum_{i\in C}(b_i - a_i + w_i) \tag{1}$$

Subject to

$$\sum_{i\in V}\sum_{k\in K} x_{ij}^k + \sum_{i\in C}\sum_{d\in DR}\sum_{k\in K} y_{ij}^{kd} = 1, \forall j \in C \tag{2}$$

$$\sum_{i\in C}\sum_{k\in K}\sum_{d\in DR} y_{ij}^{kd} <= f_j^d, \forall j \in C \tag{3}$$

$$\sum_{j\in C}\sum_{d\in DR} y_{ij}^{kd} \leq |DR| \sum_{j\in V} x_{ji}^{k}, \forall i \in C, k \in K \tag{4}$$

$$\sum_{j\in C}\sum_{k\in K} y_{ij}^{kd} \leq 1, \forall i \in C, d \in DR \tag{5}$$

$$\sum_{j\in C} x_{0j}^{k} \leq 1, \forall k \in K \tag{6}$$

$$\sum_{i\in V} x_{ij}^{k} = \sum_{i\in V} x_{ji}^{k}, \forall j \in V, k \in K \tag{7}$$

$$\sum_{i\in V}\sum_{k\in K} p_{ij}^{k} - \sum_{i\in V}\sum_{k\in K} p_{ji}^{k} = q_j * \sum_{i\in V}\sum_{k\in K} x_{ij}^{k} + \sum_{i\in C}\sum_{k\in K}\sum_{d\in DR} q_i y_{ji}^{kd}, \forall j \in C \tag{8}$$

$$p_{ij}^{k} \leq (Q - q_i - \sum_{\theta\in C}\sum_{d\in DR} q_\theta y_{i\theta}^{kd}) x_{ij}^{k}, \forall i \in C, j \in V, k \in K \tag{9}$$

$$b_i - a_j + w_i + t_{ij}^{v} \leq M(1 - \sum_{k\in K} x_{ij}^{k}), \forall i \in C, j \in C \tag{10}$$

$$a_j - b_i - w_i - t_{ij}^{v} \leq M(1 - \sum_{k\in K} x_{ij}^{k}), \forall i \in C, j \in C \tag{11}$$

$$a_i - a_j + t_{ij}^{d} \leq M(1 - \sum_{k\in K}\sum_{d\in DR} y_{ij}^{kd}), \forall i, j \in C \tag{12}$$

$$w_i \geq s_i^{v} * \sum_{j\in V}\sum_{k\in K} x_{ji}^{k}, \forall i \in C \tag{13}$$

$$(b_j - b_i + s_j^{d} + t_{ij}^{d}) - w_i \leq M(1 - \sum_{k\in K}\sum_{d\in DR} y_{ij}^{kd}), \forall i \in C, j \in C \tag{14}$$

$$b_i \geq a_i, \forall i \in C \tag{15}$$

$$u_i \leq b_i \leq l_i, \forall i \in C \tag{16}$$

$$\sum_{i\in C}\sum_{j\in C} d_{ij} y_{ij}^{kd} \leq r^{d}, \forall k \in K, d \in DR \tag{17}$$

$$x_{ij}^{k} \in \{0,1\}, \forall i \in V, j \in V, k \in K \tag{18}$$

$$y_{ij}^{kd} \in \{0,1\}, \forall i \in C, j \in C, k \in K, d \in DR \tag{19}$$

$$p_{ij}^{k} \geq 0, \forall i \in V, j \in V, k \in K \tag{20}$$

$$a_i \geq 0, \forall i \in C \tag{21}$$

$$b_i \geq 0, \forall i \in C \tag{22}$$

$$w_i \geq 0, \forall i \in C. \tag{23}$$

This mathematical formulation of the VRPTWDR is an extension of the VRPTW for which a model is presented by Cordeau et al. (2000) to consider self-driving robots as assistants. To highlight the importance of operational time, we modify the objective to minimize the total time spent in all routes as indicated in objective (1). Constraints (2) ensure each customer must be visited only once either by a vehicle or a robot. Constraints (3) imply that some customers are restricted to be visited by a vehicle. Constraints (4) and (5) make sure that each robot can only be dispatched once at a customer parking site. Constraints (6) restrict the vehicle to be used at most once in a schedule. Constraints (7) ensure the vehicle flow at each node. Payload balance is described through constraints (8). As the payload in a vehicle decreases after visiting each customer, these constraints also eliminate the formation of subtours that do not contain the depot. Constraints (9) are used to restrict the total load a vehicle carries by its capacity. Time windows are imposed by constraints (10)-(16), which are obtained through a linearization of non-linear inequalities. In addition, constraints (10) and (11) eliminate the risk of generating subtours in a solution. Constraints (17) limit the maximum radius of each delivery robot. Finally, constraints (18)-(23) enforce the binary and non-negativity restrictions on decision variables.

4. Operational features of the VRPTWDR

This section discusses the main operational characteristics of the investigated routing problem along with their mathematical formulations.

4.1. The role of customers' locations

Compared to traditional delivery modes, the advantages of self-driving delivery robots are parallel service and possibly shorter service time, while the disadvantages are lower travelling speed and limited walking range. Delivery robots can be utilized as a last mile solution when there is a clear benefit between these advantages and disadvantages, and vice versa. **Fig. 3** demonstrates a set of delivery routes with and without adopting robots for two (*a* and *b*) and three (*c* and *d*) customers.

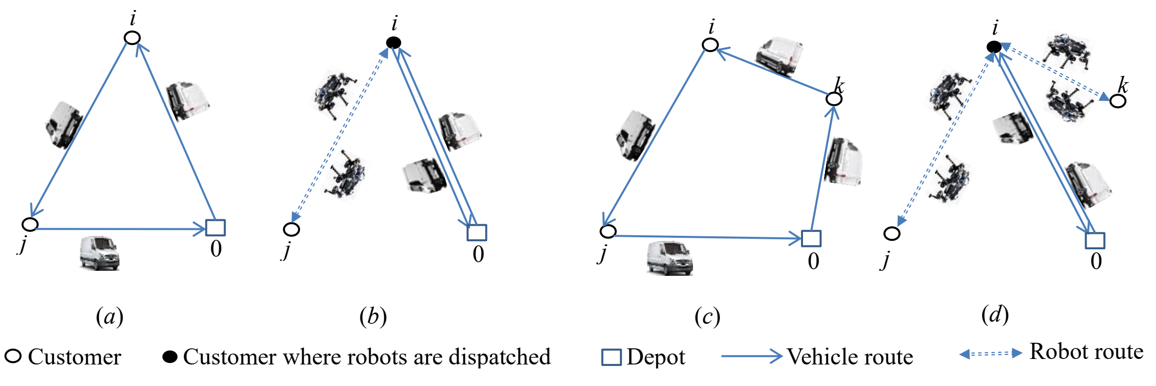

Fig. 3. Four representative routes of vehicle-visit *v*ersus robot-visit.

As shown in **Fig. 3**, if a vehicle needs to serve several customers in a single route, there are two alternative options: serving them sequentially (**Fig. 3*a*** and **Fig. 3*c***), or serving one of them and dispatching robots to serve the other(s) (**Fig. 3*b*** and **Fig. 3*d***). The difference of objective values between those two options lies on the path after the vehicle arrives at the first customer. In the case with two customers, let i represent the first customer and j represent the second.

Accumulated route times used after the vehicle arrives at customer i are $s_i^v + t_{ij}^v + s_j^v + t_{j0}^v$ and $\max\{s_i^v, 2t_{ij}^d + s_j^d\} + t_{i0}^v$ for **Fig. 3*a*** and **Fig. 3*b***, respectively. A potential time saving on route duration through using a robot to serve customer j is denoted by α_{ij}. This value is calculated with equation (24) below.

$$\alpha_{ij} = s_i^v + t_{ij}^v + s_j^v + t_{j0}^v - \max\{s_i^v, 2t_{ij}^d + s_j^d\} - t_{i0}^v. \tag{24}$$

Equation (24) can also be further rewritten into equation (25).

$$\alpha_{ij} = (s_j^v + t_{j0}^v - t_{i0}^v) + \begin{cases} t_{ij}^v, & \text{if } s_i^v - s_j^d \geq \dfrac{2d_{ij}}{vel_{ij}^d} \\ (s_i^v - s_j^d) + d_{ij}\left(\dfrac{1}{vel_{ij}^v} - \dfrac{2}{vel_{ij}^d}\right), & \text{if } s_i^v - s_j^d < \dfrac{2d_{ij}}{vel_{ij}^d} \end{cases}. \tag{25}$$

As shown, equation (25) contains two parts: the first one is related to traditional delivery service and the second part is related to the robot. If the superiority of the robot in terms of service time and speed is big enough to make $(s_i^v - s_j^d) \geq 2d_{ij} / vel_{ij}^d$ true, the value of α_{ij} is mainly determined by customers' locations based on which t_{ij}^v is calculated. In this case, a larger value of d_{ij} is preferred. Otherwise, a smaller d_{ij} is preferred as its coefficient is negative when $(s_i^v - s_j^d) < 2d_{ij} / vel_{ij}^d$, given $vel_{ij}^v \geq vel_{ij}^d$ is assumed in the VRPTWDR.

Furthermore, **Fig. 3*c*** and **Fig. 3*d*** illustrate a standard vehicle route and a case where two customers (j and k) are allocated to be visited by robots. Let α_{ijk} represent the benefit of dispatching two robots to visit customers j and k when the vehicle is visiting customer i. It can be calculated using equation (26).

$$\begin{aligned} \alpha_{ijk} &= s_i^v + s_j^v + s_k^v + t_{ij}^v + t_{jk}^v + t_{k0}^v - \max\{s_i^v, 2t_{ij}^d + s_j^d, 2t_{ik}^d + s_k^d\} - t_{i0}^v \\ &= s_i^v + s_j^v + t_{ij}^v + t_{j0}^v - \max\{s_i^v, 2t_{ij}^d + s_j^d\} - t_{i0}^v \\ &\quad + s_i^v + s_k^v + t_{ik}^v + t_{k0}^v - \max\{s_i^v, 2t_{ik}^d + s_k^d\} - t_{i0}^v \\ &\quad + t_{i0}^v - t_{j0}^v + t_{jk}^v - t_{ik}^v + \max\{s_i^v, \min\{2t_{ij}^d + s_j^d, 2t_{ik}^d + s_k^d\}\} - s_i^v \\ &= \alpha_{ij} + \alpha_{ik} + t_{i0}^v - t_{j0}^v + t_{jk}^v - t_{ik}^v + \max\{s_i^v, \min\{2t_{ij}^d + s_j^d, 2t_{ik}^d + s_k^d\}\} - s_i^v \end{aligned} \tag{26}$$

Without losing generality, $2t_{ij}^d + s_j^d \leq 2t_{ik}^d + s_k^d$ is assumed. The extra benefit of multi-dispatch (Δ_α) is the difference between α_{ijk} and the sum of α_{ij} and α_{ik}, which is expressed in equation (27).

$$\Delta_\alpha = \alpha_{ijk} - (\alpha_{ij} + \alpha_{ik}) = t_{i0}^v - t_{j0}^v + t_{jk}^v - t_{ik}^v + \max\{0, 2t_{ij}^d + s_j^d - s_i^v\}. \tag{27}$$

Since robots are used to visit nearby customers, it is reasonable to assume $t_{i0}^v - t_{j0}^v + t_{jk}^v - t_{ik}^v \approx 0$. Consequently, Δ_α can be estimated using equation (28).

$$\Delta_\alpha \approx \max\{0, 2t_{ij}^d + s_j^d - s_i^v\}. \tag{28}$$

Hence, the actual benefit of deploying two robot services is estimated to be no less than the sum of their individual values, i.e., $\alpha_{ijk}>\alpha_{ij}+\alpha_{ik}$. Similarly, the benefit of dispatching three or more robots to visit nearby customers has the same attribute.

4.2. The role of customers' time windows

When using delivery robots as assistants in the last mile delivery, customers' time window constraints may deteriorate the potential benefit. To analysis the influence of the time window, the relationships between two time windows are introduced first. **Fig. 4** shows the three kinds of relationships between $[l_1, u_1]$ and $[l_2, u_2]$. Without losing generality, $l_1 \leqslant l_2$ is assumed.

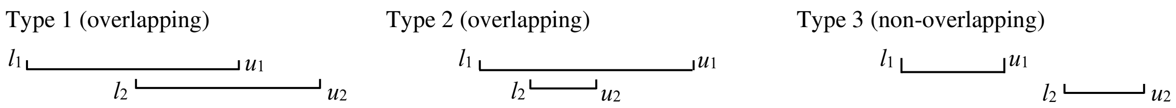

Fig. 4. Illustration of three types of relationships between two time windows.

When sending a robot to visit customer *j* at customer *i*'s parking site, there is a time interval, i.e., $[l_j - t_{ij}^d, u_j - t_{ij}^d]$, in which the robot should be dispatched to satisfy customer *j's* time window. Obviously, if $[l_j - t_{ij}^d, u_j - t_{ij}^d]$ and $[l_i, u_i]$ overlap and the vehicle arrives *i* during the overlapped time period, the time window constraints would not generate a deterioration on the objective value. Otherwise, there are two conditions in which time delay occurs: (1) $u_j - t_{ij}^d < l_i$, the vehicle needs to arrive *i* earlier than l_i, and (2) $l_j - t_{ij}^d > u_i$ and $l_j + t_{ij}^d + s_j^d > u_i + s_i^v$, the vehicle needs to spend extra time for the return of the robot. Let β_{ij} represent the deterioration resulting from the time window constraints when sending a robot to visit customer *j* from customer *i*, which is calculated with equation (29).

$$\beta_{ij} = \min\{ u_j - l_i - t_{ij}^d, u_i + s_i^v - l_j - t_{ij}^d - s_j^d, 0\} \tag{29}$$

Similarly, if customers *j* and *k* are both served by robots dispatched at customer *i*, the robot dispatching time windows are $[l_j - t_{ij}^d, u_j - t_{ij}^d]$ and $[l_k - t_{ik}^d, u_k - t_{ik}^d]$, respectively. If there is an overlap among them and customer *i*'s time window, no deterioration is induced. Otherwise, the deterioration (β_{ijk}) is calculated using three steps: (*i*) β_{ij} is calculated first using equation (29); (*ii*) a virtual customer *i'* is introduced to denote the coalition of customers *i* and *j*, and its time window and service time are calculated using equations (30)-(32); (*iii*) β_{ijk} equals to the value $\beta_{i'k}$ that can be calculated using equation (29).

$$s_{i'}^{v} = \max\{ s_i^{v}, s_j^{d} + 2t_{ij}^{d} \} - \beta_{ij} \tag{30}$$

$$l_{i'} = \begin{cases} \max\{ l_i, l_j - t_{ij}^{d} \}, & \text{if } l_i \le l_j \\ \min\{ l_i, u_j - t_{ij}^{d} \}, & \text{if } l_i > l_j \end{cases} \tag{31}$$

$$u_{i'} = \min\{ u_i, u_j - t_{ij}^{d} \}. \tag{32}$$

Similarly, the deterioration of deploying three and more robot sortie at a customer's parking site can be calculated.

The preliminary analysis in this section indicates the potential savings brought by the locations of the related customers and the deteriorations (extra waiting time) resulted from their time window constraints. In terms of potential savings, distance does not always negatively correlate with it. Moreover, simultaneously deploying multiple robots to visit nearby customers results a saving larger than the sum of that calculated one by one. This means that sometimes it is necessary to calculate the saving of dispatching multiple robots to serve other customers from one customer. In contrast, the deterioration is influenced by both time window constraints and geographical locations of customers. Finally, the formulations derived for calculating savings (α) and deteriorations (β) are useful in making the robot-delivery related decisions.

5. A matheuristic algorithm for the VRPTWDR

This section presents a matheuristic algorithm for the VRPTWDR. The VRPTWDR is NP-hard since it is an extension to the classical VRPTW. Only a simplified version of this problem can be solved to optimality with a reasonable computational time. For this reason, we have developed a two-stage matheuristic algorithm. In Stage 1, customers are clustered into several groups. In each group, only one customer is served by a driver and others are served by robots. These robots are dispatched from the customer's location visited by a driver. With the reduced number of decision variables, the problem is then solved in Stage 2. The following subsections detail the proposed algorithm.

5.1. Stage 1: Clustering

Due to the characteristics of the VRPTWDR, this stage has two consecutive clustering procedures. The first procedure is an IP approach which considers dispatching a robot between two customers: deploying a robot sortie from one customer to serve the other. The second one is a heuristic algorithm which considers dispatching serval robots at one customer' site to serve multiple customers simultaneously. These two stages are developed based on the benefits derived from allowing robot visits. The resulting benefit of allowing customer j to be visited by the robot dispatched at customer i, denoted as γ_{ij}, is calculated with equation (33). This value results from their geographic locations, time windows and the robot's radius limitation.

$$\gamma_{ij} = \begin{cases} \alpha_{ij} + \beta_{ij}, & \text{if } d_{ij} \le r^d \\ -\infty, & \text{if } d_{ij} > r^d \end{cases}. \tag{33}$$

The basic principle of clustering customers is to maximize total benefit from deploying robot services. Therefore, an integer programming (IP) approach is applied based on the benefit between every pair of customers. However, the IP approach does not consider a case when multiple (two or more) customers are visited by robots dispatched at the same customer's parking site, like α_{ijk} and β_{ijk}. Then, our heuristic-based clustering approach is implemented based on the results of the IP approach to further investigate conditions of dispatching multiple robots at a customer.

5.1.1. IP approach for clustering

Theoretically, every pair of customers with a positive benefit value can be allocated into a cluster. Here, a new decision variable λ_{ij} is introduced: equals to 1 if a customer *j* belongs to the cluster with customer *i* as the central node, and 0 otherwise. And we name each cluster after its central node. Further, let ζ_{ij} represent the potential deterioration on the objective when customers *i* and *j* are clustered into one group, which is caused by the difference of their time windows. We use equation (34) to quantify it.

$$\zeta_{ij} = \min\{\min(u_i, u_j) - \max(l_i, l_j), 0\}, \forall i \in C, j \in C. \tag{34}$$

To maximize overall benefit, an IP model ([P2]) is developed to obtain the initial cluster configuration.

[P2]

$$\text{maximize } = \sum_{i \in C} \sum_{j \in C} \gamma_{ij} \cdot \lambda_{ij} + \frac{1}{2} \sum_{\varphi \in C} \sum_{i \in C} \sum_{j \in C} \zeta_{ij} \cdot SC_{ij}^{\varphi} \tag{35}$$

subject to

$$\lambda_{ij} \le \lambda_{ii}, \forall i \in C, j \in C \tag{36}$$

$$\sum_{j \in C} \lambda_{ij} \le |DR| + 1, \forall i \in C \tag{37}$$

$$\sum_{i \in C} \lambda_{ij} = 1, \forall j \in C \tag{38}$$

$$\sum_{j \in C} q_j \cdot \lambda_{ij} \le Q, \forall i \in C \tag{39}$$

$$\lambda_{\varphi i} \le SC_{ij}^{\varphi} + M \cdot (1 - \lambda_{\varphi j}), \forall i \in C, j \in C, \varphi \in \Phi \tag{40}$$

$$SC_{ij}^{\varphi} \le \lambda_{\varphi j}, \forall i \in C, j \in C, \varphi \in \Phi \tag{41}$$

$$SC_{ij}^{\varphi} \le \lambda_{\varphi i}, \forall i \in C, j \in C, \varphi \in \Phi \tag{42}$$

$$\lambda_{ij} \in \{0,1\}, \forall i \in C, j \in C \tag{43}$$

$$SC_{ij}^{\varphi} \in \{0,1\}, \forall i \in C, j \in C, \varphi \in C. \tag{44}$$

where SC_{ij}^{φ} is introduced for the linearization of [P2]. SC_{ij}^{φ} equals 1 if both customers i and j are allocated to the cluster φ, and 0 otherwise. The objective (35) is to maximize the total benefit resulting from clustering to use the robot as an assistant. Constraints (36) make sure that each customer j can only be allocated to a cluster center node. Constraints (37) limit the maximum number of customers in a cluster according to the maximum number of robots can be installed to a van, and constraints (38) ensure that each customer can only be allocated to one cluster. Constraints (39) restrict the total demand of customers in a cluster to be within a full vehicle load capacity. Constraints (40)-(42) ensure the feasibility of decision variables. Finally, constraints (43) and (44) enforce the binary restrictions on decision variables.

5.1.2. Heuristic clustering approach

To achieve further benefit of parallel delivery, a novel cluster procedure is proposed. First, according to the initial clusters generated by [P2], dummy nodes of customer i are introduced and their time windows and service time are calculated with equations (30)-(32). Then, the benefits of allocating multiple customers to them are calculated accordingly. In each iteration, the cluster configuration with the biggest positive benefit is chosen and the related parameters are updated. This process is repeated until no further benefit can be achieved. The algorithm procedure is detailed in **Algorithm 1**.

For each customer i served as a cluster center, there is a limited number of nearby customers that can be allocated to it, which is denoted by cn_i in **Algorithm 1**. cn_i is first limited by the difference of the maximum number of robots carried by a van and the number of customers that are allocated to customer i in the IP approach. Then, it is limited by the delivery van's capacity for customers' demand. Given the maximum possible number of clustered customers, the maximum clustering benefit is calculated for each potential cluster center. And the biggest positive value is chosen.

Algorithm 1. Pseudo code of the heuristic clustering approach.

Input: γ_{ij}, λ_{ij}, q_i, Q, l_i, u_i, s_i^v, s_i^d

Output: λ_{ij}

flag:=1
while (flag) **do**
  for each customer i that can be served as a cluster center (dispatching site) **do**
    minimum_benefit$_i$=0 and $n=|DR|-\sum_{j\in C}\lambda_{ij}$
    while ($n \geqslant 2$) **do**
      Sort customers in descending order of γ_{ij} and set the first n as the set *temp*
      if $\sum_{j\in temp} q_j + \sum_{j\in C} q_j\lambda_{ij} + q_i \le Q$ **then**
        total_benefit$_i$:=total_benefit of allocating customers in *temp* to i
        if total_benefit$_i \geqslant$ minimum_benefit$_i$ **then**
          minimum_benefit$_i$= total_benefit$_i$; cn_i=n;
        end if
      end if
      n:=n-1
    end while
  end for
  if max(minimum_benefit)>0 **then**
    $i = \underset{\{j \backslash \text{in customers that can be served as a cluster center}\}}{\arg\max} (\min imun\_benefit_j)$
    allocate cn_i customers with biggest γ_{ij} to customer i (update $\boldsymbol{\lambda}_{ij}$)
    update l_i, u_i, s_i^v, s_i^d and γ_{ij}
  else flag:=0
  end if
end while

5.2. Stage 2: Solving Clustered-VRPTWDR

After forming customers into Φ clusters ($V_1 \ldots V_\varphi \ldots V_\Phi$) in stage 1 with V_0={0}, a set of clusters as denoted by $V=\{V_0, V_1 \ldots V_\Phi\}$ is generated. In stage 2, the following routing model [P3] is solved and the final solution to the VRPTWDR is derived.

[P3]

$$\text{minimize} \ = \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} x_{ij}^k \cdot t_{ij}^v + \sum_{i\in C}(b_i - a_i + w_i) \tag{1}$$

subject to

(2) ~(23)

$$\sum_{i\in V_\varphi}\sum_{j\in V\backslash V_\varphi}\sum_{k\in K} x_{ij}^k = 1, \forall \varphi \in \Phi \tag{45}$$

$$\sum_{i\in V_\varphi}\sum_{j\in V_\varphi}\sum_{k\in K} x_{ij}^k = 0, \forall \varphi \in \Phi \tag{46}$$

$$\sum_{i\in V_\varphi}\sum_{j\in V\backslash V_\varphi} x_{ij}^k = \sum_{i\in V_\varphi}\sum_{j\in V\backslash V_\varphi} x_{ji}^k, \forall \varphi \in \Phi, k\in K \tag{47}$$

$$\sum_{i\in V_\varphi}\sum_{j\in V\backslash V_\varphi}\sum_{k\in K}\sum_{d\in DR} y_{ij}^{kd} = 0, \forall \varphi \in \Phi \tag{48}$$

Constraints (45) and (46) indicate that only one customer in each cluster is visited by the vehicle. Constraints (47) ensure balance constraints for each cluster. Constraints (48) make sure that robot cannot travel between clusters. The model [P3] decides both customers that are visited by a vehicle in each cluster and a set of routes and times for vehicles to visit those customers. The proposed matheuristic algorithm reduces the complexity of the problem by decomposing it into two parts which are solved individually. Further, once the clustered-VRPTWDR can be solved to optimality, the result of clustering (stage 1) decides the quality of the result. However, as the clustered-VRPTWDR is still a NP-hard problem, the efficiency of its solution method dictates the limitation of the proposed algorithm.

6. Computational experiments

This section provides the detailed results of computational experiments. A series of numerical experiments were conducted to gain insights into potential strategies for the application of delivery robots and versify the proposed algorithm. The configuration of the computer used for those experiments is Inter Core i5-3610QM CPU @2.30GHz processor with 4GB RAM.

6.1. Test instances

We modified the well-known VRPTW instances created by Solomon (1987) to test our model and the algorithm. In these instances, there are three types of data sets: R, C and RC, correspongding to three types of customers' geographical distribution: uniform, clusted, and semi-clustered. And, these data sets have a further two subsets: sets R1, C1 and RC1 have a short scheduling horizon which allows only a few customers to be served by the same vehicle, while the sets R2, C2 and RC2 has a long scheduling horizon, permiting more customers to be serviced by the same vehicle. Each data set contains between eight and twelve instances. It is noted that all instances in sets R1 and R2 share same values on customers' locations, demands and service times, and the same in sets RC1 and RC2.

First, the features of the VRPTWDR are investigated. To ensure that the problem can be solved to optimality in reasonable computational effort using IBM ILOG CPLEX Studio 12.9.0 (IBM, 2019), we reduced the problem size due to the complexity of the VRPTWDR formulation. Based on Solomon instances, 16 customers are randomly picked from each instance in the Solomon instance set to form new set of instances. For each instance, 15 sub-instances are generated. Then, Solomon 25-customer and 50-customer instances are included in the experiments to show the performance of the proposed algorithm in Section 6.3.

As the motivation of delivery robot is assumed to be used as a last mile delivery solution in the context of city logistics (i.e., in built-up areas), it is reasonable to set average speeds of vehicle and robot to be 30 km/h and 10 km/h, respectively. Therefore, given the speed of vehicles is set as one unit in Solomon instances, we set the speed of robots as 0.3 units where no other explicit instructions. Furthermore, the maximum radius of robot's coverage is set to be equal to half of the average distance amongst customers.

Besides the time saving on parking, there is usually a distance for the driver (courier) from parking site to the door of a customer in urban last mile delivery. Therefore, the service time of robots is set to be half of that for a vehicle. Meanwhile, in these instances, a customer with a demand higher than twenty is recognized as infeasible to be served by a robot. Finally, the available number of robots in a single vehicle is assumed to be four for instances with 16 customers and six for instances with 25 and 50 customers.

6.2. Analysis of the VRPTWDR features

6.2.1. Customers' geographical distribution

The influence of customers' geographical distribution on the application of delivery robot assitants is investigated in this subsection. We compared the results on 16-customer instances which are randomly generated based on three Solomon instances, namely C101, RC101 and R101. First, the results of the VRPTW and VRPTWDR models are listed in **Table 1**. The columns Obj. report the objectives of the corresponding models. The columns NR report the number of the robot-visit deployments and columns Imp. report the improvement ($= \frac{\mathrm{OBJ}_{VRPTW} - OBJ_{VRPTWPD}}{\mathrm{OBJ}_{VRPTW}} \times 100\%$) obtained by the application of robots. As indicated in **Table 1**, introducing robots into last mile delivery improves the objective and it can even be more significant when customers are clustered. Correspondingly, higher improvement corresponds to a higher number of visits done by delivery robots.

Table 1

Comparing the results of the VRPTWDR with the VRPTW.

No. of instances	C101_16				RC101_16				R101_16			
	VRPTW	VRPTWDR		Imp.	VRPTW	VRPTWDR		Imp.	VRPTW	VRPTWDR		Imp.
	Obj.	Obj.	NR	(%)	Obj.	Obj.	NR	(%)	Obj.	Obj.	NR	(%)
01	1,835.3	1,494.4	5	18.57	595.5	523.2	2	12.14	714.4	700.7	3	1.91
02	1,742.5	1,398.7	5	19.73	541.6	467.8	2	13.63	768.8	704.1	2	8.42
03	1,758.9	1,532.7	5	12.86	506.2	500.8	1	1.07	675.0	641.1	4	5.02
04	1,680.1	1,376.4	6	18.08	529.7	450.9	4	14.88	755.1	725.9	5	3.87

05	1,707.0	1,428.6	6	16.31	554.6	481.5	6	13.18	672.0	615.7	8	8.38
06	1,724.3	1,468.0	6	14.86	501.8	488.3	4	2.69	698.7	672.7	2	3.73
07	1,907.5	1,643.5	5	13.84	514.2	514.2	0	0.00	744.1	697.6	3	6.25
08	1,691.4	1,316.4	6	22.17	507.7	503.4	2	0.85	755.8	755.8	0	0.00
09	1,706.8	1,384.9	6	18.86	567.3	495.0	3	12.74	753.7	715.8	2	5.03
10	1,813.6	1,306.8	6	27.94	486.6	482.9	2	0.77	671.3	661.5	3	1.46
11	1,964.4	1,750.6	6	10.88	530.3	530.3	0	0.00	641.0	573.4	3	10.56
12	1,707.0	1,314.1	6	23.02	481.8	474.1	5	1.60	733.1	706.3	4	3.65
13	1,770.7	1,593.9	5	9.99	517.6	515.7	2	0.38	755.7	738.6	3	2.26
14	1,696.9	1,412.5	6	16.76	555.8	478.8	4	13.84	736.3	736.3	0	0.00
15	1,815.5	1,652.3	4	8.99	562.9	492.2	4	12.56	741.8	741.8	0	0.00
Average			5.5	16.86			2.9	6.69			2.8	4.04

To further examine the importance of customers' location, we also studied instances without time windows, i.e., VRP and VRPDR. As showed in **Fig. 5**, when there is no time window constraint, the VRPTWDR improves the objective value for clustered customers, while the improvement is very limited or none in R and RC instances.

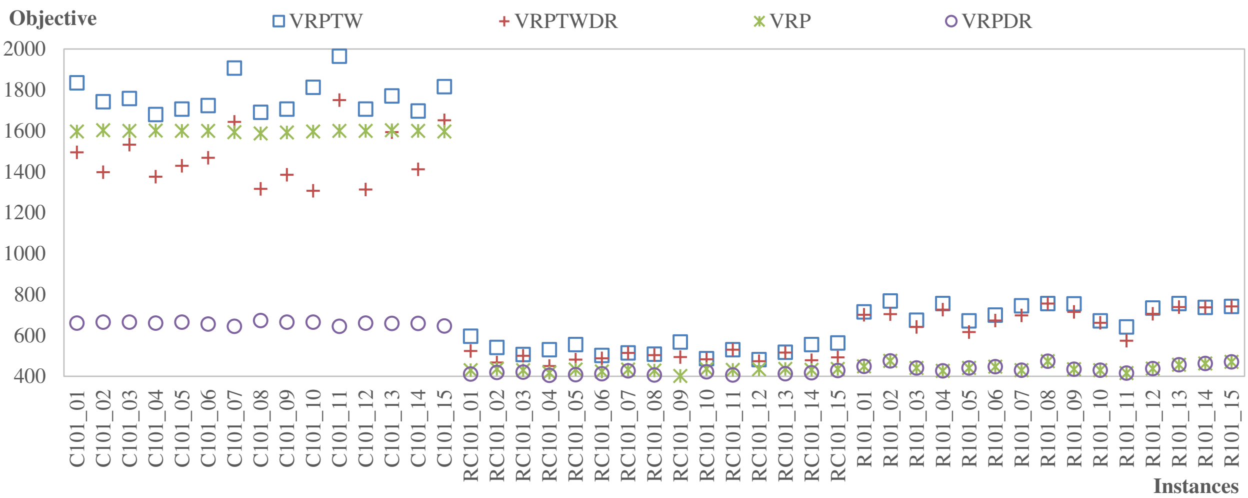

Fig. 5. Objective values of four problem for different types of customers' distribution.

6.2.2. Customers' time windows

To investigate the performance of the VRPTWDR with different types of time windows, we compared the results of 16-customer instances generated based on the Solomon instances with 100% time windows (i.e., C101, C105~C109, C201, C205~C209, R101, R105, R109~R112, R101, R205, R209~R211, RC101, RC105~RC108, RC201, and RC205~RC208). **Fig. 6** illustrates the relationship between the average width of customers' time windows (AWT) and average improvement in the objective compared to the VRPTW for each instance set.

As shown in **Fig. 6**, the benefit of using robots as assistants is strongly linked to the width of customers' time windows. First, delivery robots help to reduce total traveling time when time windows are tighter as the instances of type 1 show larger improvement than those of type 2: R1>R2, C1>C2, RC1>RC2. It is indicated that parallel deliveries conducted by robots can improve the delivery performance. The application of robots slightly reduces the number of vehicles as well. Second, the overall improvement of instances with different types are quite different. The improvement of C type instances is significant with an average value of 32.8%, while those of RC type are less notable, whose average value is 4.3%. Instances in set R also show slight improvement as no robot is deployed in any instance of R2 type. This is explained by the fact that with the limited speed the efficiency of the robot is not comparable with that of traditional vehicle systems, even with the form of swarm logistics. Third, for instances of C and RC types, the improvement increases as the AWT increases. This is because the differences among customers' time windows decrease the potential benefits.

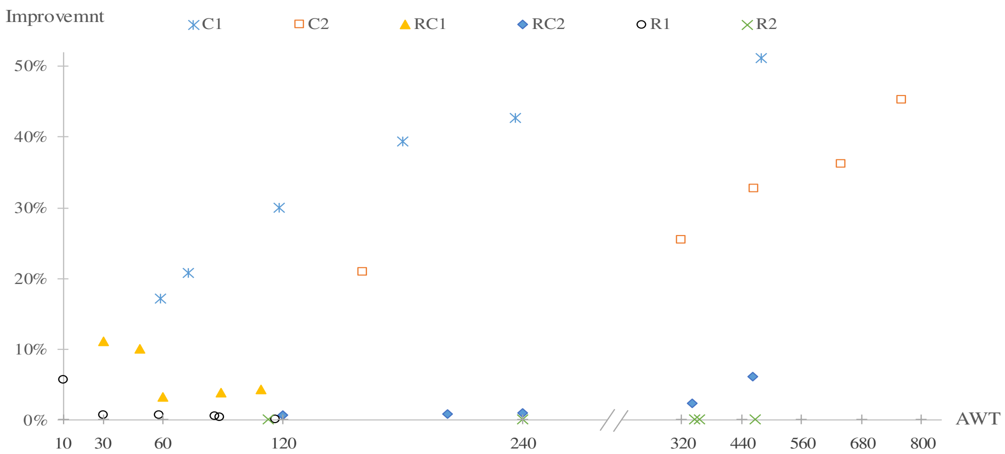

Fig. 6. Illustration of the relationship between the width of customers' time windows and improvement of $M_{VRPTWDR}$ on M_{VRPTW}.

The VRPTW requires more vehicles to handle tighter time windows imposed by customers, while the VRPTWDR can benefit through parallel deliveries implemented by robots. In fact, less vehicles are observed in the solution of VRPTWDR for those instances. Finally, for R type instances with the AWT above 120, there is no robot deployment at all in any of the solutions. The improvement is also equal to zero, which indicates that delivery robots provide no advantage when customers have wide time windows and are remotely distributed. Therefore, it can be concluded that using the delivery robot as an assistant in last mile delivery has significant superiority in serving customers with tight time windows.

6.2.3. *Speed and radius of the delivery robot*

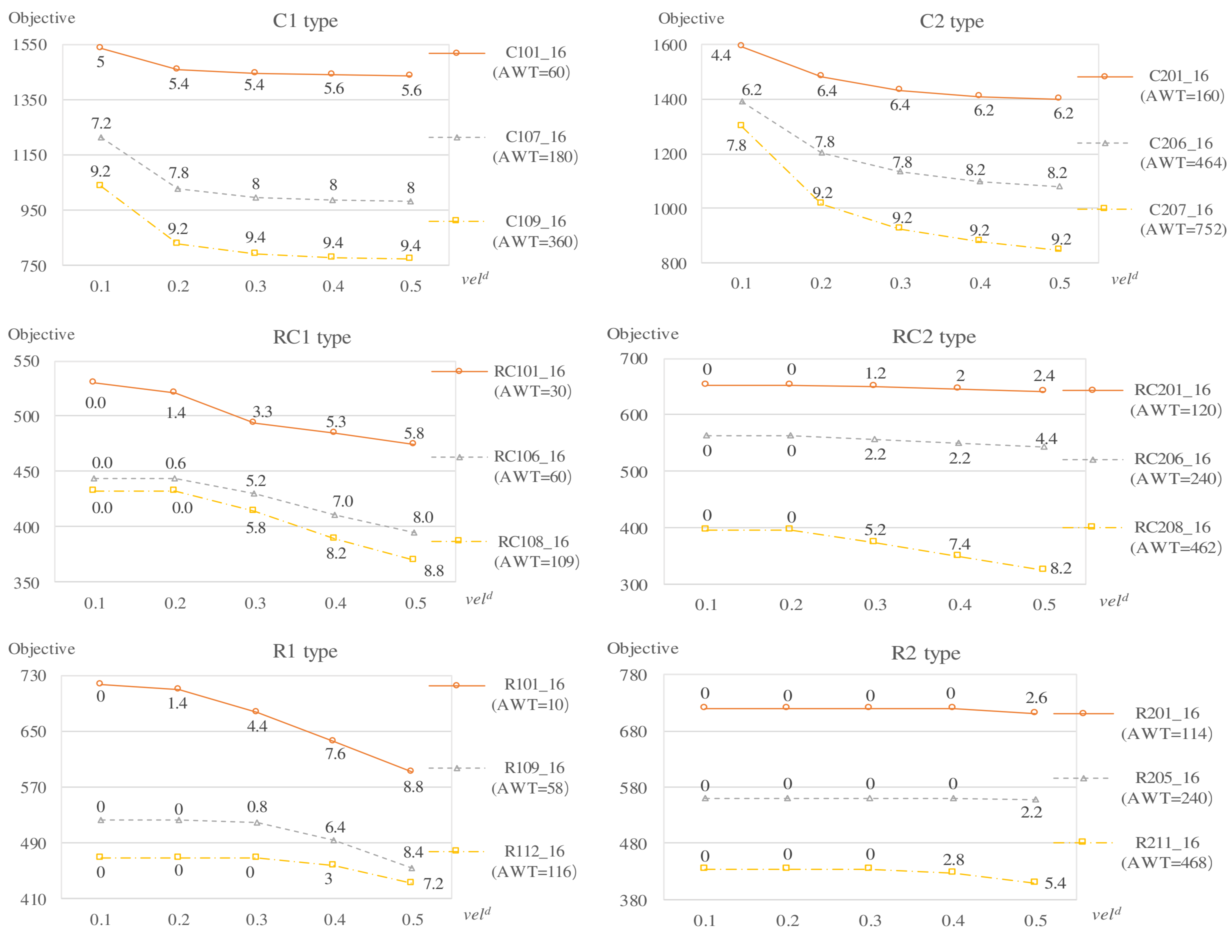

Fig. 7. Speed analysis of the delivery robot.

As the related technologies of delivery robot are continuously developed, we also investigate the influence of robot's speed on solutions. Keeping the speed of a vehicle as one unit, we varied the speed of the robot from 0.1 to 0.5 at an interval of 0.1. Based on the 16-customer instances set, for each type, we chose three scenarios to represent different values of AWT: Small, medium and large. The average objective value of 15 instances for each type are illustrated in **Fig. 7**. The values on the curves report the average number of customers that are served by delivery robots.

As expected, the speed of a robot is inversely proportional to the objective and directly proportional to the number of robots deployed. However, different types of instances show different trends. For instances in C types (C1 and C2), the curves appear to be concave, which indicates that the improvement of speed at lower intervals would decrease the objective greatly when customers are clustered. Even if the speed of delivery robots is at a low level, using them as assistants is beneficial to the efficiency of the last mile delivery. By contrast, the curves of instances in R and RC types are basically convex, which means only when the robot's speed is developed to a relatively high level, e.g., 0.4 and above in our experiments, there is improvement on objective values if customers are not highly clustered. It is verified again that robots need a sufficiently high speed before they show distinct advantages in last mile delivery practices.

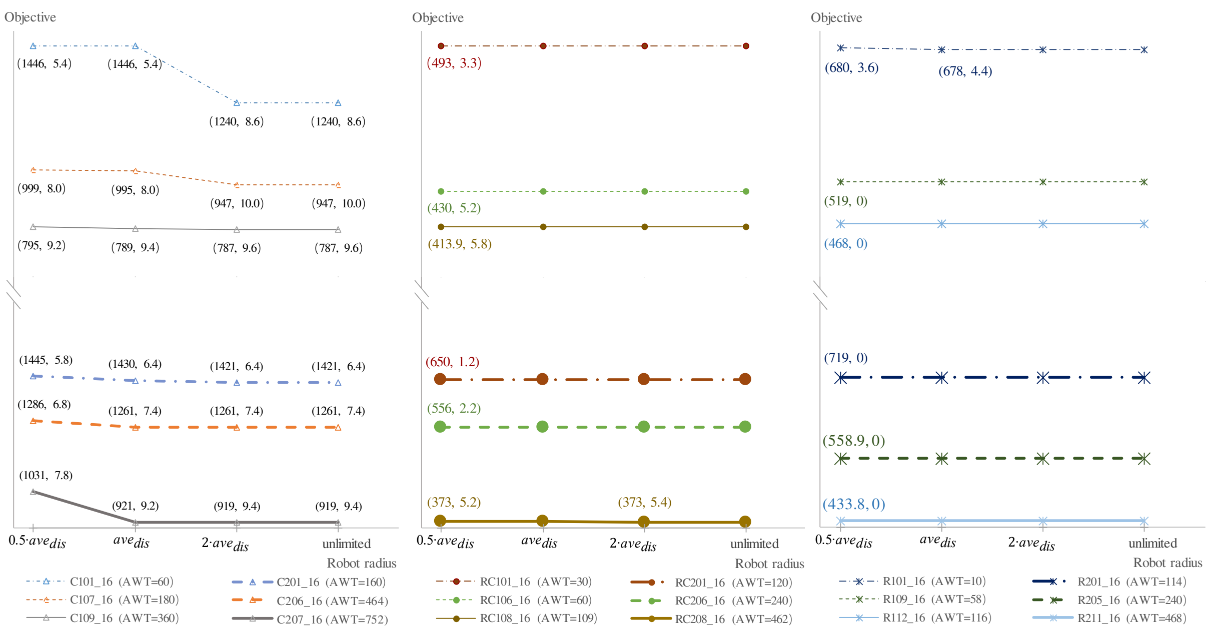

Fig. 8. Analysis on delivery robot's coverage radius.

It is noticed that the increase of the number of robots' sorties is not strictly proportional to the rise of the speed and the decrease of the objective values. This is because higher speed results in not only a higher possibility of the beneficial application of robot services but also less time spent by vehicles at dispatch sites. For example, in C type instances, under different robot traveling speeds, objective values vary while the number of customers visited by robots may be the same. On the other hand, there is a slight decline in the objective value with a large increase in the number of robots' services deployed, which occurs in instances of RC201 and RC206. This can be explained by the fact that dispatching more robots at a site simultaneously may achieve higher time savings.

We also conducted experiments to investigate robots' radius. The average results of 15 instances for each problem are reported in **Fig. 8**. Let ave_{dis} represent average distance among customers. The radius is set to be $0.5 \cdot ave_{dis}$, ave_{dis}, $2 \cdot ave_{dis}$ and unlimited. The data is labelled with the average objective value (the first value in brackets) and the average number of robot sorties (the second value in brackets). To make the figure concise, the label is not displayed if it is the same with its previous one (from left to right).

Longer radius value mean that the robot can serve more customers, which provides more choices for robot(s) deployments. Hence, a higher level of improvement on the objective and robot sorties are expected as the radius increases. As most curves in **Fig. 8** are strictly flat and the others are with very tiny slopes, the robot's radius plays a less important role than its speed. In term of customers' distributions, the instances of C type are more sensitive to the radius than those of RC and R types.

6.3. Analysis of the two-stage algorithm

In this section, we conducted experiments to evaluate the proposed two-stage algorithm. MILP and ILP models are solved via IBM CPLEX Studio 12.9.0, while the matheuristic algorithm is coded in MATLAB R2019b (MathWorks, 2019).

As most of R-type instances employ no robots at all, we focus on C-type and RC-type instances in our experiments. Error! Not a valid bookmark self-reference. provides the performance of the algorithm for instances with 16 customers.

Table 2

The two-stage algorithm's performance on the instances with 16 customers.

Instance	The two-stage algorithm			VRPTWDR formulation		Instance	The two-stage algorithm			VRPTWDR formulation	
	OBJ	Gap (%)	CPU (s)	OBJ	CPU (s)		OBJ	Gap (%)	CPU (s)	OBJ	CPU (s)
C101_16_01	1,494.4	0.00	3.4	1,494.4	448.7	RC101_16_01	523.2	0.00	2.9	523.2	15.1
C102_16_01	1,112.7	0.00	3.7	1,112.7	36.2	RC102_16_01	463.0	0.00	4.2	463.0	38.9
C103_16_01	1,106.5	0.00	2.9	1,106.5	2520.2	RC103_16_01	454.3	0.33	3.5	452.8	72.4
C104_16_01	818.4	0.65	2.6	813.1	27.5	RC104_16_01	434.1	0.35	3.8	432.6	330.3
C105_16_01	1,158.7	1.44	2.4	1,142.2	100.7	RC105_16_01	479.7	0.02	4.2	479.6	42.4
C106_16_01	1,383.0	0.00	2.5	1,383.0	102.6	RC106_16_01	451.0	0.00	3.9	451.0	93.1
C107_16_01	984.6	0.00	2.9	984.6	28.3	RC107_16_01	424.7	0.83	3.3	421.2	92.2
C108_16_01	999.3	0.03	3.7	993.0	90.2	RC108_16_01	421.2	1.20	4.7	416.2	277.8
C109_16_01	810.0	0.00	2.8	810.0	18.1	RC201_16_01	705.9	0.00	4.7	705.9	19.9
C201_16_01	1,315.5	0.00	1.8	1,315.5	99.4	RC202_16_01	587.7	0.27	3.8	586.1	61.3
C202_16_01	1,228.1	0.00	4.3	1,228.1	1608.2	RC203_16_01	539.7	0.00	3.9	539.7	2,076.0
C203_16_01	1,228.1	0.00	3.0	1,228.1	1610.9	RC204_16_01	480.4	1.14	4.1	475.0	7,492.3
C204_16_01	921.6	0.00	4.1	921.6	67.6	RC205_16_01	674.9	0.51	3.4	674.9	400.8
C205_16_01	1,364.7	1.28	3.2	1,347.4	32.3	RC206_16_01	514.2	0.00	3.2	514.2	181.5
C206_16_01	1,673.3	0.00	4.6	1,221.4	1673.3	RC207_16_01	503.1	0.14	2.3	502.4	131.8
C207_16_01	935.6	1.06	4.1	925.7	30.7	RC208_16_01	387.0	1.81	4.0	386.3	2,992.9
C208_16_01	1,075.9	1.52	4.4	1,059.8	109.0						
Average		0.35	3.3		506.1			0.41	3.8		894.9

The "Gap" reported is the percentage difference between the solutions obtained by CPLEX and the two-stage algorithm. The exact algorithm is highly parameter sensitive as the computational times for instances with same scale are quite different. By contrast, the computational time used by the two-stage algorithm is much shorter and more stable, although there is a slight gap in objective values. With wider time windows, C2 type instances generally require more running time than C1 type instances, and this is also true for RC1 and RC2 type instances.

Table 3

The two-stage algorithm's performance on Solomon_25 instances.

Instance	The two-stage algorithm			VRPTWDR formulation		Instance	The two-stage algorithm			VRPTWDR formulation	
	OBJ	Gap (%)	CPU (s)	OBJ	CPU (s)		OBJ	Gap (%)	CPU (s)	OBJ	CPU (s)
C101_25	1,982.5	1.03	30.7	1,962.2	1,800*	RC101_25	599.6	1.58	3.8	590.2	884.5
C102_25	1,350.1	-1.43	13.6	1,369.6	1,800*	RC102_25	591.8	6.77	4.0	554.3	1,800*
C103_25	982.9	-5.40	6.4	1,039.0	1,800*	RC103_25	531.8	3.03	5.0	516.2	1,800*
C104_25	966.3	0.06	6.7	965.7	1,800*	RC104_25	541.9	5.86	4.5	511.8	1,800*
C105_25	1,519.2	6.57	49.9	1,425.6	1,800*	RC105_25	573.9	1.19	0.1	567.1	1,800*
C106_25	1,934.1	2.35	17.6	1,889.6	1,800*	RC106_25	557.2	6.90	2.0	521.2	1,800*
C107_25	1,200.8	6.36	12.5	1,129.0	1,800*	RC107_25	489.9	1.40	4.2	483.1	1,800*
C108_25	1,024.4	1.45	21.8	1,009.8	1,800*	RC108_25	482.4	0.34	4.7	480.9	1,800*
C109_25	820.3	0.15	5.7	819.1	381.5	RC201_25	942.6	-3.08	36.7	972.5	1,800*
C201_25	1,619.4	-12.58	133.7	1,852.5	1,800*	RC202_25	843.4	3.31	86.5	816.4	1,800*
C202_25	1,368.9	-14.21	11.2	1,595.8	1,800*	RC203_25	702.8	-2.25	4.4	719.0	1,800*
C203_25	1,212.4	-4.09	6.2	1,264.0	1,800*	RC204_25	636.4	-17.06	5.0	767.3	1,800*
C204_25	1,094.7	-0.79	28.6	1,103.5	1,800*	RC205_25	809.4	-8.11	3.0	880.8	1,800*
C205_25	1,302.2	-4.68	17.1	1,366.0	1,800*	RC206_25	730.1	-5.13	285.7	769.6	1,800*
C206_25	1,239.1	0.32	92.8	1,235.2	1,800*	RC207_25	618.2	-5.42	46.8	653.6	1,800*
C207_25	1,248.6	2.04	65.8	1,223.6	1,800*	RC208_25	450.0	-1.74	31.2	458.0	1,800*
C208_25	1,211.2	1.97	237.4	1,187.8	1,800*						
Average		-1.23	44.6		-			-0.78	33.0		-

*: best feasible solution within a time limit of 30 minutes.

We also compared the performance of the two-stage algorithm with that of the exact algorithm on the Solomon instances with 25 customers. Because most instances with 25 customers cannot be solved to optimality within reasonable times, the computational time limit is set to be 30 minutes (1,800 seconds) for the VRPTWDR formulation. We provide the best feasible solution found within the time limit. As showed in **Table 3**, only two out of the total thirty-three instances can be solved to optimality within 30 minutes and there may be negative gaps when the solution of exact algorithm is not global optimal. It is noteworthy that the number of negative Gap increases as the instance size grows, especially in instances with loose time windows, like C2 and RC2 types.

To investigate the limitation on problem size of the proposed two-stage algorithm, the numerical experiments on Solomon_50 instances are also conducted. In the C1 and C2 types, there are nine customers which are not allowed to be visited by a robot, while in the RC type, this number is 13. **Table 4** lists these results within a maximum CPU time of 30 minutes. The results of the VRPTWDR formulation are not presented because feasible solutions cannot be found in 30 minutes CPU time limit. The columns named *No. of clusters* provide the number of clusters obtained at stage 1.

Table 4

Results of Solomon_50 instances using the two-stage algorithm.

Instance	OBJ	No. of clusters	CPU(s)	Instance	OBJ	No. of clusters	CPU(s)
C101_50	3,993.6	28	1,800*	RC101_50	1,513.9	28	1,800*
C102_50	2,988.9	21	1,800*	RC102_50	1,251.5	27	891.4
C103_50	2,205.6	17	738.5	RC103_50	1,062.6	21	31.7
C104_50	1,458.9	12	93.9	RC104_50	1,008.8	15	15.6
C105_50	3,009.6	22	1,800*	RC105_50	1,309.8	23	19.3
C106_50	3,520.3	25	1,800*	RC106_50	1,346.1	13	11.7
C107_50	2,317.3	18	1,800*	RC107_50	1,055.5	14	11.9
C108_50	1,913.7	15	1,800*	RC108_50	856.2	13	11.5
C109_50	1,511.0	12	182.1	RC201_50	2,360.0	43	1,800*
C201_50	3,579.8	21	1,800*	RC202_50	1,621.4	31	1,800*
C202_50	2,776.6	18	1,800*	RC203_50	1,412.0	24	1,800*
C203_50	2,198.0	15	1,800*	RC204_50	990.3	17	1,800*
C204_50	1,556.1	12	726.1	RC205_50	1,591.4	28	1,800*
C205_50	2,285.7	16	1,800*	RC206_50	1,455.4	27	1,800*
C206_50	1,911.4	14	1,800*	RC207_50	1,150.7	18	1,669.5
C207_50	1,825.3	14	1,800*	RC208_50	792.5	13	216.2
C208_50	1,815.0	14	1,800*				

*: best feasible solution found within a time limit of 30 minutes.

The required CPU time for clustering (stage 1) is very small and stable for instances with same scale (two and six seconds for 25-customer and 50-customer instances, respectively). In terms of solving the clustered VRPTW, the required CPU time increases with the number of clusters. Further, as instances of C2 and RC2 types have wider time spans and loose time windows than those of C1 and RC1 types, fewer instances can be solved to optimality. Besides, the number of robot-inaccessible customers also reduces the solution space. For example, seven out of eight instances in the RC1 type are solved to optimality in 30 minutes CPU time limit.

7. Conclusions

The demand for contactless delivery has expanded dramatically in 2020. Several technology companies are testing their autonomous robots to do last mile operations. While a parcel delivery by drones still has a few challenges to overcome, the other alternative of contactless delivery technology is becoming popular. Fortunately, several delivery robot developers have already reached the stage of trials by a number of real-world applications.

As a new last mile logistics solution, we have investigated an integrated vehicle delivery robot system to be used in the context of city logistics. In order to realize this promising last mile solution, it is important to know how much benefit it can achieve, and how to schedule vehicles and robots synchronously to achieve an efficient vehicle routing plan. We have formulated the investigated problem as a MILP model and conducted extensive numerical experiments to unveil the essential characteristics of using self-driving robots as delivery assistants.

Based on the operational factors studied in this paper, a simple but effective matheuristic algorithm is proposed for medium-sized instances. We have presented results of extensive computational experiments of the proposed algorithm and compared them against the solutions produced using the MILP formulation to evaluate its effectiveness and efficiency. The results highlight that the proposed algorithm is highly effective in finding good-quality solutions on instances with up to 50 customers. Because the two-stage algorithm employs an "IP" approach for solving the clustered VRPTW, its capability is also limited. However, the incorporation of effective clustered VRPTW heuristics or even VRPTW heuristics can enhance its performance on large-sized instances. As the purpose of this paper is not to determine the definitive heuristic, exploring and identifying alternative VRPTW metaheuristics is left as a future research topic. As a foundation work, this paper considers the minimization of total route times as a single objective. Future work may investigate other aspects of delivery robot application, such as other operational costs and energy consumption of delivery robots.

Acknowledgements

We thank the anonymous referees for their useful and constructive comments and suggestions.

References

Akeb, H., Moncef, B., Durand, B., 2018. Building a collaborative solution in dense urban city settings to enhance parcel delivery: An effective crowd model in Paris. *Transportation Resesrch Part E: Logistics and Transportation Review*, 119, 223-233.

BBCNEWS, 2019. Google's Wing delivery drones head to Europe. https://www.bbc.co.uk/news/technology - 46456694.

Bouton, S., Hannon, E., Haydamous, L., Heid, B., Knupfer, S., Naucler, T., Neuhaus, F., Nijseen, J.T., Ramanathan, S., 2017. An integrated perspective on the future of mobility, Part 2: transforming urban delivery. (McKinsey & Company). http://www.mckinsey.com/business-functions/ sustainability-and-resource-productivity/our-insights/urban-commercial-transport-and-the-future-of-mobility?cid=eml-web.

Boysen, N., Schwerdfeger, S., Weidinger, F., 2018. Scheduling last mile deliveries with truck-based autonomous robots. *European Journal of Operational Research*, 271(3), 1085-1099.

Braekers, K., Ramaekers, K., Nieuwenhuyse, I., 2016. The Vehicle Routing Problem: State of the Art Classification and Review. *Computers & Industrial Engineering*, 99, 300-313.

Chang, Y.S., Lee, H.J., 2018. Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. *Expert Systems with Applications*, 104, 307-317.

Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F., 2001. The VRP with time windows. P. Toth, D. Vigo, eds. The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, PA, 157–194.

DHL, 2016. Logistic Trends Radar, in: Markus, k. (Ed.). DHL Customer Solution & Innovation, Germany.

El Hachemi, N., Gendreau, M., Rousseau, L.-M., 2013. A heuristic to solve the synchronized log-truck scheduling problem. *Computers & Operations Research*, 40(3), 666-673.

Ghiami, Y., Demir, E., Van Woensel, T., Christiansen, M., & Laporte, G. (2019). A deteriorating inventory routing problem for an inland liquefied natural gas distribution network. Transportation Research Part B: Methodological, 126, 45–67.

Golden, B., Raghavan, S., Wasil, E., 2008. (eds)The vehicle routing problem: Latest advances and new challenges. Springer, New York.

Grolms, M., 2019. Autonomous Shuttles and Delivery Robots. https://www.advancedsciencenews.com/ autonomous-shuttles-and- delivery-robots/.

Hutter, M., Gehring, C., Lauber, A., Gunther, F., Bellicoso, C.D., Tsounis, V., Fankhauser, P., Diethelm, R., Bachmann, S., Bloesch, M., Kolvenbach, H., Bjelonic, M., Isler, L., Meyer, K., 2017. ANYmal - toward legged robots for harsh environments. *Advanced Robotics*, 31(17), 918-931.

IBM ILOG. 2019. Copyright ©International Business Machines Corporation, 1987.

Jeong, H.Y., Song, B.D., Lee, S., 2019. Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones. *International Journal of Production Economics*, 214, 220-233.

Karak, A., Abdelghany, K., 2019. The hybrid vehicle-drone routing problem for pick-up and delivery services. *Transportation Research Part C: Emerging Technologies,* 102, 427-449.

Keskin, M., & Çatay, B. (2018). Computers and Operations Research A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. Computers and Operations Research, 100, 172–188.

Kottasova, I., 2016. Forget drones, here come delivery robots. https://money.cnn.com/2015/11/03/technology/ starship-delivery-robots/?iid=EL >

Lin, C.K.Y., 2011. A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources. *Computers & Operations Research*, 38(11), 1596-1609.

Matlab R2019b. Copyright ©MathWorks, 1984.

McFarland, M., 2016. This Mercedes-Benz van will carry a fleet of delivery robots. http://money.cnn.com/2016/ 09/07/technology/starship-robot-mercedes-benz/index.html.

McFarland, M., 2019. UPS broke in to drone deliveries shuttling medical samples.https://www.

advancedsciencenews. com/autonomous-shuttles-and-delivery-robots/.

Melo, S.P., Baptista, 2017. Evaluating the impacts of using cargo cycles on urban logistics: integrating traffic, environmental and operational boundaries. *European Transport Research Review*, 9(2), 1-10.

Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies,* 54, 86-109.

Nguyễn, T.B.T., Bektaş, T., Cherrett, T.J., McLeod, F.N., Allen, J., Bates, O., Piotrowska, M., Piecyk, M., Friday, A., Wise, S., 2018. Optimising parcel deliveries in London using dual-mode routing. *Journal of the Operational Research Society*, 70(6), 998-1010.

Perboli, G., Tadei, R., Vigo, D., 2011. The Two-Echelon Capacitated Vehicle Routing Problem: Models and Math-Based Heuristics. *Transportation Science*, 45(3), 364-380.

Poikonen, S., Golden, B., 2020. The mothership and drone routing problem. *INFORMS Journal on Computing*, 32(2), 249-262.

Rothenbächer, A.K., Drexl, M., Irnich, S., 2018. Branch-and-price-and-cut for the truck-and-trailer routing problem with time windows. *Transportation Science*, 52(5), 1174-1190.

Sacramento, D., Pisinger, D., Ropke, S., 2019. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies,* 102, 289-315.

Schermer, D., Moeini, M., Wendt, O., 2019. A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Computers & Operations Research*, 109, 134-158.

Shields, N., 2018. China's Largest Courier is Starting Drone Deliveries. http://www.businessinsider.com/chinas–largest - courier-to-start-drone-deliveries-2018-4.

Solomon, M., 1987. Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraint. *Operations Research*, 35, 254-265.

Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A.L., Velasco, N., 2013. A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, 230(2), 231-244.

Vincent, J., Gartenberg, C., 2019. Here's Amazon's new transforming Prime Air delivery drone. https://www.theverge.com/2019/6/5/18654044/amazon-prime-air-delivery-drone-new-design-safety-transforming-flight-video.

Wang, Z., Sheu, J.-B., 2019. Vehicle routing problem with drones. *Transportation Research Part B: Methodological*, 122, 350-364.