Vehicle routing problems with drones equipped with multi-package payload compartments

Vehicle Routing Problem with Drones Equipped with Multi-Package Payload Compartment

Dr. M. Amine Masmoudi
Rabat Business School, International University of Rabat, Marocco.
Email : mohamed-amine.masmoudi@uir.ac.ma

Dr. Simona Mancini
University of Klagenfurt, Department of Production and Logistics Management, Austria
Email: simona.mancini@aau.at
University of Eastern Piedmont, Department of Innovation, Science and Technology, Alessandria, Italy.
Email: simona.mancini@uniupo.it

Prof. Roberto Baldacci
Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi", University of Bologna, Cesena, Italy
Email: r.baldacci@unibo.it

Dr. Yong-Hong Kuo*
Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong, China
Email: yhkuo@hku.hk

* corresponding author

Vehicle Routing Problem with Drones Equipped with Multi-Package Payload Compartment

Abstract

The Vehicle Routing Problem with Drones (VRP-D) consists of designing combined trucks-drones' routes and schedules to serve a set of customers with specific requests and time constraints. In this paper, we extend the VRP-D by including a fleet of drones equipped with multi-package payload compartments to serve more customers on a single trip. Moreover, a drone can return to a different truck from the departure one to swap its depleted battery and/or to pick up more packages. The problem aims to maximize the total revenue. We denote this problem as VRP-D equipped with Multi-package payload Compartment (VRP-D-MC). We propose an Adaptive Multi-Start Simulated Annealing (AMS-SA) metaheuristic algorithm to efficiently solve it. Experimental results show that our algorithm outperforms the current state-of-the-art algorithms for the Vehicle Routing Problem with Drones (VRP-D) in terms of solution quality. Extensive analyses have been conducted to provide managerial insights. The analyses carried out show (i) the benefits of using drones equipped with different compartment configurations, (ii) the increment of the total revenue obtainable using a combined trucks-drones fleet respect to a fleet of trucks in terms of total revenue, (iii) the savings for allowing a return to a different truck, (iv) the benefit of swapping drone battery and pick up the items at the same time. Moreover, a sensitivity analysis is conducted to assess the impact of time window constraints on the solutions. We also show that our different intensification and diversification mechanisms improve the convergence of the traditional SA.

Keywords: Vehicle Routing Problem with Drones, simulated annealing, multi payload compartments, multi start approach

1. Introduction

Unmanned aerial vehicles (UAVs), commonly known as drones, have been considered for numerous applications, including defense, territory monitoring, transportation and logistics, disaster relief, and farming (Chung et al., 2020), to name a few. In particular, the growing e-commerce market and the increasing demand for efficient last-mile logistics, along with the maturity of drones' technologies, have promoted considerable efforts to conceptualize the use of drones to provide package delivery services. For example, Wing, a division of Alphabet, was launched in 2012 to offer small package delivery services using small drones. The company was among the first to receive Air Carrier certification from the Federal Aviation Administration (FAA) and has recently announced conducting more than 100,000 flights across three continents. Similarly, in 2013, Amazon launched the project "Prime Air," where the

goal is to use drones to deliver light packages to customers within 30 minutes from the purchasing (Amazon, 2016). The company received the FAA approval in 2020 and tested many different vehicles designs and delivery mechanisms to find out the best configurations for delivery in various operating environments (FFA, 2020).

A widely conceptualized delivery system integrates trucks and drones (e.g., the mothership system) to provide package delivery services. As FAA requires drones to stay within the pilot's visual line of sight, trucks carry the packages and the drones to locations close to the customers' addresses. Drones are then loaded with packages and dispatched to these addresses. A drone can make either a single delivery or multiple deliveries per dispatch. In both cases, while drones fly to deliver the package(s), the truck may also be traveling to fulfill some of the delivery tasks (e.g., delivering heavy packages). As a result, the truck may collect its drones at a location different from where they have been dispatched.

This integrated truck-drone delivery problem is considered as a significant extension of the traditional Vehicle Routing Problem (VRP) (Murray and Chu, 2015), which is commonly known in the literature as the Vehicle Routing Problem with Drone (VRP-D) (Wang et al., 2017). The problem consists into determining the optimal routes for the trucks and the drones to complete a list of required delivery services while minimizing the system's total operational cost (Poikonen et al., 2017). Consequently, extensive research effort has been devoted to formulating and efficiently solving the VRP-D problem and to capture many of its real-world aspects and operational constraints. Examples of these aspects/constraints include satisfying the customers' time windows, the drones' load carrying capacity and flying range, the length of the working shift for drivers, the truck-drone routing interdependence, to name a few.

While a considerable portion of this research is devoted to study how drones serve multiple customers per dispatch, existing studies ignored some very important practical issues of the problem. For example, one crucial aspect is related to the drones' payload compartment configuration to support multiple deliveries per dispatch (Liu et al., 2020, Poikonen and Golden, 2020). Designing the drones' compartment involves: (a) maximizing the number of packages that a drone can carry considering the heterogeneity in their sizes and the drone's maximum load-carrying capacity, and (b) eliminating the need for direct interaction with the end customers. A commonly suggested design is to equip the drone with multiple compartments of different sizes. Each compartment is filled with one package and opens automatically as the drone arrives at the package delivery address. As the drone compartment configuration significantly affects the operation efficiency, it should be explicitly integrated into modeling frameworks studying the truck-drone delivery systems.

Another important aspect of this problem is the procedure to reload the drones and swap their depleted batteries. In many papers a simple policy is adopted, according to which, drones are obliged return to the same truck from which there were initially dispatched (e.g., Wang et al., 2017; Sacramento et al., 2019; Kitjacharoenchai et al., 2020; Liu et al., 2020). A more flexible tactic allows the drones to return to any other truck in their vicinities. While the latter tactic provides more flexibility in the route

planning, it involves complex synchronization operations among trucks and the drones (Macrina et al., 2020).

A typical assumption in the existing VRP-D studies is that a drone can only make one delivery per trip. Our work aims to address a new drone delivery feature where drones are equipped with a multi-package payload compartment. This new design implies that a drone can make multiple deliveries on the same trip. Another innovative aspect of our work is that a drone is allowed to meet another truck (different from the one it is launched) at a rendezvous location where the drones can not only swap the depleted battery, but also pick up additional packages (depending on the available capacity) and then continue their travel to visit more customer locations. Indeed, recent technological advancements have made the first assumption less relevant because newer drone models capable of carrying multiple customers' packages simultaneously have been proposed and developed. The Vulcan UAV Airlift is equipped with a multi-compartment payload (with two compartments) and it can lift and move a load of about 30 kg (i.e., about 15kg per compartment). Motivated by this new operational mode of drones, we consider in this research the new problem features, which allow drones to carry more than one item, return any truck to recharge/swap the battery, and pick up other packages from the truck. Other examples of drones can be found in https://www.unmanned systems technology.com.

Figure 1: Vulcan UAV Airlift (see https://filmora.wondershare.com/drones/top-heavy-lift-drones.html)

Finally, to deal with a problem as realistic as possible, an energy consumption function is also considered. This function takes into account the traditional elements like drone speed, mass, rotor, etc. (e.g., D'Andrea, 2014; Figliozzi, 2017; Stolaroff et al., 2018; Kirchstein, 2020). Despite this, we have considered some characteristics such as the energy consumption being depended on the current payload of the package by considering the resources during the fly.

The resulting problem is denoted as the Vehicle Routing Problem with Drone equipped with multi-packages payload Compartments (VRP-D-MC). To the best of our knowledge, this problem has not been addressed in the literature.

The main contributions of this paper are as follows:

i) First, we introduce the VRP-D-MC, which extends the traditional VRP-D of Wang et al. (2017) by considering the possibility for drones to carry more than one package at the same time, and to return

to a different vehicle respect to their departure one. We also consider the possibility to swap the drone's battery and to pick up further packages from the truck, during the same operational stop.

ii) We develop an Adaptative Multi-Start Simulated Annealing (AMS-SA) algorithm to solve the VRP-D-MC. Several diversification and intensifications procedures are introduced in our AMS-SA. Specifically, we add an exploration mechanism to avoid local optima using crossover operators, and an intensification mechanism using a local search procedure. A multi-start procedure to generate a new solution at each SA iteration is also developed. Also, we propose several special characteristics and algorithmic improvements to improve the AMS-SA.

iii) We perform extensive computational experiments on different benchmark sets.

a. On VRP-D instances from the literature, an improvement average gap of 0.61% to the best-know solutions for the VRP-D is obtained by our AMS-SA compared to 1.23% of the state-of-the art-approaches.
b. The results show the benefit of allowing drones to return to different vehicles along their routes. Moreover, we conduct some experiments to show the benefit of using multi-packages payload compartments compared to one compartment, as considered in the VRP-D addressed in the literature. Also, we show the impact of using drones in tandem with trucks as well as the impact of using time windows.

The rest of the paper is organized as follows. In Section 2, we review the literature. A formal description of the problem is given in Section 3. In Section 4, a mathematical formulation is provided. In Section 5, we present our algorithmic approach. In section 6, we provide the numerical results whereas managerial insights are reported in Section 7. Section 8 concludes our paper and indicates future research directions.

2. Literature review

In this section, we review the works related to the problem considered in this paper. For a comprehensive review of the literature on drone-aided routing, the reader is referred to Macrina et al. (2020) and Poikonen and Campbell (2021).

The VRP-D is a generalization of the Traveling Salesman Problem with Drone (TSP-D) (Wang et al., 2017) where a fleet of trucks work in tandem with a fixed number of drones to serve a set of customers (Salama and Srinivas, 2020). The VRP-D was first introduced by Wang et al. (2017) and then studied by Poikonen et al. (2017), Sacramento et al. (2019), Schermer et al. (2019), Wang and Sheu (2019), Kitjacharoenchai et al. (2020) and Tamke and Buscher (2021).

In the work of Wang et al. (2017), the drones are dispatched and recovered by the trucks either at the depot or at customers' locations. The objective is to minimize the completion time. The authors studied and analysed different worst-case scenarios depending on the number of drones in each truck and drones' and vehicles' speed. However, they neglected the limits on the drones flying range in their model. Schermer et al. (2018) extend the VRP-D of Wang et al. (2017) by considering limited driving

range of the drones to deliver one package. They developed two heuristics composed of the initialization and improvements phases. The results show that the proposed heuristics provide better results than the results obtained by Wang et al. (2017). Wang and Sheu (2019) study a variant of the VRP-D where multiple trucks and multiple drones of limited capacities are dispatched. Sacramento et al. (2019) developed an Adaptive Large Neighborhood Search (ALNS) to solve the VRP-D in order to minimize the total routing costs. A fleet of trucks is considered where each of them is equipped with only one drone. Pugliese and Guerriero (2017) extended the VRP-D by considering time windows to serve the customers and a maximum route duration imposed to the trucks. A similar setting is studied in Ham (2018). Chiang et al. (2019) develop a Genetic Algorithm (GA) to minimize the total delivery costs of the drones and vehicles and carbon emitted by conventional trucks. In their problem, each vehicle carries only one drone. Their experiments show that using drones in tandem with vehicles results in cost-effective solutions which are also environmentally friendly. Kitjacharoenchai et al. (2019) study a VRP-D with multiple drones and trucks. In their problem, a drone is deployed from a truck and can return to a different truck at another location. The authors propose a MILP formulation, whose objective is to minimize the makespan. Since the problem is difficult to solve, an insertion heuristic algorithm is developed to deliver solutions for large-scale problems, which consist of up to one hundred customer locations. In a later paper, Kitjacharoenchai et al. (2020) present a capacitated VRP-D with multiple trucks and drones. They also consider battery for drones, which limits their flying times. In their application, a drone is deployed by truck and has to return to the same truck but can perform multiple deliveries in the same trip. The completion time of the tasks is minimized in the objective function, subject to limited truck and drone capacities. Liu et al. (2020) study the effect of the variation in the payload on the energy consumed by the drones. In their work, they model the energy consumption with the factors such as payload, motor efficiency, distance travelled, and flying velocity. They develop a heuristic that incorporates both savings and nearest neighbourhood strategies. Their computational experiments on randomly generated problem instances of different sizes show the performance of their proposed method. Gonzalez-R et al. (2020) considered a fleet of truck where each of them is equipped with one drone. In addition, they considered the battery capacity drones and also multiple visits to customers in a single flight. An Iterated Greedy (IG) algorithm is used to solve this problem. The results show that the algorithm outperforms the solutions obtained by the commercial Gurobi solver in a shorter computational time. Euchi and Sadok (2021) proposed a Hybrid Genetic Algorithm (HGA) to solve the traditional VRP-D proposed by Sacramento et al. (2019). Poikonen and Golden (2020) considered a single truck and multiple drones where each drone is allowed to carry multiple items at a time, i.e., the drone serve several customers before returning to the truck to be recharged and to pickup other packages. The truck plays the role as a depot and also as a recharge station for the drones. In addition, the vehicle can move at a rendezvous point (customers' locations) in order to swap the battery. To solve this problem, the authors proposed a flexible heuristic named Route Tansform Shortest path (RTS). Similarity, Luo et al. (2021) studied the same problem of Poikonen and Golden (2020) by considering

that drones are limited by both energy consumption constraints based on flight-time and payload, as well as the maximum payload capacity constraints. To solve the problem, the authors proposed a Tabu Search (TS) algorithm which is tested on a new benchmark data set based on the Solomon's instances.

In this paper, we consider a variant of the VRP-D where each drone is also equipped with multi-compartments that can be used to deliver packages to several customers before returning to the truck. A drone can return to a vehicle different from its launching vehicle, and can also swap its depleted battery to service more packages. In addition to capacity constraints for the trucks and the drones, we also consider time windows constraints for the customers. Our problem therefore generalises the problem studied by Poikonen and Golden (2020).

Inspired from Macrina et al. (2020), Table 1 presents a summary of the main features of the most relevant works studied in the literature mentioned above.

Table 1
Summary of related works on the VRP-D

Reference	#T/#D	OF	TW	SY	ED	DR	DC	TC	DMV	DRA	MC	SD
Wang et al. (2017)	n/m	Min. completion time		√				√				
Pugliese and Guerriero (2017)	n/m	Min. completion time	√	√				√				
Ham (2018)	n/m	Min. makespen	√									
Wang and Sheu (2019)	n/m	Min. logistics costs		√			√	√	√	√		
Kitjacharoenchai et al. (2019)	1/m	Min. delivery time		√			√	√				
Sacramento et al. (2019)	n/m	Min. operational costs		√			√	√				
Schermer et al. (2019)	n/m	Min. makespen		√				√				
Chiang et al. (2019)	n/m	Min. total cost and CO_2 emission		√			√	√				
Kitjacharoenchai et al. (2020)	n/m	Min. completion time		√			√	√				√
Liu et al. (2020)	n/m	Min. operational costs		√			√	√				
Gonzalez-R et al. (2020)	n/m	Min.travel time		√			√	√	√			
Poikonen and Golden (2020)	1/m	Min. makespen		√	√	√	√		√			
Euchi and Sadok (2021)	n/m	Min. traveled time		√			√	√				
Luo et al. (2021)	1/m	Min. makespen		√	√	√	√		√			
Our	**n/m**	**Max.revenue**	**√**	**√**	**√**	**√**	**√**	**√**	**√**	**√**	**√**	**√**

#T/#D: Number of Trucks and Drones used. **OB**: Objective Function. **TW**: Time Windows. **SY**: Synchronization. **ED**: Energy Drone. **DR**: Drone Recharge. **DC**: Drone Capacity. **TC**: Capacity of the Truck; **DMV**: Drone Multiple Visit. **DRA**: Drone launch from truck and Return to Another truck. **MC**: Multi-Compartment payload. **SD**: the trucks and drones assumes the services simultaneously (both vehicles deliver the packages)

3. Problem definition

The VRP-D-MC consists of designing a set of trucks and drones' routes collaboratively to deliver packages to customers. The objective is to maximize the total revenue.

Due to the limited flying range of drones, trucks are used as a base to swap batteries of drones. Such a problem setting is studied in Wang et al. (2017) and Gonzalez-R et al. (2020). In addition to the features of the basic problem's settings, we consider the flexibility for drones to return to any truck (not necessarily the truck where the drone is dispatched), in contrast to the typical problem settings used in the literature (e.g., Sacramento et al., 2019; Kitjacharoenchai et al., 2020; Liu et al.,

2020). In addition, we consider that trucks can also deliver items which are supposed to be delivered by drones, if this is more convenient for the system. The latter has been observed in real-life delivery operations for heavy deliveries (Sacramento et al., 2019; Gonzalez-R et al., 2020), but not for small item delivery or reconnaissance missions. In short, in our study, the problem setting presents a more general case in which each customer can be served or visited either by a vehicle or by a drone.

We organize the different constraints into four categories including customers, trucks, drones and truck-drone. Below are reported the characteristics of our VRP-D-MC.

Customers:

- Each customer must be visited by either a truck or a drone.
- Similar to several drones routing problems studies (e.g., Pugliese and Guerriero, 2017; Pugliese et al., 2020; and Han, 2020), we consider time windows constraints where each customer must be visited within its time window. If a vehicle arrives at a customer before the beginning of its availability time window, the service is delayed to the earliest available time.

Trucks:

- Each truck route must start and end at the same depot.
- Truck capacity constraints must be respected.
- Each truck is equipped with one drone but there is sufficient space in a truck to host more than one drone simultaneously.

Drones:

- The total demand of the drone's route for the visited nodes must not exceed the capacity of the drone.
- A drone can visit multiple customer locations on a single trip, as long as the battery is sufficient for the drone to visit all these locations and to return to a vehicle at a rendezvous point (i.e., a customer location)
- Each time a drone returns to a truck to swap its battery, it can pick up more packages from the current truck.
- There is no restriction on the number of times that the batteries can be swapped. We assume that the swapping time is negligible.

Truck-drone:

- Drones can return to a truck only at a customer node (i.e., a rendezvous location). Batteries of drones could only be swapped at the rendezvous points. Also, routes of trucks and drones are synchronized at the rendezvous locations. That is, if the truck arrives at the customer location before the drone does, the truck must wait for the drone, and vice versa.
- A drone can wait at a customer location without energy consumption. In addition, as in Puglia Pugliese et al. (2020), we assume a maximum allowed waiting time for the drone at a rendezvous

location, defined by parameter *T*. Thus, a drone can wait at most *T* units of time for the truck, which it is supposed to land on. Instead, no limitations on the maximum waiting time are imposed for trucks.

- It is allowed that the drone is carried by a truck for part of its route. In this case, no energy is consumed by the drone's battery.
- Drones may be deployed by or return to the truck only at specific location point. The locations where a drone is launched and retrieved can be different.
- It is not necessary for a drone to return to the same truck where it was deployed for swapping batteries.
- The duration of each truck route must not exceed the maximum working time.
- Service times required by trucks and drones may be different because their delivery operations are different.

Figure 2 shows an example of the VRP-D-MC with two trucks (T1 and T2) and two drones (D1 and D2). As shown by the figure, the drones service multiple users before finally return to the trucks.

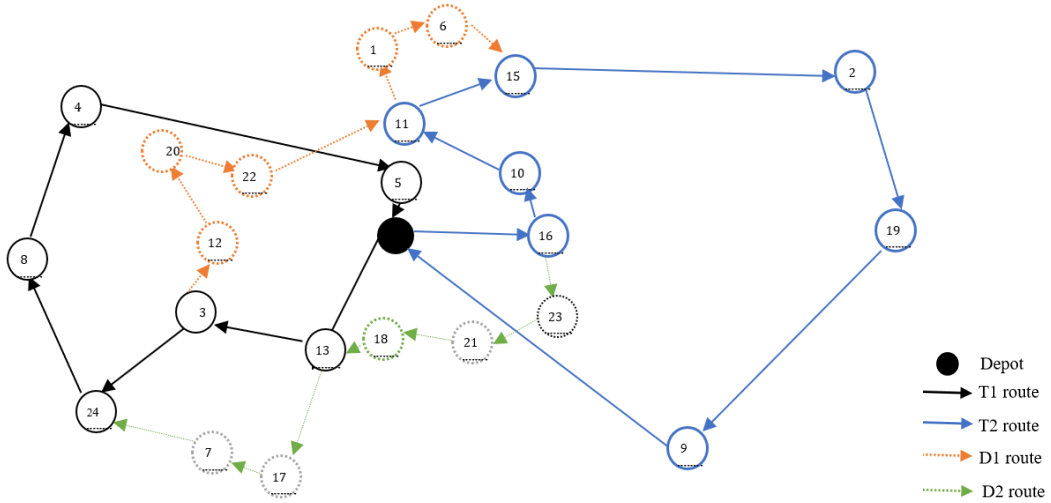


Figure 2. Illustrative VRP-D-MC solution

In this example, we can see for instance that the drone D1 is lunched from the truck T1 to serve the customers 12, 20 and 22 then meet another truck T2 at the rendezvous customer location 11 to swap their depleted battery and get the packages to serve the customers 1 and 6 before ending their travel at the truck T2 in the meeting point 15.

4. A mathematical formulation

In this section, we provide a formal mathematical model description and the energy consumption energy of the drone.

4.1. Mathematical model

The VRP-D-MC is defined as follows. Let $G = (V, A)$ be a complete directed graph, where V is the set of all nodes with $V = N \cup st \cup en$ and $A = \{(i,j): i,j \in V, i \neq j\}$ is the set of arcs. Let's $N = \{1, \dots, n\}$ the set of n customers while st and en represent the starting and ending depot, respectively. We denote by $\Delta^+ i(\Delta^- i)$ represent the set of nodes that can be reached(used) from node $i \in N$.

We consider a fleet composed of homogeneous trucks T and drones D to service the customers. The drones are assumed to pickup and delivery the items from the trucks to the customers distributed in the networks. Each truck and drone start from a depot and ends at the departure depot. Let V^T *and* V^D the speed of the truck and drone, respectively. Each arc $(i,j) \in A$, we associate a distance d_{ij}^T and a non-negative travel time t_{ij}^T between two locations i and j traveled by each truck T, where $t_{ij}^T = d_{ij}^T / V^T$. Moreover, a distance d_{ij}^D and a non-negative travel time t_{ij}^D between two locations i and j traveled by each drone D, where $t_{ij}^D = d_{ij}^D / V^D$. In addition, an operational cost c^T and c^D is associated for each truck and drone, respectively for each traveling arc.

We assume that the number of visits that the drones can recharge (swap) its battery in the trucks is unlimited. Q^T are the load capacity of each truck. $Q^{r,D}$ is the available number of resources of type $r \in R = \{1, 2, 3\}$ in the drone. More specifically, r takes the value 0 if the customer requires a place for a package of volume equal to 20 cm^3 place in the drone, r=1; where the customer requires a package of volume equal to 40 and r=3; the customer requires a package of volume equal to 40 cm^3 into the drone. The battery capacity of the drone is expressed by H, in which its energy is consumed at each traveling arc (i,j) at a EC energy rate. Each customer $i \in N$ is associated with a time window $[T_i^-, T_i^+]$, a demand q_i^r for each resource type r that can be served by a drone and a demand q_i can be served by a truck, and a service time s_i. We note that the customer request can be satisfied either by drone or by truck. The demand is measured by weight unit in case the delivery is assumed by truck and by both weight and volume in the case where the delivery is assumed by drone. Finally, the maximum allowed working time T_{max} is considered. For each delivered package, we associate a rent cost into the vehicle. Let $f^{r,D}$ and f^T a fixed rent cost of each package putted in a resource of the drone and truck, respectively.

Below, a summarize of the list of sets, parameters and variables used in this formulation.

Sets

N : set of n customers where $N = \{1, \dots, n\}$

Δ^+: set of nodes that can be reached from node $i \in N$

$\Delta^- i$: set of nodes that can be used from node $i \in N$

V: set of nodes

A: set of arcs with $A = \{(i,j): i,j \in V, i \neq j\}$

Parameters

st: starting depot

en: ending depot

V^T: speed of the truck T

V^D: speed of the drone D

d_{ij}^T: distance between two locations i and j traveled by each truck T

d_{ij}^T: non-negative travel time between two locations i and j traveled by each truck T

d_{ij}^T: distance between two locations i and j traveled by each drone D

d_{ij}^T: non-negative travel time between two locations i and j traveled by each drone D

c^T: operational cost for each traveling arc by the truck T

c^D: operational cost for each traveling arc by the drone D

Q^T: load capacity of each truck T

$Q^{r,D}$: available number of resources of type $r \in R = \{1, 2, 3\}$ in the drone D

H: battery capacity of the drone

$[T_i^-, T_i^+]$: starting and ending of the time windows of each customer i

q_i^r : demand requested by the customer i that can be serviced by resource type r of the drone D

q_i : demand can be served by a truck of customer i

s_i: service time

$f^{r,D}$: fixed rent cost of each package putted in the resource r of the drone D

f^T: fixed rent cost of each package putted in the truck

T_{max}: maximum working day

Variables

x_{ij}^T : binary variable equal to 1 if the truck T travel from i to j, and 0 otherwise.

x_{ij}^D : binary variable equal to 1 if the drone D travel from i to j, and 0 otherwise.

y_i^T : binary variable equal to 1 if the truck T is used to serve the customer i, and 0 otherwise.

y_i^D : binary variable equal to 1 if the drone D is used to serve the customer i, and 0 otherwise.

za_i: auxiliary variable that indicate if there is a drone arc is coming from node i

zb_i: auxiliary variable that indicate if there is a drone arc is entered to node i

la_i: auxiliary variable that indicate if the drone can be launched from node i or not

z_i^T: continuous variables that represent the service time of the truck T on node i.

z_i^D: continuous variables that represent the service time of the drone T on node i.

Q_i^T: continuous variables that represent the load on the truck immediately after servicing i.

$Q_i^{r,D}$: continuous variables that represent the load of resource r on the drone D immediately after servicing i.

o_i^+:continuous variables that represent the battery level of the drone when departing from the node i.

o_i^-: continuous variables that represent the battery level of the drone when departing to the node i.

We provide the following mixed-integer programming formulation for the VRP-D-MC Based on the VRP-D formulation of Gonzalez-R et al. (2020) and Kitjacharoenchai et al. (2020).

$$\text{Maximize } \left(\sum_{i\in V}\sum_{r\in R} f^{r,D} y_i^D + \sum_{i\in V} f^T y_i^T\right) - \left(\sum_{(i,j)\in A} c^T x_{ij}^T + \sum_{(i,j)\in A} c^D x_{ij}^D\right) \quad (1)$$

subject to

$$\sum_{j\in\delta^+(en)} x_{en,j}^D = 1 \quad (2)$$

$$\sum_{j\in\delta^-(st)} x_{i,st}^D = 1 \quad (3)$$

$$\sum_{j\in\delta^-(i)} x_{ji}^D \le 1 \qquad i\in V \quad (4)$$

$$\sum_{j\in\delta^+(i)} x_{ij}^D = \sum_{j\in\delta^-(i)} x_{ji}^D \qquad i\in V \quad (5)$$

$$\sum_{j\in\delta^+(en)} x_{en,j}^T = 1 \quad (6)$$

$$\sum_{j\in\delta^-(st)} x_{i,st}^T = 1 \quad (7)$$

$$\sum_{j\in\delta^-(i)} x_{ji}^T \le 1 \qquad i\in V \quad (8)$$

$$\sum_{j\in\delta^+(i)} x_{ij}^T = \sum_{j\in\delta^-(i)} x_{ji}^T \qquad i\in V \quad (9)$$

$$\sum_{i\in\delta^-(j)} x_{ij}^D + \sum_{i\in\delta^-(j)} x_{ij}^T \ge 1 \qquad j\in V \quad (10)$$

$$z_j^T \ge z_j^T + t_{ij}^T x_{ij}^T - M(1-x_{ij}^T) \qquad (i,j)\in A \quad (11)$$

$$z_j^D \ge z_j^D + t_{ij}^D x_{ij}^D - M(1-x_{ij}^D) \qquad (i,j)\in A \quad (12)$$

$$T_i^- \le z_i^T \le T_i^+ \qquad \forall i\in V \quad (13)$$

$$T_i^- \le z_i^D \le T_i^+ \qquad \forall i\in V \quad (14)$$

$$M(y_i^D - 1) + z_i^D - t_{0i}^D - z_0^D \le T \qquad \forall i\in V \quad (15)$$

$$\sum_{j\in N} x_{ij}^D (z_i^T - z_i^D) = 0 \qquad \forall i\in N \quad (16)$$

$$\sum_{j\in N} x_{ji}^D (z_i^T - z_i^D) = 0 \qquad \forall i\in N \quad (17)$$

$$Q_j^T \ge q_j x_{ij}^T + Q_i^T + \sum_{r\in R} Q_i^{r,D} - Q^T(1-x_{ij}^T) \qquad \forall i,j\in V \quad (18)$$

$$0 \le Q_j^T \le Q^T \qquad \forall j\in V \quad (19)$$

$$Q_j^{r,D} \ge q_j^r x_{ij}^{r,D} + Q_i^{r,D} - Q^{r,D}(1-x_{ij}^{r,D}) \qquad \forall i,j\in V, \forall r\in R \quad (20)$$

$$0 \le Q_j^{r,D} \le Q^{r,D} \qquad \forall j\in V, \forall r\in R \quad (21)$$

$$o_j^- \le H + M(2 - x_{ij}^D - x_{ij}^T) \qquad (i,j)\in A \quad (22)$$

$$o_j^- \ge H - M(2 - x_{ij}^D - x_{ij}^T) \qquad (i,j)\in A \quad (23)$$

$$o_j^+ \le H + M(2 - x_{ij}^D - x_{ij}^T) \qquad (i,j)\in A \quad (24)$$

$$o_j^+ \ge H - M(2 - x_{ij}^D - x_{ij}^T) \qquad (i,j)\in A \quad (25)$$

$$o_j^- \le o_i^+ - EC_{ij} d_{ij}^D x_{ij}^D + M\left(1 - x_{ij}^D + x_{ij}^T + \sum_{k\ne i} x_{kj}^T\right) \qquad (i,j)\in A \quad (26)$$

$$o_j^- \ge o_i^+ - EC_{ij} d_{ij}^D x_{ij}^D + M\left(1 - x_{ij}^D + x_{ij}^T + \sum_{k\ne i} x_{kj}^T\right) \qquad (i,j)\in A \quad (27)$$

$$o_j^+ \le o_i^- + M(1 - x_{ij}^D + x_{ij}^T + \sum_{k \ne i} x_{kj}^T) \qquad (i,j) \in A \qquad (28)$$

$$o_j^+ \ge o_i^- + M(1 - x_{ij}^D + x_{ij}^T + \sum_{k \ne i} x_{kj}^T) \qquad (i,j) \in A \qquad (29)$$

$$o_j^- \le o_i^+ - EC_{ij} d_{ij}^D x_{ij}^D - M(1 - x_{ij}^D + x_{ij}^T + 1 - \sum_{k \ne i} x_{kj}^T) \qquad (i,j) \in A \qquad (30)$$

$$o_j^- \ge o_i^+ - EC_{ij} d_{ij}^D x_{ij}^D - M(1 - x_{ij}^D + x_{ij}^T + 1 - \sum_{k \ne i} x_{kj}^T) \qquad (i,j) \in A \qquad (31)$$

$$o_j^+ \le H + M(1 - x_{ij}^D + x_{ij}^T + 1 - \sum_{k \ne i} x_{kj}^T) \qquad (i,j) \in A \qquad (32)$$

$$o_j^+ \ge H + M(1 - x_{ij}^D + x_{ij}^T + 1 - \sum_{k \ne i} x_{kj}^T) \qquad (i,j) \in A \qquad (33)$$

$$o_{st}^+ = H \qquad (34)$$

$$o_j^- \ge 0.1H \qquad \forall j \in V \qquad (35)$$

$$\sum_{j \in N} x_{ij}^D \ge (za_i - 1) = 0 \qquad \forall i \in N \qquad (36)$$

$$\sum_{j \in N} x_{ji}^D \ge (zb_i - 1) = 0 \qquad \forall i \in N \qquad (37)$$

$$\sum_{j \in N} x_{ij}^D \ge 1 - M(2 - y_i^D - za_i) \qquad \forall i \in N \qquad (38)$$

$$\sum_{j \in N} x_{ji}^D \ge 1 - M(2 - y_i^D - zb_i) \qquad \forall i \in N \qquad (39)$$

$$x_{ij}^D \ge 2 - (y_i^D - y_j^D) \qquad \forall i, j \in N \qquad (40)$$

$$x_{ji}^T (\sum_{\substack{s \in N \\ s \ne i}} x_{sj}^D) la_j = 0 \qquad \forall i, j \in N \qquad (41)$$

$$la_i (\sum_{j \in N} x_{ij}^D) = 0 \qquad \forall i \in N \qquad (42)$$

$$x_{ij}^T (\sum_{\substack{f \in N \\ f \ne j}} x_{iq}^D)(1 - \sum_{\substack{s \in N \\ s \ne i}} x_{sj}^D)(1 - la_j) = 0 \qquad \forall i, j \in N \qquad (43)$$

$$x_{ij}^T la_i (1 - \sum_{\substack{s \in N \\ s \ne i}} x_{sj}^D)(1 - la_j) = 0 \qquad \forall i, j \in N \qquad (44)$$

$$z_j^D \le T_{\max} - (s_j + t_{j,en}^D) \qquad \forall j \in V \qquad (45)$$

$$z_j^T \le T_{\max} - (s_j + t_{j,en}^T) \qquad \forall j \in V \qquad (46)$$

$$x_{ij}^T, x_{ij}^D, y_i^D, y_i^T, za_i, zb_i, la_i \in \{0,1\} \qquad \forall i, j \in V, i \ne j. \qquad (47)$$

The objective function (1) maximizes the total revenue. The first part of (1) represents the rent places costs of the packages while the second part is the total operational costs. Constraints (2) and (3) guarantee that each drone starts and ends its route at the depot. Constraints (4) guarantee that each customer can be served by one drone as maximum. The flow conservations are defined in constraints (5). Constraints (6-9) represents the same constraints definitions as in (2-5) but for the trucks. Constraints (10) guarantee that each node can be visited either by a drone or a truck or by both (drone and truck) if it is used as a rendezvous point. Constraints (11) and (12) define the service time. The service must be performed within the time windows as guaranteed by constraints (13) and (14). Constraint (15) impose that the waiting time of the drone at a customer *i* before starting service should not exceed a limited time *T*. Constraints (16) and (17) represents the synchronisation time when the drone and the truck merge at the rendezvous points. Constraints (18)-(21) guarantee that the capacity of

each vehicle type (trucks and drone) is respected. Constraints (18) takes into consideration the total demands of both truck and drone (in case when drone meet at customer location i to swap and to get load from the truck) that should be less than the capacity of the truck that is expressed in constraints (19)**.** Constraints (22)-(33) track the charge of the drone's battery level. More specifically, constraints (22)-(25) guarantee that the drones don't' consume energy if they travel with the truck during an arc (i, j). Constraints (23)-(29) define that the electricity level is reduced when the drone arrives at j according to the distance from i to j and the electricity consumption rate if i is the customer location node and node j is served immediately after i. Constraints (30)-(33) ensure that the drone's battery capacity is fully recharged (battery swap) after leaving node j. Constraints (34) guarantee that the battery is full recharged at the depot. Constraints (35) ensure the battery's safety security, in which the energy fuel should not exceed 10% of the capacity battery. Constraints (36) assure that the auxiliary variable za_i is equal to 1 when the drone departs from node i. Similarity, constraints (37) assure that the auxiliary variable zb_i is equal to 1 when the drone enters to node i. Constraints (38) assures that the node i is considered as a lunching node if the truck serve node i and there is a drone leave the same node i where it is lunched. While constraints (37) impose that the node i is considered as a lands node if the truck serve node i and there is a drone entered the same node i, where it is lands. In addition, the constraints (36) to (39) represent the case when the drone meet the truck at a rendezvous point to swap the battery and to pickup the packages.

Constraints (40) forbid a drone to directly travel from node *i* to node *j* where node *i* and node *j* are already served by a truck. Constraints (41). In constraints (43), the auxiliary variable la_j , must be equal to 1 when a drone is launched from node *i*, and a truck travels from node *i* to node *j*, which the drone has not yet returned to. Constraints (42) impose that a drone is not allowed to be launched or land at node *i* if la_i is equal to 1 and vice versa. Similarly, constraints (44) deal with the case when a drone was previously launched (not able to be launched at node *i* again) and has not returned to node *j*. Moreover, constraints (40) to (44) impose that the drone must be lunched before to land. Constraints (45) and (46) guarantee that the maximum route duration of each vehicle (truck and drone) is respected. Finally, constraints (47) specify binary decision variables.

The above mathematical formulation is impractical to solve even for small size problem instances. For this reason, in Section 5 we describe effective heuristic algorithms capable of solving large size problem instances.

4.2. Modeling of the energy consumption for electricity

Recently, research studies have incorporated realistic energy consumption functions in routing models for drone delivery problems (e.g., Poikonen and Golden, 2020). Thus, in addition to the routing problem, we also consider energy consumption by adopting the energy consumption proposed by Leishman (2006) and Cheng et al. (2020) to estimate the energy requirement. Furthermore, we capture

the mass of the drone as a variable in the energy consumption function. This modeling feature reflects practical consideration, as suggested in some existing studies (e.g., Masmoudi et al., 2018). In other words, the energy consumption depends on the current payload of the packages.

Hence the energy consumption E_{ij} when a drone uses an arc (i, j) with a load, whose weight is set to the remaining customer demand $Q_j^{r,D}$, is expressed as follows:

$$E_{ij}[\mathrm{W}] = \left(\sum_{r\in R} m\left(\left(Q_j^{r,D}\right) + F + m\right)^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\varepsilon\omega}}\right.$$

Where F and m represents the mass of the drone body and the battery, respectively. The gravitational constant is denoted by g, ρ is the air density, ε is the spinning area of one rotor and ω is the number of rotors. We denote $m\left(Q_j^{r,D}\right)$ the weight available in the drone upon the drone's arrival at the next customer j.

The detailed description, coefficients values, and properties used in our model are summarized in Table 2.

Table 2
Notation and parameter values used in the model

Notation	Definition	Value	Reference
g	gravitational constant (m/s^2)	9.807	Zhang et al. (2021)
F	mass of the drone body (kg)	7	Zhang et al. (2021)
m	mass of battery (kg)	10	Zhang et al. (2021)
ρ	air density	1.225	Zhang et al. (2021)
ε	spinning area of one rotor	3	Zhang et al. (2021)
ω	number of rotors	8	Zhang et al. (2021)
Q^D	capacity of the truck without drone (kg)	1300	Sacramento et al. (2019)
Q^T	capacity of the truck with drone (kg)	1400	Sacramento et al. (2019)
V^D	speed of the drone (mph)	50	Sacramento et al. (2019)
V^T	speed of the truck (mph)	35	Sacramento et al. (2019)
H	capacity of the battery (kW)	1,040,400 Joules (289Wh)	Troudi (2018)
c_D	operational cost for the drone (dollar/mile)	0.15	Salama and Srinivas (2020)
c_T	operational cost for the truck (dollar/mile)	1.25	Salama and Srinivas (2020)

5. Adaptative Multi-Start Simulated Annealing algorithm for the VRP-D-MC

Since the VRP-D is an NP-hard problem (Poikonen and Golden, 2020), the VRP-D-MC is also NP-hard since it is a generalization of VRP-D. Due to the complexity of the VRP-D, instances with up to 10 customers may be solved using the commercial solvers, such as CPLEX or Gurobi, with several hours (e.g., Murray and Chu, 2015; Yurek and Ozmutlu, 2018; Chung et al., 2020). VRP-D and its variants are usually solved with metaheuristic methods (e.g., Sacramento et al., 2019; Schermer et al., 2019; Kitjacharoenchai et al., 2019; Wang and Sheu, 2019; Poikonen and Golden, 2020). We propose a new Adaptative Multi-Start Simulated Annealing (AMS-SA) metaheuristic algorithm to solve VRP-D-MC.

The traditional SA is a single-solution-based metaheuristic first proposed by Kirkpatrick et al. (1984). It is successfully applied to various VRPs, including VRP-D (see, e.g., Lin and Vincent, 2012; Dorling et al., 2016; Yu et al., 2018). The traditional SA consist that in each SA iteration, a new solution

x' is generated using a neighborhood search operator based on the current solution x. The new solution x' is accepted to become the as the new current solution x if the objective function value of $x'(f\ (x'))$ is better than found in x. On the other hand, if the objective function of x' is higher, x' can be accepted subject to the simulated annealing acceptance criterion $e^{f(x)-f(x')/T_i}$, proposed by Metropolis et al. (1953), where T_i is the current temperature in the iteration i. The temperature cooling schedule is defined as: $T_i = \alpha * T_{i-1}$, where α is the cooling rate.

The advantage of the SA is that easy to implement, is flexible, and provides good solutions. However, the main disadvantage of simulated annealing is that once the algorithm is trapped in low temperature in a local minimum, it is impossible to get out. In addition, the difficulty of determining the initial temperature; if it is too low, the search quality will be bad. On other hand, if it is too high, the calculation time will be high. In this regard, we try to enhance the performance of SA and its convergence towards better quality solutions, by performing more intensification around the solutions and also diversifying the search to different regions. More specifically, we propose several modifications to the classic SA used in the literature by hybridizing it with other intensification and diversification procedures and techniques from population-based metaheuristic such as, Genetic Algorithm (GA), a multi-start approach, restoring the generation of the solution using the current neighborhood and also using efficient procedure to reduce the temperature. This hybridization process can boost the performance of a traditional SA and improve its convergence towards good solutions. The enhanced SA we propose is a novel method, proposed for solving the VRP-D-MC, but can be also seen as a new generalized algorithmic framework.

In the vast majority of the SA algorithms proposed in the literature, when the newly generated solution is not promising, the SA tries to generate a new solution using the neighborhood operators on the same current solution. In our approach, we instead construct a new solution by using the crossover operator mainly used in genetic algorithms. Specifically, we apply the crossover between a randomly selected best solution (x_{best}) found in a previous step best solution and a new solution generated by the constructive heuristic to build the restarting initial solution of the SA. The positive feedback of the crossover operator, in terms of transferring favorable characteristics from the parent to the child, helps to generate better solutions. The new solution will inherit the information from both the best solution, which was found by the algorithm in the current iteration, and the one newly generated by the constructive heuristic, which is characterized by a high diversity. This way, the construction of a new solution enable a good balance between the intensification around the best solutions, and the required diversification to explore new regions of the search space in a controlled, rather than arbitrary way.

Second, when the newly generated solution is promising, the current neighborhood is re-considered, increasing SA intensification power. This is in contrast to what is usually done in the traditional SA, where the search switches to a new operator/neighbourhood even when the newly generated solution is promising (see., e.g., Xiao and Konak, 2015; Masmoudi et al., 2016; Wei et al., 2018; Karagul et al.,

2019). This procedure allows to explore more deeply the region search abilities of the current neighbourhood structure around the current solution, and to focus the exploration in areas who has shown to be promising.

Finally, in each iteration and in order to explore different search regions, our algorithm restarts from a different initial solution, which further enhances diversification and permits escaping superfluous iterations around local optima. This technique is known under the name "multi-start approach" in the literature and has been successfully applied in a variety of metaheuristics such as the multi-start SA (e.g., Yu and Lin., 2014) and the multi-start VNS (e.g., Henke et al., 2015).

Moreover, we enhance the SA even more by increasing its intensification around the current solution, as will be explained later.

The AMS-SA developed in this paper is described in Algorithm 1. Initially, we set the temperature T to its maximum value T_{max}, let att, initialized to zero, be the number of the multi-start step, let i initialized to zero, be the number of iterations in which the best solution is not improved and let j initialized to zero, be the number of iterations at a selected neighborhood search structure is applied. Furthermore, the initial solution x and the best solution x_{best} are set equal and are generated using a constructive heuristic (Subsection 5.1).

The AMS-SA performs several runs n_{SA}. In each run, the algorithm starts with a new initial solution constructed using the constructive heuristic (line 4 of Algorithm 1). Within each run, several inner iterations n_{iter} are performed, such that in each iteration, a new solution x' is generated based on the current solution x using a selected neighborhood search from an available set of neighborhoods N1, N2, N3 or N4 (line 12 of Algorithm 1) as described in subsection 5.2. To explore the search space, our AMS-SA makes use of several neighborhood structures. The selection of the neighborhoods is not random. Still, it is instead determined based on their performance score at the previous iteration, which is in line with most SA implementations in the literature (e.g., Wei et al., 2018; Masmoudi et al., 2016). The neighborhoods' selection process is further described in subsection 4.4.

In our algorithm, a new solution is accepted if it improves the objective value or adheres to the Cauchy criterion (line 14 of Algorithm 1) proposed in Tiwari et al. (2006). According to Tiwari et al. (2006), the Cauchy probability function is more efficient to escape local optima than the Boltzmann function applied in the majority of SA algorithms reported in the literature. This is implemented by randomly generating a number $0 < \beta < 1$ and checking whether $\beta < T/(T^2 + \Delta E^2)$, where ΔE represents the difference in revenue between the current and the new solution. To foster a smooth annealing process, the initial temperature (T) is calculated as $T = \frac{-0.05}{\ln 0.5} * f(x)$, where $f(x)$ represents the initial solution revenue. The accepted new solution is further improved through a selected local search operator (line 16 of Algorithm 1) I1, I2, I3, or I4. The local search operator's choice is based on their previous iteration performance and is performed according to a roulette wheel rule. If the improved

solution is accepted (line 19 of Algorithm 1), the current neighborhood search is applied in the next iteration (line 22 of Algorithm 1) to intensify the search around the incumbent solution. Otherwise, if the improved solution is rejected (line 24 of Algorithm 1), a new solution is generated using the MX1 crossover operator (lines 27 of Algorithm 1). The newly developed solution inherits information from the randomly selected improved solution x_{best} (line 26 of Algorithm 1) and another highly diversified solution, using our constructive heuristic (line 25 of Algorithm 1). If improvements are achieved, the incumbent solution is updated (lines 28-31 of Algorithm 1).

A critical operation in the traditional SA consists of reducing the temperature using a cooling rate. In our approach, we opt for a more flexible way of controlling the temperature parameter. In other words, if the best solution x_{best} is not improved after a predefined consecutive number of runs (line 7 of Algorithm 1), the current temperature value is then reduced (line 8 of Algorithm 1), and the number of iterations n_{iter} is increased (line 9 of Algorithm 1). This overcomes the problem of quickly converging to a non-acceptable local optimum or slowly converging to an acceptable solution. The number of iterations n_{iter} is initialized to its default value (line 36 of Algorithm 1) when it reaches its maximum.

Algorithm 1: Adaptative Multi-Start Simulated Annealing

1. **Initialize** $i = 0$; $j = 0$; Temperature $T = T_{max}$; $x = x_{best}$ = the constructive heuristic; $att = 1$;
2. **Repeat**
3. **If** $att > 1$ **Then**
4. Construct a new solution using the constructive heuristic;
5. **End If**
6. $j:=j+1;$
7. **If** $i > 3$ **Then**
8. n_{iter}:=n_{iter}**+1;**
9. $T = \alpha * T$;
10. **End if**
11. **While** ($j<n_{iter}$)
12. Select a neighborhood search and generate a new solution x' based x;
13. Update the scores of the selected neighborhood search;
14. **If** $f(x') < f(x)$ **or** accepted by the Cauchy function **Then**
15. $x \leftarrow x'$;
16. Select a local search and apply it on x to obtain a new improved solution x'';
17. Update the scores of the selected local search;
18. **If** $f(x'') < f(x)$ **Then**
19. $x \leftarrow x''$;
20. **End if**
21. **If** the solution x is updated in line 16 or line 20;
22. Keep the current neighborhood search for the next j iteration
23. **End if**
24. **Else If** $f(x') > f(x)$ **Then**
25. $x_{new} \leftarrow$ generate a new constructive heuristic solution;
26. Select randomly a solution $x_{best;}$
27. $x \leftarrow$ Crossover (x_{best}, x_{new});
28. **If** $f(x) < f(x_{best})$ **then**
29. $x_{best} \leftarrow x$;
30. $x_{best} \leftarrow f(x_{best})$;
31. $i=0;$
32. **Else**
33. i++;
34. j++**;**
35. **End While**

36. $n_{iter} = 0$;
37. Adjust the weights of operators using the scores obtained aftern_{iter} iterations;
38. $att = att + 1$;
39. **Until** n_{SA} is reached
40. **Return** x_{best}

5.1. Construction heuristic

A modified insertion heuristic based on the one proposed by Kitjacharoenchai et al. (2020) and Gonzalez-R et al. (2020) is implemented. The constructive heuristic handles several features related to swapping battery of the drones, pickup of multiple packages of the drone, and the possibility that the drone to return at a different vehicle than the start vehicle. The heuristic starts with a list L that contains all biggest packages that can be serviced by the trucks to L' customers, i.e., the packages having weights or volumes greater than the weight or the volume of the drone. In addition, we create a list D that contains the smaller packages that can be carried by the drones to be served to D' customers. The heuristic is it based on two main phases.

In the first phase, we create an empty route for each vehicle with finite capacity. Customers from the list L' are selected randomly and inserted one by one in their best position on the existing routes if time windows, maximum route duration and vehicle capacity are respected. This phase is repeated until L' is empty. Figure 3-a represents the first phase of the constructive heuristic by building a partial solution where the trucks serve all the customers.

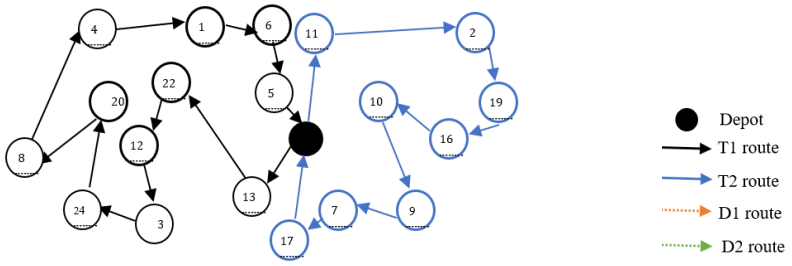


Figure 3a: First phase; partial solution where all customers are serviced by two trucks.

The second phase consist of building a sub routes that contains the drones. To do so, we adopt the relocation procedure of Fosin et al. (2014). For each customer in D, the selected customer is then removed from D and all possibility to insert the customer into the sub route of a selected drone are considered that respect the time windows and drone capacity. If the insertion is feasible then the next customer from D is selected and the procedure of the relocation is repeated to the same selected drone, otherwise another drone is selected.

In case where the drone cannot visit the next customer due to the energy constraint, the nearest truck located at a customer node to the current node is identified then the drone meet with this vehicle to swap the depleted battery. At the same time, the drone can carry some packages from this vehicle (if possible)

that respect the capacity constraint. Finally, the customer is returned to the vehicle route in case of any feasible insertion. This phase is repeated until all customers are checked.

Figure 3b-d represents the second phase of the proposed constructive heuristic by, deleting the customers that can be serviced by the drones, and by inserting the deleted customers to the different available drones.

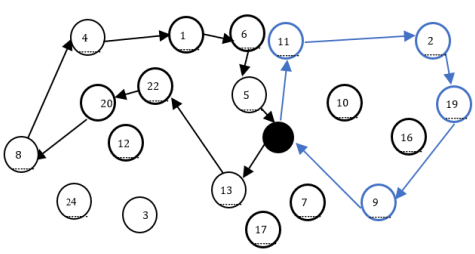


Figure 3b: Second phase; solution after deleting of a set of customers (12, 24, 3, 17, 7, 10 and 16) that can be serviced by the drone.

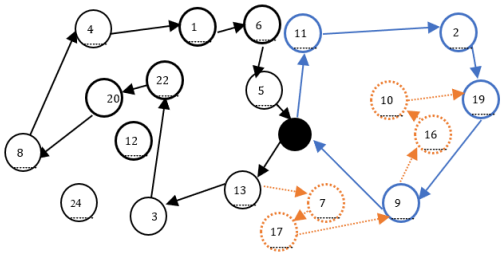


Figure 3c: Second phase; solution after insertion of the customers to the first drone where the drone is lunched from the T1, serve customers 7 and 17, meet the T2 to swap the battery, then serve customers 16 and 10.

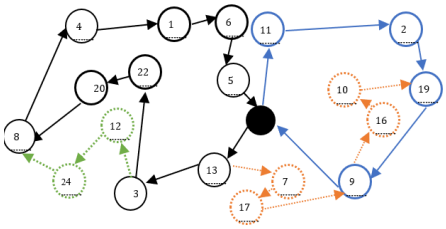
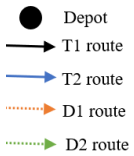


Figure 3d: Figure 2b: Second phase; solution after insertion of all customers to the drones.



5.2. Neighborhood structures

In this section, we describe the neighbourhood search operators embedded in our hybrid AMS-SA. They are an essential component that should satisfy the balance between conserving the promising and perturbating the less promising part of the current solution. In this regard, several well-known neighborhood operators adopted in the literature (e.g., Lin and Yu, 2012; Masmoudi et al., 2018) are applied to explore the current solution's search space. In particular, we implement four neighborhood structures (i.e., N1, N2, N3, and N4), which we describe below.

Exchange Neighborhood (N1): This structure consists of randomly choosing two routes with the same mode of transportation (trucks or drones). From one route, a random segment comprised of a sequence of consecutive nodes is cut that may consist of customers. The number of nodes in the cut segment is randomly chosen between one and five, depending on the number of customers in a route (Masmoudi et

al., 2018). From the removed segment customers are re-inserted one by one at their best positions in the other routes. When inserting a customer, the algorithm determines the best feasible insertion positions.

Swap Neighborhood (N2): The swap operator consists of swapping two customers of two different routes. To apply this operator, we first select two randomly routes. Then, a random customer is selected from a truck (drone) and swapped with another customer from another truck(drone). The first customer's is inserted in the same position as the second customer in the second route. While the second customer's is inserted at any place in the first route.

Cross-exchange (N3) -(N4): A segment of nodes of size *b* randomly selected is cut from a route (truck or drone) $r1$ and inserted in another route (truck or drone) $r2$. Sequentially, a segment of nodes of size *d* randomly selected between *b* and *b-1*, is cut from a route *r2* and inserted into route *r1*. This exchange results in an improved diversification, as proven in e.g., Hemmelmayr et al. (2009). Therefore, this operator is often applied at the perturbation stage, as demonstrated in e.g., Hemmelmayr et al., 2009; Masmoudi et al., 2018). For a feasible and effective swap, the segment length used in our operator's move is set to two and denoted by (**N3**) and set to three and referred by (**N4)**.

5.3. Local search operators

During each step of the AMS-SA algorithm, several well-known local search operators inspired and adapted from the literature studies are applied to improve the current solution. In particular, four operators with different movements are involved. Two inter-route operators namely 2-opt* proposed by Potvin and Rousseau (1995), and the remove two insert one operator presented by Xiang et al. (2016). Moreover, two intra-routes operators composed of the relocate operator proposed by Savelsbergh (1992) and the 2-opt operator Potvin and Rousseau (1995).

5.4. Adaptive weight adjustment procedure

In each iteration of the AMS-SA (as described in Algorithm 1), selecting an appropriate neighborhood and operator is made through a roulette wheel mechanism. Such a mechanism achieves a right balance between solution quality and running time. The probability of choosing a neighborhood operator *N* ($N \in$ {N1, N2, N3, N4}) at iteration *j*, is calculated as $P_N^{j+1} = P_N^1(1- r_p) + r_p\pi_N/\omega_N$, where r_p is the roulette wheel parameter, π_N is the score of an operator, and ω_N is the number of times the operator *N* is used in the last n_{iter} iterations. The score of an operator is updated according to the following procedure: i) if the operator finds a new best solution, the score is incremented by σ_1, ii) if it locates a better solution than the current one, the score is incremented by σ_2 iii) if it finds a non-improving feasible solution, the score is incremented by σ_3. After n_{iter} iterations, the weights are adjusted using the scores obtained. Similarly, the probability of choosing a local search operator *I* ($I \in$ {I1, I2, I3, I4}) is computed as $P_I^{j+1} = P_I^1(1- r_p) + r_p\pi_I/\omega_I$.

6. Experimental results

In this section, our aim is twofold. First, to test the performance of the algorithm against well-known benchmark instances. Second, we propose several experiments to analyze the features of VRP-D-MC. We have implemented the AMS-SA algorithm in C and the experiments were performed on a machine equipped with an Intel Core i5-10310U 2.21 GHz and 8 GB RAM.

6.1. Data set instances

Since the VRP-D-MC is new to the literature, we generate a new set of instances derived from the benchmark instances of Pugliese et al. (2020) for the drone routing problem, which are modified versions of the instances created by Solomon (1987) for the well-known VRPTW benchmark instances of. These instances consist of six categories, i.e., C1, C2, R1, R2, RC1 and RC2. Each data set contains between 50 and 100 customers or requests. All customers locations are randomly distributed data sets in categories C1 and C2. In classes R1 and R2 locations are within predefined clusters, while in the categories RC1 and RC2, the coordinates are generated as randomly and clustered. The coordinates nodes are randomly generated in the square area $[-10, 10]^2$. The data sets of C1, R1 and RC1 are characterized with a tight time window and longer travel time compared to shortest travel time and large time windows in C2, R2 and RC2. The weight of each package requested by the customers ranges between 1kg and 100 kg. Small size packages are subject to drone delivery (Bezos, 2013; Poikonen and Golden, 2020). To capture the limitation of drone delivery, we adopt similar ideas from Pugliese et al. (2020) for instance generation. We set that only half (50%) of the customers N can be served by drones under each instance. These customers have a smaller value of quantity q_i^r.The service time for each customer served by a truck is set to the original service time, while the service time by a drone is set to half of the service time by a truck (i.e., a drone is two times faster than a truck). We note that such a choice of the velocities of the trucks and drones provides higher feasibility of having drone and truck routes synchronized (Sacramento et al., 2019). Moreover, we associate for each allocation of a package in the vehicle (drone and truck) a rent cost that equal to 20$, 40$ and 60$ for a place of 20 cm^3, 40 cm^3 and 60 cm^3, respectively and 100$ for the package that surpasses 60 cm^3. The maximum waiting time T is equal to 10 min (Pugliese et al., 2020). The new data sets can be downloaded from on https://vrp-d-mc-47.webselfsite.net/.

6.2. Setting the parameters

Like most meta-heuristics available in the literature, the proposed AMS-SA algorithm's performance is expected to depend on the algorithm parameters, namely, the cooling rate α, the scores of the local search operators (π_1, π_2 and π_3), the number of iterations for a given temperature n_{iter}, and the number of iterations of the whole algorithm n_{SA}. The parameters values are chosen based on the recommendation and suggestions of previous experiment turning in the literature (α, π_1, π_2 and π_3) and based on our preliminary experiments ($n_{iter}$and n_{SA}.). In general, the values of parameters that are based on

experimental designs followed in the literature as follow; we set the cooling rate α to 0.99975 as suggested by Demir et al. (2012) and Masmoudi et al. (2016; 2020). We adopt the same parameter values as recommended in the most successful ALNS applied in the majority of VRP variants (e.g., Demir et al., 2012; Alinaghian et al., 2018; Žulj et al., 2018; Masmoudi et al., 2020). As such, we set the score values of the operators π_1, π_2 and π_3 as 15, 5 and 10, respectively. Two parameters setting may affect the performance of our algorithm which are the overall number of iterations of the algorithm n_{SA}, and the number of iterations for a given temperature n_{iter} are set to where n_{SA}=50,000 and n_{iter}=100, respectively. These values were defined based on preliminary tests we conducted using 20 instances with diverse features ranging from tight to large time windows, and with a different number of customers. Each instance is solved five times. The aggregate results of the preliminary tests are reported in Table 3. We have n_{SA} = 30.000, 50.000 and 100.000, and n_{iter} = 50, 100, 150, 200. The row Best (Avg) represents the average best of solution value found for each pair of parameters (n_{SA}, n_{iter}) of all the selected instances in the small data set, while the column CPU represents the average run time in minutes.

Table 3
Parameter setting of the AMS-SA

n_{SA}	30.000				50.000				100.000			
n_{iter}	50	100	150	200	50	**100**	150	200	50	100	150	200
Best	1301.79	1302.18	1302.40	1303.26	1304.39	**1305.61**	1304.94	1304.83	1305.94	1306.02	1306.98	1307.02
Avg	1286.55	1287.15	1289.89	1291.27	1303.23	**1299.87**	1300.48	1298.59	1292.73	1291.69	1291.16	1290.67
CPU (min)	2.58	2.72	2.90	2.80	2.72	**2.78**	2.84	2.86	2.95	3.38	3.48	3.66

In Table 3, we observe that on the one hand, no significant improvements are obtained when n_{SA} is set to be larger than 50.000 iterations. On the other hand, computation times increase, as expected, when we increase the number of iterations. Hence, n_{SA} =50,000 and n_{iter}=100 provide a good trade-off between solution quality and run time.

6.3. Testing the AMS-SA algorithm on benchmark instances

In addition to the newly generated instances described in subsection 6.1, we also run the algorithm on the instances of Sacramento et al. (2019) for the VRP-D. Table 4 reports the results for the small and medium-large size instances of Sacramento et al. (2019) that contain between 5 and 200 customers.

The performance of our algorithm is compared with the Hybrid Genetic Algorithm (HGA) method of Euchi and Sadok (2020) on the data set instances of Sacramento et al. (2019). For these instances, the objective function is to minimize the total travelled time for the delivery. A subset of medium-large size instances of 46 problems are considered in this work. Each instance is solved ten times by each of the algorithms (i.e., ours and the HGA algorithms). In Table 4, column "Best" ("Avg") report the best (average) solution values. The column "CPU" indicates CPU time in seconds. Column "BKS" reports the Best-Known Solution values. Furthermore, column "Best%" (Avg%) refers to the gap of the best (average) solutions compared to the best-known solution (BKS). Table 4 provides the results of run on

the medium and large sizes instances. Since AMS-SA and HGA were run on similar machines, the corresponding computing times can be compared directly.

Table 4
Comparison of AMS-SA with the HGA algorithm on medium and large instances of Euchi and Sadok (2020)

Inst	BKS[a]	HGA[b]					AMS-SA				
		Best	Best%	Avg	Avg%	CPU(s)	Best	Best%	Avg	Avg%	CPU(s)
100.10.1	6.36332	6.36332	0.00	6.3859	0.35	3.306	6.36332	0.00	6.36332	0.00	1.048
100.10.3	7.26979	8.11945	13.03	8.22664	14.52	3.32	7.26979	1.20	7.39332	2.92	2.917
100.10.4	6.89257	6.89257	0.00	7.04396	2.20	3.326	6.99257	1.45	7.0852	2.79	4.368
100.20.1	12.541	12.541	0.00	13.2443	5.61	3.332	12.541	0.00	12.76424	1.78	3.269
100.20.2	14.0118	14.0118	0.00	14.5979	4.18	6.339	14.0118	0.00	14.0118	0.00	4.07
100.20.3	12.4879	12.4879	0.00	12.5325	0.36	6.345	12.51459	0.21	12.51459	0.21	6.337
100.20.4	12.135	12.135	0.00	12.2157	0.67	8.351	12.135	0.00	12.24307	0.89	7.403
100.30.1	22.9686	23.5686	4.34	23.9006	5.81	6.356	22.9686	1.68	23.23507	2.86	4.76
100.30.2	22.0396	22.0396	0.00	22.135	0.43	7.375	22.0396	0.00	22.39161	1.60	7.108
100.30.3	23.2988	23.2988	0.00	23.5033	0.88	70.392	23.2988	0.00	23.2988	0.00	33.055
100.40.1	29.1397	29.1397	0.00	29.5463	1.40	9.427	29.6397	1.72	29.80102	2.27	3.425
100.40.2	30.361	30.361	0.00	30.4129	0.17	10.446	30.361	0.00	30.79356	1.42	8.286
100.40.3	29.0112	29.0112	0.00	29.0635	0.18	10.473	29.0112	0.00	29.09314	0.28	7.122
100.40.4	28.5904	28.5904	0.00	28.636	0.16	10.49	28.8904	1.05	29.0891	1.74	6.886
150.10.1	8.5865	8.5865	0.00	9.63644	12.23	0.495	8.5865	0.00	8.63456	0.56	0.1
150.10.2	8.14615	8.14615	0.00	10.9009	33.82	0.994	8.14615	0.00	8.14615	0.00	0.927
150.10.3	8.67899	9.70515	14.23	9.84302	15.85	1.522	8.67899	2.15	8.84433	4.10	1.305
150.10.4	8.9979	10.8302	22.55	10.9725	24.16	2.012	8.9979	1.82	9.05767	2.49	1.643
150.20.1	17.0099	17.0099	0.00	17.6175	3.57	2.323	17.0099	0.00	17.31543	1.80	2.075
150.20.2	16.1837	16.1837	0.00	16.6333	2.78	2.612	16.2837	0.62	16.58436	2.48	2.066
150.20.3	16.983	16.983	0.00	16.983	0.00	3.025	17.32011	1.98	17.49209	3.00	2.581
150.20.4	16.8102	16.8102	0.00	16.8102	0.00	3.339	16.87836	0.41	16.90572	0.57	1.354
150.30.1	25.8524	25.8524	0.00	25.8524	0.00	3.489	25.8524	0.00	25.92485	0.28	2.249
150.30.2	26.137	26.137	0.00	26.137	0.00	3.569	26.137	0.00	26.27017	0.51	1.541
150.30.3	25.0101	25.0101	0.00	25.0101	0.00	3.707	25.18698	0.71	25.23735	0.91	5.494
150.30.4	25.9812	25.9812	0.00	25.9812	0.00	3.863	25.9812	0.00	25.9812	0.00	3.79
150.40.1	29.4683	29.4683	0.00	29.8651	1.35	6.477	29.4683	0.00	29.93088	1.57	4.25
150.40.2	35.2343	35.2343	0.00	35.9885	2.14	6.997	35.57243	0.96	36.27218	2.95	7.436
150.40.3	36.8769	36.8769	0.60	36.8769	0.60	8.353	36.97655	0.87	37.16228	1.38	11.466
150.40.4	35.5854	35.5854	1.63	35.7586	2.12	10.29	35.67847	1.89	36.37678	3.89	11.46
200.10.1	10.0956	10.0956	0.01	10.1574	0.62	10.446	10.25848	1.62	10.39568	2.98	13.022
200.10.2	10.2855	10.2855	0.00	10.5303	2.38	10.597	10.2855	0.00	10.2855	0.00	7.048
200.10.3	9.10497	9.10497	0.00	9.10497	0.00	13.983	9.1988	1.03	9.31069	2.26	14.838
200.10.4	10.1251	10.1251	0.00	10.1251	0.00	15.617	10.1251	0.00	10.15315	0.28	12.996
200.20.1	21.2151	21.2185	0.02	21.2185	0.02	15.755	21.2151	0.00	21.43482	1.04	12.057
200.20.2	21.0193	21.0193	0.00	21.1948	0.83	17.884	21.0193	0.00	21.40677	1.84	16.344
200.20.3	19.0312	19.0312	0.00	19.0312	0.00	19.398	19.0312	0.00	19.0312	0.00	15.082
200.20.4	18.884	18.884	0.00	19.184	1.59	19.547	18.884	0.00	18.94839	0.34	17.563
200.30.1	30.073	30.073	0.00	30.683	2.03	19.699	30.46585	1.31	30.52591	1.51	20.538
200.30.2	32.0915	32.0915	0.00	32.0915	0.00	19.835	32.46581	1.17	32.51378	1.32	24.934
200.30.3	32.0835	32.0835	0.00	32.0967	0.04	21.699	32.0835	0.00	32.4495	1.14	19.011
200.30.4	32.0375	32.0375	0.00	32.0375	0.00	21.853	32.46979	1.35	33.05164	3.17	24.659
200.40.1	41.2954	41.2954	0.00	41.5796	0.69	22.018	41.2954	0.00	41.88108	1.42	17.765
200.40.2	43.0717	43.0717	0.00	43.1256	0.13	25.568	43.0717	0.00	43.82258	1.74	19.325
200.40.3	43.1857	43.1857	0.00	43.1857	0.00	26.182	43.46598	0.65	44.08434	2.08	28.089
200.40.4	42.0313	42.0313	0.00	42.0313	0.00	26.428	42.86598	1.99	42.92061	2.12	30.071
Avg	***22.04963***	***22.14336***	***1.23***	***22.38453***	***3.13***	***11.497***	***22.15204***	***0.61***	***22.36***	***1.49***	***9.85***

[a] *Best-known solution results provided from Euchi and Sadok (2020)*
[b] *Results provided by the ALNS of Euchi and Sadok (2020) programmed on C++ language and executed on Intel Core i5-2450M with 2.50 GHz and 4GB of RAM.*

Given the results of Table 4, we conclude that our algorithm outperforms on average the HGA of Euchi and Sadok (2020) even with small gap. Our AMS-SA improves the results of Euchi and Sadok

(2020) in the average deviation (Best%) for the best-known solutions results over ten runs by 0.61% compared to 1.23% for the HGA. The AMS-SA has an average deviation (Avg%) of 1.49%, against 3.74% of the HGA. In addition, our algorithm is faster than the HGA by reporting an average of 9.850 seconds compared to 11.497 seconds reported by the HGA. All in all and based on the results in Table 4, we conclude that our AMS-SA is more effective than HGA of Euchi and Sadok (2020). We believe that this is due to the additional diversification and intensification mechanisms introduced in our algorithm which allow the converge toward good quality solutions. In the next subsection, we analyse the impact of the different algorithm components to analyse their effectiveness.

6.4. Impact of the different AMS-SA features

In AMS-SA, several features namely, the multi-start approach, the crossover operator, restoring to the same neighborhood search, and controlling the temperature parameter by the Cauchy function are applied. We believe these components are crucial for the good performance of our algorithm. In this subsection, we evaluate the impact of each of these algorithmic components. We compare seven different algorithms, each implementing a different set of features. We denote these algorithms SA1, SA2, SA3, SA4, SA5, SA6, and SA7. Table 5 shows the features included in each of the algorithms.

Table 5
Combinations of algorithmic features

Features	SA1	SA2	SA3	SA4	SA5	SA6	SA7
Cauchy function	√	√	√	√	√	√	√
Returning to the same neighborhood operator in case the new solution is accepted		√		√	√	√	√
Reducing the temperature when x_{best} is not improved			√	√	√	√	√
Crossover operator					√		√
Multi-start approach						√	√

For instance, algorithm SA1 implements only the Cauchy function for controlling the temperature parameter (i.e., without crossover and multi-start), and hence it is equivalent to the traditional SA. Algorithm SA2 implements both the Cauchy function and the option to return to the same neighborhood operator in case the new solution is accepted. Algorithm SA7 implements all the features. In Table 6, we provide a comparison between these different algorithm configurations, where all variants are run on the VRP-D benchmark instances of Sacramento et al. (2019). We report the gap percentage "Best%" and "Avg%" values that represent the deviation gap from the best-known solutions (BKS).

Table 6
Components of the AMS-SA

Inst.	BKS[a]	HGA[b]		SA1		SA2		SA3		SA4		SA5		SA6		SA7	
		Avg%	Best%	Avg%	Best%	Avg%	Best%	Avg%	Best%	Avg%	Best%	Avg%	Best%	Avg%	Best%	Avg%	Best%
100.10.1	6.36332	0.35	0.00	3.47	2.66	2.93	1.85	2.66	1.59	2.45	1.32	1.59	0.53	2.12	0.26	0.00	0.00
100.10.3	7.18353	14.52	13.03	3.69	3.11	3.30	2.53	3.11	2.34	2.87	2.15	2.34	1.58	2.72	1.39	2.92	1.20
100.10.4	6.89257	2.20	0.00	10.60	8.42	9.14	6.28	8.42	5.57	7.45	4.88	5.57	2.81	6.98	2.13	2.79	1.45
100.20.1	12.54100	5.61	0.00	9.18	6.99	7.71	4.84	6.99	4.14	6.68	3.44	4.14	1.36	5.57	0.68	1.78	0.00
100.20.2	14.01180	4.18	0.00	1.90	1.46	1.61	1.02	1.46	0.87	1.42	0.73	0.87	0.29	1.20	0.15	0.00	0.00
100.20.3	12.48790	0.36	0.00	5.28	4.09	4.49	2.91	4.09	2.52	3.41	2.13	2.52	0.98	3.30	0.59	0.21	0.21
100.20.4	12.13500	0.67	0.00	1.85	1.42	1.57	0.99	1.42	0.85	1.18	0.71	0.85	0.28	1.11	0.14	0.89	0.00
100.30.1	22.58820	5.81	4.34	4.62	3.93	4.16	3.25	3.93	3.03	3.79	2.80	3.03	2.13	3.46	1.91	2.86	1.68
100.30.2	22.03960	0.43	0.00	3.79	2.91	3.20	2.02	2.91	1.73	2.44	1.44	1.73	0.57	2.32	0.29	1.60	0.00
100.30.3	23.29880	0.88	0.00	3.32	2.54	2.80	1.77	2.54	1.52	2.17	1.26	1.52	0.50	2.02	0.25	0.00	0.00
100.40.1	29.13970	1.40	0.00	5.04	4.27	4.53	3.49	4.27	3.24	4.05	2.98	3.24	2.22	3.74	1.97	2.27	1.72
100.40.2	30.36100	0.17	0.00	3.15	2.41	2.66	1.68	2.41	1.44	2.01	1.20	1.44	0.48	1.94	0.24	1.42	0.00
100.40.3	29.01120	0.18	0.00	4.03	3.09	3.40	2.15	3.09	1.84	2.58	1.53	1.84	0.61	2.48	0.30	0.28	0.00
100.40.4	28.59040	0.16	0.00	11.95	9.34	10.20	6.78	9.34	5.94	7.71	5.11	5.94	2.66	7.62	1.85	1.74	1.05
150.10.1	8.58650	12.23	0.00	2.18	1.67	1.84	1.17	1.67	1.00	1.54	0.83	1.00	0.33	1.33	0.17	0.56	0.00
150.10.2	8.14615	33.82	0.00	7.16	5.46	6.03	3.79	5.46	3.24	5.06	2.70	3.24	1.07	4.34	0.53	0.00	0.00
150.10.3	8.49602	15.85	14.23	12.26	9.84	10.64	7.47	9.84	6.70	8.63	5.93	6.70	3.65	8.26	2.90	4.10	2.15
150.10.4	8.83734	24.16	22.55	11.10	8.89	9.62	6.72	8.89	6.00	8.76	5.29	6.00	3.19	7.43	2.50	2.49	1.82
150.20.1	17.00990	3.57	0.00	12.00	9.11	10.07	6.29	9.11	5.37	8.06	4.46	5.37	1.76	7.23	0.88	1.80	0.00
150.20.2	16.18370	2.78	0.00	7.03	5.51	6.01	4.02	5.51	3.53	4.93	3.04	3.53	1.58	4.49	1.10	2.48	0.62
150.20.3	16.98300	0.00	0.00	12.60	10.06	10.90	7.57	10.06	6.75	9.42	5.94	6.75	3.55	8.40	2.76	3.00	1.98
150.20.4	16.81020	0.00	0.00	11.56	8.88	9.77	6.27	8.88	5.41	8.34	4.56	5.41	2.05	7.14	1.22	0.57	0.41
150.30.1	25.85240	0.00	0.00	3.21	2.46	2.71	1.71	2.46	1.47	2.20	1.22	1.47	0.49	1.96	0.24	0.28	0.00
150.30.2	26.13700	0.00	0.00	10.03	7.63	8.42	5.28	7.63	4.51	7.36	3.74	4.51	1.48	6.06	0.74	0.51	0.00
150.30.3	25.01010	0.00	0.00	10.87	8.44	9.25	6.06	8.44	5.28	7.75	4.50	5.28	2.21	6.84	1.46	0.91	0.71
150.30.4	25.98120	0.00	0.00	7.06	5.39	5.94	3.74	5.39	3.20	4.77	2.66	3.20	1.05	4.27	0.53	0.00	0.00
150.40.1	29.46830	1.35	0.00	8.12	6.19	6.83	4.30	6.19	3.67	5.06	3.05	3.67	1.21	4.93	0.60	1.57	0.00
150.40.2	35.23430	2.14	0.00	6.53	5.22	5.66	3.92	5.22	3.49	4.67	3.07	3.49	1.80	4.36	1.38	2.95	0.96
150.40.3	36.65740	0.60	0.60	3.48	2.87	3.07	2.26	2.87	2.06	2.62	1.86	2.06	1.27	2.46	1.07	1.38	0.87
150.40.4	35.01560	2.12	1.63	3.24	2.93	3.04	2.62	2.93	2.51	2.79	2.41	2.51	2.10	2.73	2.00	3.89	1.89
200.10.1	10.09450	0.62	0.01	12.43	9.84	10.70	7.31	9.84	6.48	8.28	5.65	6.48	3.22	8.18	2.42	2.98	1.62
200.10.2	10.28550	2.38	0.00	9.57	7.28	8.04	5.04	7.28	4.31	6.58	3.58	4.31	1.42	5.78	0.71	0.00	0.00
200.10.3	9.10497	0.00	0.00	2.94	2.50	2.65	2.06	2.50	1.91	2.21	1.76	1.91	1.32	2.21	1.18	2.26	1.03
200.10.4	10.12510	0.00	0.00	4.61	3.53	3.89	2.46	3.53	2.10	2.95	1.75	2.10	0.70	2.81	0.35	0.28	0.00
200.20.1	21.21510	0.02	0.02	10.99	8.56	9.37	6.18	8.56	5.40	7.14	4.63	5.40	2.34	6.95	1.58	1.04	0.00
200.20.2	21.01930	0.83	0.00	11.98	9.10	10.05	6.28	9.10	5.36	8.35	4.45	5.36	1.76	7.23	0.87	1.84	0.00
200.20.3	19.03120	0.00	0.00	11.85	9.00	9.94	6.22	9.00	5.30	8.73	4.40	5.30	1.74	7.14	0.87	0.00	0.00
200.20.4	18.88400	1.59	0.00	4.96	3.80	4.18	2.64	3.80	2.26	3.28	1.88	2.26	0.75	3.05	0.37	0.34	0.00
200.30.1	30.07300	2.03	0.00	13.39	10.48	11.44	7.64	10.48	6.71	8.61	5.79	6.71	3.08	8.57	2.19	1.51	1.31
200.30.2	32.09150	0.00	0.00	6.08	4.92	5.31	3.78	4.92	3.40	4.73	3.03	3.40	1.91	4.17	1.54	1.32	1.17
200.30.3	32.08350	0.04	0.00	7.86	5.99	6.61	4.16	5.99	3.55	5.94	2.95	3.55	1.17	4.76	0.58	1.14	0.00
200.30.4	32.03750	0.00	0.00	6.97	5.65	6.08	4.34	5.65	3.91	5.33	3.48	3.91	2.19	4.78	1.77	3.17	1.35
200.40.1	41.29540	0.69	0.00	3.85	2.95	3.25	2.05	2.95	1.76	2.92	1.46	1.76	0.58	2.36	0.29	1.42	0.00
200.40.2	43.07170	0.13	0.00	5.10	3.90	4.30	2.71	3.90	2.32	3.52	1.93	2.32	0.77	3.11	0.38	1.74	0.00
200.40.3	43.18570	0.00	0.00	4.07	3.27	3.54	2.48	3.27	2.21	2.98	1.95	2.21	1.17	2.74	0.91	2.08	0.65
200.40.4	42.03130	0.00	0.00	8.78	7.17	7.71	5.59	7.17	5.07	7.15	4.55	5.07	3.00	6.11	2.49	2.12	1.99
Avg	***22.01483***	***3.13***	***1.23***	***7.08***	***5.55***	***6.05***	***4.04***	***5.55***	***3.54***	***5.02***	***3.05***	***3.54***	***1.58***	***4.54***	***1.10***	***1.49***	***0.61***

[a] Best-known solution results provided from Euchi and Sadok (2020)
[b] Results provided by the ALNS of Euchi and Sadok (2020) programmed on C++ language and executed on Intel Core i5-2450M with 2.50 GHz and 4GB of RAM.

From Table 6, the HGA of Euchi and Sadok (2020) outperforms algorithm SA1. The average gaps for the best solution and the average solution of the SA1 from the best know solutions results are 5.55% and 7.05%, respectively. Furthermore, we observe a slight improvement, compared to the traditional SA (SA1), when we incorporate the new way of controlling the temperature parameter (i.e., algorithm SA2)

and the restoring to the same neighborhood operator technique (i.e., algorithm SA3). The average gaps are 4.04% (5.04%) and 3.54% (4.54%) for SA2, and SA3, respectively. Some improvements are achieved when combining the features of SA2 and SA3, resulting in algorithm SA4. Yet, the algorithm does not outperform the HGA of Euchi and Sadok (2020). In fact, an average gap improvement of 0.99% (to the best solution) and 1.02% (with regard to the average solution) compared to SA2, and 0.49% (to the best solution) and 0.52% (with regard to the average solution) compared to SA3. We believe this improvement is due to the diversification mechanism that facilitates exploring different search regions by using the new way of controlling the temperature parameter. The intensification capability is contributed by the restoring mechanism. We start observing the outperformance of our algorithm over the other combinations when we include the crossover operator (i.e., algorithm SA5). We only start outperforming HGA of Euchi and Sadok (2020) on the average of best solutions (columns Best%) when we include the multi-start approach resulting in algorithm SA6 by obtaining an average gap equal to 1.10% compared to 1.23% for the HGA.

On average, SA5 and SA6 deviate from the best-known results by 1.58% (3.54%) and 1.10% (4.53%), respectively. This can be explained by the good diversification capabilities of crossover in combination SA5. In addition, by including either the multi-start approach into the combination SA4, which refers to the combination SA7, or the crossover operator (SA6), we observe that the algorithm does not outperform HGA of Euchi and Sadok (2020). Thus, we believe that applying only this strategy cannot escape from the convergence to local optima. However, when applying together all components (features), we find that the new combination SA7 outperforms of the HGA in averages by reporting average gap of 0.60% (1.49%), compared to 1.23% (3.13%) obtained by Euchi and Sadok (2020). In conclusion, the adoption of all components provides both diversification and intensification during the search to the traditional SA and appears as the most effective combination, compared to the other combinations.

6.5. Effect of pickup packages under recharging energy

In this subsection, we investigate the impact of picking up packages during recharging energy. We consider two scenarios: (*i*) packages can be picked up by a drone during recharging on a truck (depending on the capacity), and (*ii*) packages cannot be picked up by a drone during recharging / swapping batteries on a truck. A small set of instances is selected with different characteristics, e.g., number of customers, tight and wide time windows, and different customers locations (clustered and randomly dispersed). The results of the two scenarios are reported in Table 7.

Table 7
An analysis of the benefits of using loading items under recharging drone battery

Inst.	First scenario		Second scenario				
	Best	Avg	Best	Best%	Avg	Avg%	CPU (min)
c102	1195.99	1182.30	1163.10	-2.75	1119.83	-6.37	0.95
c104	1253.90	1180.38	1250.64	-0.26	1233.76	-1.61	1.09
c105	825.55	812.20	819.94	-0.68	800.67	-3.01	1.66
c106	556.35	551.73	555.29	-0.19	548.57	-1.40	1.01
c108	1122.84	1087.83	1079.84	-3.83	1019.80	-9.18	1.66
c109	1281.53	1266.11	1269.23	-0.96	1244.73	-2.87	0.96
c202	1275.96	1223.24	1244.19	-2.49	1181.36	-7.41	2.50
c204	1330.49	1320.20	1279.00	-3.87	1198.30	-9.94	2.35
c206	1933.53	1921.67	1882.68	-2.63	1792.50	-7.29	2.11
c207	1660.66	1648.03	1589.58	-4.28	1477.99	-11.00	2.63
r102	910.65	899.30	884.33	-2.89	834.10	-8.41	2.72
r104	877.41	864.94	889.34	1.36	881.87	0.51	2.24
r105	747.90	738.11	735.11	-1.71	690.20	-7.72	4.33
r107	1391.46	1377.53	1338.31	-3.82	1344.06	-3.41	3.52
r110	1347.98	1337.17	1386.13	2.83	1366.03	1.34	2.54
r201	1209.47	1199.25	1214.31	0.40	1202.29	-0.59	3.58
r202	1092.47	1073.46	1064.61	-2.55	1007.76	-7.75	2.72
r206	1032.19	1005.27	1003.50	-2.78	952.12	-7.76	2.27
r207	2027.92	1993.21	2090.99	3.11	2126.54	4.86	1.46
r208	1631.45	1592.06	1586.59	-2.75	1517.41	-6.99	1.56
r210	1734.09	1631.64	1672.88	-3.53	1708.51	-1.48	1.45
rc105	974.47	959.33	987.14	1.30	975.79	0.14	0.87
rc107	1279.52	1240.29	1246.64	-2.57	1200.51	-6.17	1.10
rc108	959.95	926.84	952.37	-0.79	933.03	-2.80	1.22
rc202	1161.81	1132.85	1148.33	-1.16	1108.95	-4.55	2.24
rc203	1101.82	1068.76	1059.40	-3.85	992.34	-9.94	2.39
Avg	***1227.59***	***1201.30***	***1207.44***	***-1.59***	***1171.50***	***-4.65***	***2.04***

Table 7 suggests that the second scenario decrease the total revenue by an average gap of 1.59%. This is reasonable because a detour to a truck is required for picking up packages, and trucks may be needed to deliver the small packages, but they are slower than drones (Chung et al., 2020).

6.6. Impact of using different drones' configurations

In this section, we analyse the impact of using different drones' configuration. To do so, we consider three configurations for the resources of the drone that can be accommodated as described in the Table 8.

Table 8
Drone configurations

Configuration	Resources		
	20 cm^3	40 cm^3	60 cm^3
C1	4	0	0
C2	0	4	0
C3	0	0	4

Taking for example, the first configuration C1, that contain four places associated for the packages of size 20 cm^3, zero place for the packages of size of 40 cm^3 and zero place for the packages of size 60 cm^3.

Table 9 report the best (average) solutions results using each configuration C1, C2 and C3 of each solution found on each given instance. The columns Best%(Avg%) present the percentage of the deviation from the best solutions found by our main configuration denoted C0 of this paper which is two places associated for the packages of size 20 cm^3, one place for the packages of size of 40 cm^3 and one place for the packages of size 60 cm^3. We note that the trucks are used in tandem with the drone in each configuration.

Table 9
Results of using different drones' configurations profiles

	C0		C1				C2				C3			
Inst	Best	Avg	Best	Best%	Avg	Avg%	Best	Best%	Avg	Avg%	Best	Best%	Avg	Avg%
c101	886.89	882.97	882.10	-0.54	870.63	-1.83	884.85	-0.23	876.00	-1.23	882.81	-0.46	872.04	-1.67
c102	1191.05	1190.52	1188.91	-0.18	1177.73	-1.12	1188.91	-0.18	1177.73	-1.12	1190.22	-0.07	1180.34	-0.90
c103	1096.76	1094.83	1097.42	0.06	1094.35	-0.22	1093.69	-0.28	1092.16	-0.42	1097.31	0.05	1094.13	-0.24
c104	1247.78	1239.50	1237.80	-0.80	1226.53	-1.70	1247.16	-0.05	1245.04	-0.22	1240.67	-0.57	1232.23	-1.25
c105	804.72	796.95	804.16	-0.07	798.29	-0.80	804.64	-0.01	799.25	-0.68	805.04	0.04	800.13	-0.57
c106	552.53	537.74	549.10	-0.62	542.85	-1.75	550.93	-0.29	546.47	-1.10	552.59	0.01	549.77	-0.50
c107	1217.08	1214.33	1214.04	-0.25	1200.20	-1.39	1220.00	0.24	1212.07	-0.41	1217.20	0.01	1206.49	-0.87
c108	1105.68	1095.87	1097.50	-0.74	1082.57	-2.09	1103.91	-0.16	1097.40	-0.75	1100.37	-0.48	1088.27	-1.57
c109	1275.84	1271.97	1265.76	-0.79	1253.48	-1.75	1273.93	-0.15	1272.91	-0.23	1269.59	-0.49	1261.08	-1.16
c201	1067.41	1062.93	1065.17	-0.21	1059.74	-0.72	1069.22	0.17	1067.94	0.05	1065.81	-0.15	1061.12	-0.59
c202	1258.99	1251.63	1254.96	-0.32	1240.65	-1.46	1258.86	-0.01	1248.54	-0.83	1254.58	-0.35	1239.90	-1.52
c203	1586.48	1580.65	1586.64	0.01	1576.64	-0.62	1583.31	-0.20	1570.01	-1.04	1583.47	-0.19	1570.32	-1.02
c204	1318.80	1307.70	1313.52	-0.40	1305.91	-0.98	1315.11	-0.28	1309.06	-0.74	1313.92	-0.37	1306.69	-0.92
c205	1279.98	1279.70	1275.12	-0.38	1261.85	-1.42	1278.96	-0.08	1269.49	-0.82	1279.08	-0.07	1269.75	-0.80
c206	1924.72	1905.43	1916.06	-0.45	1902.26	-1.17	1920.29	-0.23	1910.69	-0.73	1918.37	-0.33	1906.86	-0.93
c207	1636.11	1618.28	1637.42	0.08	1627.59	-0.52	1631.69	-0.27	1620.11	-0.98	1631.04	-0.31	1614.73	-1.31
c208	1591.56	1590.86	1587.26	-0.27	1574.72	-1.06	1579.62	-0.75	1570.62	-1.32	1592.67	0.07	1585.35	-0.39
Avg C	***1237.79***	***1230.70***	***1233.70***	***-0.35***	***1223.29***	***-1.21***	***1235.59***	***-0.16***	***1228.56***	***-0.74***	***1234.98***	***-0.22***	***1225.84***	***-0.95***
r101	857.68	819.33	855.11	-0.30	850.49	-0.84	856.82	-0.10	853.91	-0.44	853.13	-0.53	846.57	-1.30
r102	865.05	828.71	858.91	-0.71	848.86	-1.87	869.29	0.49	863.38	-0.19	865.83	0.09	862.63	-0.28
r103	756.32	736.32	751.48	-0.64	742.99	-1.76	758.59	0.30	757.22	0.12	756.24	-0.01	752.54	-0.50
r104	842.43	819.54	838.22	-0.50	833.52	-1.06	843.10	0.08	843.19	0.09	840.83	-0.19	838.73	-0.44
r105	728.70	717.43	727.75	-0.13	723.46	-0.72	730.16	0.20	728.19	-0.07	727.46	-0.17	722.81	-0.81
r106	1371.38	1298.89	1372.48	0.08	1373.30	0.14	1371.79	0.03	1371.79	0.03	1365.89	-0.40	1360.02	-0.83
r107	1344.71	1301.23	1340.41	-0.32	1328.34	-1.22	1351.57	0.51	1342.38	-0.17	1339.20	-0.41	1325.94	-1.40
r108	734.55	725.05	733.96	-0.08	726.99	-1.03	733.52	-0.14	726.11	-1.15	730.51	-0.55	720.06	-1.97
r109	1184.05	1137.95	1179.08	-0.42	1166.23	-1.51	1185.94	0.16	1179.90	-0.35	1180.73	-0.28	1169.52	-1.23
r110	1318.89	1293.57	1321.00	0.16	1314.26	-0.35	1318.10	-0.06	1308.34	-0.80	1312.95	-0.45	1298.25	-1.56
r111	877.34	845.06	870.85	-0.74	857.09	-2.31	875.59	-0.20	866.48	-1.24	878.83	0.17	873.03	-0.49
r112	1169.42	1155.87	1165.91	-0.30	1154.49	-1.28	1170.12	0.06	1162.87	-0.56	1167.78	-0.14	1158.21	-0.96
r201	1145.02	1099.80	1138.15	-0.60	1131.21	-1.21	1145.25	0.02	1145.48	0.04	1141.81	-0.28	1138.62	-0.56
r202	1062.64	1061.96	1055.52	-0.67	1046.97	-1.47	1062.53	-0.01	1061.05	-0.15	1059.24	-0.32	1054.37	-0.78
r203	1130.49	1126.42	1123.59	-0.61	1111.35	-1.69	1129.25	-0.11	1122.58	-0.70	1130.49	0.00	1124.95	-0.49
r204	1104.19	1089.18	1098.12	-0.55	1082.19	-1.99	1101.43	-0.25	1088.76	-1.40	1098.12	-0.55	1082.08	-2.00
r205	1599.18	1591.84	1601.74	0.16	1592.13	-0.44	1601.42	0.14	1591.49	-0.48	1599.98	0.05	1588.62	-0.66
r206	997.14	983.22	989.66	-0.75	976.99	-2.02	996.54	-0.06	990.66	-0.65	993.25	-0.39	984.11	-1.31
r207	1936.24	1899.32	1926.75	-0.49	1905.37	-1.59	1940.50	0.22	1932.93	-0.17	1936.24	0.00	1924.24	-0.62
r208	1590.11	1589.14	1581.52	-0.54	1558.75	-1.97	1590.27	0.01	1576.12	-0.88	1585.18	-0.31	1566.00	-1.52
r209	1979.24	1967.86	1971.92	-0.37	1947.07	-1.63	1975.28	-0.20	1953.95	-1.28	1977.46	-0.09	1958.08	-1.07
r210	1716.47	1716.04	1705.31	-0.65	1680.42	-2.10	1721.62	0.30	1713.01	-0.20	1711.66	-0.28	1693.18	-1.36
r211	1611.58	1590.44	1604.01	-0.47	1591.65	-1.24	1606.91	-0.29	1597.59	-0.87	1613.03	0.09	1609.64	-0.12
Avg R	***1214.04***	***1191.05***	***1209.19***	***-0.41***	***1197.57***	***-1.35***	***1214.59***	***0.05***	***1207.71***	***-0.50***	***1211.56***	***-0.22***	***1202.27***	***-0.97***
rc101	705.96	697.64	702.85	-0.44	694.14	-1.67	702.50	-0.49	698.57	-1.05	704.62	-0.19	697.71	-1.17
rc102	919.33	900.22	912.34	-0.76	904.95	-1.56	916.66	-0.29	913.55	-0.63	914.55	-0.52	909.34	-1.09
rc103	879.11	859.40	875.86	-0.37	867.80	-1.29	881.66	0.29	879.46	0.04	876.03	-0.35	868.24	-1.24
rc104	763.37	759.67	757.80	-0.73	751.36	-1.57	763.14	-0.03	762.00	-0.18	762.76	-0.08	761.23	-0.28
rc105	943.89	922.49	944.83	0.10	944.65	0.08	944.27	0.04	943.61	-0.03	944.27	0.04	943.51	-0.04
rc106	672.74	658.70	671.87	-0.13	668.84	-0.58	671.19	-0.23	667.50	-0.78	672.40	-0.05	669.85	-0.43

rc107	1255.21	1240.28	1248.31	-0.55	1239.32	-1.27	1255.59	0.03	1253.83	-0.11	1251.07	-0.33	1244.81	-0.83
rc108	944.89	941.81	941.11	-0.40	930.66	-1.51	944.04	-0.09	936.49	-0.89	944.61	-0.03	937.62	-0.77
rc201	1541.24	1536.49	1534.46	-0.44	1522.80	-1.20	1538.62	-0.17	1531.08	-0.66	1539.70	-0.10	1533.23	-0.52
rc202	1131.43	1102.39	1130.98	-0.04	1123.06	-0.74	1129.96	-0.13	1120.92	-0.93	1129.73	-0.15	1120.47	-0.97
rc203	1063.97	1037.20	1055.99	-0.75	1039.52	-2.30	1065.35	0.13	1058.22	-0.54	1065.03	0.10	1057.58	-0.60
rc204	1519.90	1501.97	1516.56	-0.22	1499.57	-1.34	1515.95	-0.26	1498.51	-1.41	1512.76	-0.47	1492.03	-1.83
rc205	1461.94	1456.31	1463.40	0.10	1461.65	-0.02	1457.55	-0.30	1449.97	-0.82	1464.86	0.20	1464.57	0.18
rc206	1421.86	1410.88	1420.44	-0.10	1416.75	-0.36	1421.43	-0.03	1421.86	0.00	1415.32	-0.46	1406.54	-1.08
rc207	1336.45	1331.13	1336.72	0.02	1326.83	-0.72	1333.91	-0.19	1321.11	-1.15	1333.64	-0.21	1320.57	-1.19
rc208	1619.83	1607.53	1618.70	-0.07	1605.42	-0.89	1621.13	0.08	1610.43	-0.58	1614.32	-0.34	1596.73	-1.43
Avg RC	***1136.32***	***1122.76***	***1133.26***	***-0.30***	***1124.83***	***-1.06***	***1135.18***	***-0.10***	***1129.19***	***-0.61***	***1134.10***	***-0.18***	***1126.50***	***-0.83***
Avg	***1196.05***	***1181.50***	***1192.05***	***-0.35***	***1181.90***	***-1.21***	***1195.12***	***-0.07***	***1188.49***	***-0.61***	***1193.55***	***-0.20***	***1184.87***	***-0.92***

The results show that using the second configuration C2 provide a similar result with the respect of the main configuration C0. Using configuration C1 lead to increase the revenue cost and provide higher gap compared to the other configurations with an average gap equal to 0.35% compared to 0.07 % for C2 and 0.20% for C3. In addition, we can see that using the configuration C2 in data set R where the customers are randomly located provide much better results than the other configurations with an improvement (positive value) of 0.05%. We can conclude that by choosing the right drone configuration, the revenue costs can be significantly augmented, and that the magnitude of the increasing also depends on the nature of the instances (clustered or random).

6.7. Impact of the time windows

In this subsection, we evaluate the impact of using the time windows. We are interested in making a comparison between instances with and without time windows and see how this could affect the performance of the VRP-D-MC solution. To do so, we keep the same instances described in subsection 6.1 by removing the time windows. The results with and without time windows are given in Table 10 by reporting the waiting time on each instance. Columns "Best" ("Avg") represent the best (average) solutions values for both with and without time windows cases. The column "Best%" ("Avg%") indicates the percentage of deviation from the best solutions established by using time windows (our main problem). The column "Waiting" presents the waiting time. We note that the positive percent deviations when deleting time windows indicate an improvement in solution with respect to the best value found by using time windows.

Table 10
Comparison between with and without time windows

Inst	With Time Windows				Without Time Windows					
	Best	Avg	Waiting	CPU(min)	Best	Best%	Avg	Avg%	Waiting	CPU(min)
c101	886.89	882.97	14.81	1.41	959.30	8.16	876.37	-1.19	0.00	1.01
c102	1191.05	1190.52	21.23	1.56	1394.47	17.08	1155.58	-2.98	0.00	1.13
c103	1096.76	1094.83	11.31	1.34	1178.31	7.44	1088.47	-0.76	0.00	0.96
c104	1247.78	1239.50	14.66	0.82	1370.87	9.87	1225.58	-1.78	0.00	0.62
c105	804.72	796.95	14.45	1.10	874.07	8.62	789.49	-1.89	0.00	0.78
c106	552.53	537.74	15.28	1.16	572.97	3.70	535.43	-3.09	0.00	0.85
c107	1217.08	1214.33	13.61	2.16	1321.32	8.56	1204.90	-1.00	0.00	1.57
c108	1105.68	1095.87	15.03	2.18	1200.99	8.62	1085.79	-1.80	0.00	1.62
c109	1275.84	1271.97	16.15	2.38	1441.43	12.98	1249.39	-2.07	0.00	1.70
Avg C1	***1042.04***	***1036.08***	***15.17***	***1.57***	***1145.97***	***9.45***	***1023.44***	***-1.84***	***0.00***	***1.14***
c201	1067.41	1062.93	7.44	2.25	1106.80	3.69	1061.12	-0.59	0.00	1.72

c202	1258.99	1251.63	8.06	2.94	1307.65	3.86	1249.12	-0.78	0.00	2.22
c203	1586.48	1580.65	6.37	2.13	1668.23	5.15	1575.80	-0.67	0.00	1.65
c204	1318.80	1307.70	9.19	3.35	1368.26	3.75	1304.90	-1.05	0.00	2.48
c205	1279.98	1279.70	7.95	2.78	1349.02	5.39	1275.94	-0.32	0.00	2.01
c206	1924.72	1905.43	9.29	2.91	1981.64	2.96	1902.38	-1.16	0.00	2.15
c207	1636.11	1618.28	8.78	3.44	1703.29	4.11	1613.81	-1.36	0.00	2.61
c208	1591.56	1590.86	9.25	2.93	1679.48	5.52	1585.92	-0.35	0.00	2.25
Avg C2	***1458.01***	***1449.65***	***8.29***	***2.84***	***1520.54***	***4.30***	***1446.13***	***-0.79***	***0.00***	***2.14***
r101	857.68	819.33	5.47	0.90	874.39	1.95	815.63	-4.90	0.00	0.66
r102	865.05	828.71	6.02	1.02	876.55	1.33	825.95	-4.52	0.00	0.76
r103	756.32	736.32	5.10	1.56	767.48	1.48	735.00	-2.82	0.00	1.11
r104	842.43	819.54	7.16	0.95	881.37	4.62	814.88	-3.27	0.00	0.70
r105	728.70	717.43	6.03	1.60	770.77	5.77	713.46	-2.09	0.00	1.15
r106	1371.38	1298.89	6.27	0.90	1369.28	-0.15	1295.07	-5.56	0.00	0.64
r107	1344.71	1301.23	6.54	2.36	1385.23	3.01	1295.81	-3.64	0.00	1.68
r108	734.55	725.05	6.11	2.24	764.20	4.04	722.94	-1.58	0.00	1.61
r109	1184.05	1137.95	5.91	2.12	1195.73	0.99	1135.02	-4.14	0.00	1.59
r110	1318.89	1293.57	6.46	2.48	1371.54	3.99	1288.87	-2.28	0.00	1.81
r111	877.34	845.06	6.15	2.60	880.91	0.41	843.54	-3.85	0.00	1.84
r112	1169.42	1155.87	4.45	2.19	1186.02	1.42	1155.08	-1.23	0.00	1.65
Avg R1	***1004.21***	***973.25***	***5.97***	***1.74***	***1026.96***	***2.40***	***970.10***	***-3.32***	***0.00***	***1.27***
r201	1145.02	1099.80	0.51	3.90	1134.38	-0.93	1098.71	-4.04	0.00	2.86
r202	1062.64	1061.96	0.94	3.24	1070.05	0.70	1061.90	-0.07	0.00	2.35
r203	1130.49	1126.42	0.00	2.47	1126.42	-0.36	1126.42	-0.36	0.00	1.82
r204	1104.19	1089.18	0.00	3.50	1102.86	-0.12	1089.01	-1.37	0.00	2.55
r205	1599.18	1591.84	1.41	2.58	1591.84	-0.46	1591.84	-0.46	0.00	1.94
r206	997.14	983.22	0.00	3.04	1011.39	1.43	982.41	-1.48	0.00	2.14
r207	1936.24	1899.32	2.06	3.71	1929.29	-0.36	1898.85	-1.93	0.00	2.82
r208	1590.11	1589.14	0.81	4.55	1662.53	4.55	1585.75	-0.27	0.00	3.37
r209	1979.24	1967.86	0.00	4.56	2003.45	1.22	1967.22	-0.61	0.00	3.34
r210	1716.47	1716.04	0.00	4.09	1733.37	0.98	1715.87	-0.04	0.00	3.12
r211	1611.58	1590.44	2.58	5.94	1610.80	-0.05	1590.18	-1.33	0.00	4.30
Avg R2	***1442.94***	***1428.66***	***0.76***	***3.78***	***1452.40***	***0.60***	***1428.01***	***-1.09***	***0.00***	***2.78***
rc101	705.96	697.64	10.56	1.40	741.69	5.06	694.86	-1.57	0.00	1.05
rc102	919.33	900.22	11.90	1.51	958.36	4.25	896.47	-2.49	0.00	1.14
rc103	879.11	859.40	8.88	1.38	913.79	3.95	855.96	-2.63	0.00	1.00
rc104	763.37	759.67	11.96	0.84	804.80	5.43	756.99	-0.84	0.00	0.63
rc105	943.89	922.49	9.94	1.11	975.34	3.33	919.46	-2.59	0.00	0.79
rc106	672.74	658.70	10.90	1.10	700.87	4.18	656.00	-2.49	0.00	0.82
rc107	1255.21	1240.28	10.97	2.12	1338.40	6.63	1232.52	-1.81	0.00	1.56
rc108	944.89	941.81	11.37	2.40	1002.49	6.10	937.90	-0.74	0.00	1.77
Avg RC1	***885.56***	***872.53***	***10.81***	***1.48***	***929.47***	***4.86***	***868.77***	***-1.89***	***0.00***	***1.10***
rc201	1541.24	1536.49	5.30	2.44	1623.12	5.31	1531.61	-0.63	0.00	1.82
rc202	1131.43	1102.39	1.57	2.18	1139.76	0.74	1101.12	-2.68	0.00	1.65
rc203	1063.97	1037.20	0.00	2.90	1067.98	0.38	1036.29	-2.60	0.00	2.05
rc204	1519.90	1501.97	2.46	2.18	1577.74	3.81	1498.15	-1.43	0.00	1.60
rc205	1461.94	1456.31	0.00	3.16	1527.70	4.50	1452.81	-0.62	0.00	2.28
rc206	1421.86	1410.88	5.10	2.95	1462.11	2.83	1409.02	-0.90	0.00	2.12
rc207	1336.45	1331.13	2.41	2.98	1389.36	3.96	1328.58	-0.59	0.00	2.19
rc208	1619.83	1607.53	5.83	3.26	1651.73	1.97	1606.31	-0.83	0.00	2.49
Avg RC2	***1387.08***	***1372.99***	***2.83***	***2.76***	***1429.94***	***2.94***	***1370.49***	***-1.29***	***0.00***	***2.03***
Avg	***1203.30***	***1188.86***	***7.31***	***2.36***	***1250.88***	***4.09***	***1184.49***	***-1.70***	***0.00***	***1.74***

From the Table 10, we can observe that in many instances a positive percent values are obtained when deleting time windows with an augmentation of 4.09% on the total revenue (last line). More specifically, high improvement is obtained in instances of type C1, R1 and RC1 where tight time windows are considered with an average gap equal to 9.45%, 2.40% and 4.86% compared to their counterparts C2, R2 and RC2 with an average gap equal to 4.30%, 0.60% and 2.94%. In addition, we

can observe that the instances of time C1 where the customers are randomly located provide the high saving costs with 9.45% while R2 that characterized with the customers are clustered and with the large time windows provide the less saving costs with 0.60%. The same observation can be given for the waiting time in which the instances of C1 provide about 15 minutes while the waiting time of the instances of type R2 is about one minute. We also note that in some instances, the waiting time is equal to zero under the time windows scenario

7. Managerial insights

In this section, several managerial insights have been conducted. In the first experiment, we aim to quantify the benefit of the "flexibility of returning the drone to another vehicle". In the second experiment, we examine the impacts of using trucks only (i.e., without drones). In the third experiment, we analyse the impact of using drones with only a single compartment.

7.1. Impact of flexible returning a drone to another vehicle

We compare the performances of the VRP-D-MC where drones returning to another truck is allowed or not allowed. In Table 11, we denote by "Flexible Return" that refer to our proposed VRP-D-MC and by "Fixed Return" where each drone must return to the same vehicle where it is launched.

The columns "Best" and "Avg" in Table 11 report the best and average solution values, respectively. Moreover, the columns "Best%" and "Avg%" present the percentage of deviation from the best solution found when constraining the drone to return to the same vehicle (i.e., "Fixed Return"), and the column "CPU" refers to the average solution time in minutes.

Table 11
Comparison between VRP-D-MC with flexible returning and VRP-D-MC with fixed Returning

Inst	Fixed Return			Flexible Return				
	Best	Avg	CPU (min)	Best	Best%	Avg	Avg%	CPU (min)
c101	886.89	882.97	1.41	887.90	0.11	879.23	-0.86	1.24
c102	1191.05	1190.52	1.56	1195.99	0.41	1182.30	-0.73	1.36
c103	1096.76	1094.83	1.34	1109.86	1.19	1093.55	-0.29	1.23
c104	1247.78	1239.50	0.82	1253.90	0.49	1180.38	-5.40	0.78
c105	804.72	796.95	1.10	825.55	2.59	812.20	0.93	1.04
c106	552.53	537.74	1.16	556.35	0.69	551.73	-0.14	1.04
c107	1217.08	1214.33	2.16	1230.75	1.12	1176.34	-3.35	2.11
c108	1105.68	1095.87	2.18	1122.84	1.55	1087.83	-1.61	2.07
c109	1275.84	1271.97	2.38	1281.53	0.45	1266.11	-0.76	2.05
c201	1067.41	1062.93	2.25	1089.77	2.09	1058.43	-0.84	2.06
c202	1258.99	1251.63	2.94	1275.96	1.35	1223.24	-2.84	2.84
c203	1586.48	1580.65	2.13	1594.00	0.47	1583.19	-0.21	1.94
c204	1318.80	1307.70	3.35	1330.49	0.89	1320.20	0.11	3.06
c205	1279.98	1279.70	2.78	1314.21	2.67	1294.86	1.16	2.57
c206	1924.72	1905.43	2.91	1933.53	0.46	1921.67	-0.16	2.42
c207	1636.11	1618.28	3.44	1660.66	1.50	1648.03	0.73	3.18
c208	1591.56	1590.86	2.93	1594.90	0.21	1567.81	-1.49	2.67
Avg C	***1237.79***	***1230.70***	***2.17***	***1250.48***	***1.07***	***1226.30***	***-0.93***	***1.98***
r101	857.68	819.33	0.90	899.08	4.83	895.21	4.38	0.82
r102	865.05	828.71	1.02	910.65	5.27	899.30	3.96	0.90
r103	756.32	736.32	1.56	778.50	2.93	765.14	1.17	1.44

r104	842.43	819.54	0.95	877.41	4.15	864.94	2.67	0.89
r105	728.70	717.43	1.60	747.90	2.63	738.11	1.29	1.49
r106	1371.38	1298.89	0.90	1449.29	5.68	1426.42	4.01	0.88
r107	1344.71	1301.23	2.36	1391.46	3.48	1377.53	2.44	2.10
r108	734.55	725.05	2.24	753.70	2.61	749.18	1.99	2.01
r109	1184.05	1137.95	2.12	1238.25	4.58	1234.48	4.26	1.86
r110	1318.89	1293.57	2.48	1347.98	2.21	1337.17	1.39	2.25
r111	877.34	845.06	2.60	920.74	4.95	906.72	3.35	2.41
r112	1169.42	1155.87	2.19	1213.54	3.77	1205.33	3.07	1.95
r201	1145.02	1099.80	3.90	1209.47	5.63	1199.25	4.74	3.67
r202	1062.64	1061.96	3.24	1092.47	2.81	1073.46	1.02	2.99
r203	1130.49	1126.42	2.47	1154.86	2.16	1144.30	1.22	2.10
r204	1104.19	1089.18	3.50	1148.71	4.03	1137.15	2.98	3.27
r205	1599.18	1591.84	2.58	1642.81	2.73	1624.31	1.57	2.26
r206	997.14	983.22	3.04	1032.19	3.52	1005.27	0.82	2.88
r207	1936.24	1899.32	3.71	2027.92	4.73	1993.21	2.94	3.38
r208	1590.11	1589.14	4.55	1631.45	2.60	1592.06	0.12	4.14
r209	1979.24	1967.86	4.56	2033.34	2.73	2010.91	1.60	4.07
r210	1716.47	1716.04	4.09	1734.09	1.03	1631.64	-4.94	3.77
r211	1611.58	1590.44	5.94	1648.75	2.31	1637.54	1.61	5.21
Avg R	***1214.04***	***1191.05***	***2.72***	***1255.85***	***3.54***	***1236.90***	***2.07***	***2.47***
rc101	705.96	697.64	1.40	718.49	1.77	707.82	0.26	1.28
rc102	919.33	900.22	1.51	942.41	2.51	927.25	0.86	1.41
rc103	879.11	859.40	1.38	906.71	3.14	896.03	1.92	1.30
rc104	763.37	759.67	0.84	775.43	1.58	762.91	-0.06	0.78
rc105	943.89	922.49	1.11	974.47	3.24	959.33	1.64	1.05
rc106	672.74	658.70	1.10	691.98	2.86	673.80	0.16	1.05
rc107	1255.21	1240.28	2.12	1279.52	1.94	1240.29	-1.19	1.93
rc108	944.89	941.81	2.40	959.95	1.59	926.84	-1.91	2.26
rc201	1541.24	1536.49	2.44	1558.91	1.15	1536.55	-0.30	2.20
rc202	1131.43	1102.39	2.18	1161.81	2.69	1132.85	0.13	2.03
rc203	1063.97	1037.20	2.90	1101.82	3.56	1068.76	0.45	2.74
rc204	1519.90	1501.97	2.18	1565.70	3.01	1531.58	0.77	1.99
rc205	1461.94	1456.31	3.16	1482.88	1.43	1459.60	-0.16	2.80
rc206	1421.86	1410.88	2.95	1462.39	2.85	1438.46	1.17	2.75
rc207	1336.45	1331.13	2.98	1349.42	0.97	1339.64	0.24	2.55
rc208	1619.83	1607.53	3.26	1644.62	1.53	1626.12	0.39	2.83
Avg RC	***1136.32***	***1122.76***	***2.12***	***1161.03***	***2.24***	***1139.24***	***0.27***	***1.93***
Avg	***1196.05***	***1181.50***	***2.33***	***1222.45***	***2.28***	***1200.81***	***0.47***	***2.13***

The adoption of “Flexible Return” improves the gap on average by 2.28%, as compared to “Fixed Return)”. From Table 11, we observe that positive percentages (“Best%”) are obtained in all the instances. The improvements are higher for instances within category R with a gap ranging between 1.03% and 5.68%. The reason is that fixing the base of a drone to a truck would restrict the distance between them because drone cannot fly a long range. Therefore, the flexibility of allowing a drone to return to another truck is beneficial. This observation is further verified by the gaps for the solutions resulting from the clustered and mixture/randomly locations within categories C and RC. These gaps are lower and range between 0.11% and 2.67% for category C and between 0.97% and 3.56% for category RC.

7.2. Impact of using multi-compartment drones in tandem with trucks

In this section, we analyze the impact of using trucks without drones to serve the customers. In Table 12, all columns have the same titles and meanings as in Table 12, except that "Without drones" and "With drones" represent the problem settings where drones are used or not used in the delivery process.

Table 12
Impact of using drones with the trucks

Inst	Without drone			With drones				
	Best	Avg	CPU (min)	Best	Best%	Avg	Avg%	CPU (min)
c101	882.57	878.34	0.73	888.43	0.66	886.23	0.41	0.95
c102	1174.82	1157.67	0.82	1196.54	1.85	1182.84	0.68	1.04
c103	1059.47	1022.39	1.33	1099.38	3.77	1094.06	3.26	0.96
c104	1218.41	1186.00	0.87	1256.88	3.16	1250.07	2.60	0.58
c105	779.07	744.64	1.44	811.84	4.21	810.43	4.03	0.77
c106	535.32	529.32	0.84	556.56	3.97	551.94	3.10	0.80
c107	1202.81	1189.58	2.14	1240.18	3.11	1233.98	2.59	1.62
c108	1092.07	1068.38	1.88	1125.81	3.09	1092.56	0.04	1.55
c109	1243.21	1207.66	1.59	1283.97	3.28	1268.52	2.04	1.58
c201	1069.17	1063.93	2.24	1098.98	2.79	1087.52	1.72	1.50
c202	1258.22	1256.46	2.47	1283.41	2.00	1281.77	1.87	2.10
c203	1543.31	1506.27	1.89	1600.34	3.70	1589.48	2.99	1.44
c204	1311.73	1293.63	3.66	1342.36	2.34	1331.97	1.54	2.38
c205	1281.09	1279.43	2.84	1321.63	3.16	1302.17	1.65	1.89
c206	1904.14	1899.19	1.67	1945.75	2.19	1933.81	1.56	1.84
c207	1612.67	1570.90	3.19	1671.38	3.64	1658.67	2.85	2.34
c208	1567.95	1543.64	2.28	1600.33	2.07	1573.15	0.33	1.96
Avg C	***1219.77***	***1199.85***	***1.88***	***1254.34***	***2.88***	***1242.89***	***1.96***	***1.49***
r101	885.14	870.27	1.17	902.65	1.98	900.65	1.75	0.64
r102	882.88	861.77	1.25	897.32	1.64	889.48	0.75	0.7
r103	744.48	713.36	1.24	767.56	3.10	766.55	2.96	1.09
r104	911.01	865.56	0.77	919.18	0.90	914.97	0.43	0.65
r105	726.58	712.63	1.03	739.43	1.77	728.17	0.22	1.1
r106	1427.26	1404.71	1.03	1433.84	0.46	1429.6	0.16	0.64
r107	1322.44	1255.40	1.92	1363.54	3.11	1337.2	1.12	1.61
r108	727.62	709.80	2.07	746.00	2.53	733.74	0.84	1.55
r109	1287.04	1231.05	1.95	1290.45	0.26	1288.99	0.15	1.36
r110	1393.95	1343.07	1.91	1404.52	0.76	1399.98	0.43	1.71
r111	931.05	865.79	2.74	939.19	0.87	933.75	0.29	1.81
r112	1237.81	1184.71	1.74	1246.97	0.74	1243.1	0.43	1.43
r201	1221.08	1194.46	2.87	1232.03	0.90	1228.39	0.60	2.83
r202	1048.55	1033.24	2.51	1076.07	2.62	1059.09	1.01	2.2
r203	1132.46	1125.89	2.22	1140.26	0.69	1138.51	0.53	1.53
r204	1173.98	1119.04	3.10	1181.11	0.61	1175.79	0.15	2.39
r205	1647.74	1600.45	2.35	1653.08	0.32	1649.41	0.10	1.76
r206	1044.16	1015.14	2.89	1049.63	0.52	1045.11	0.09	2.23
r207	2073.95	1896.63	3.23	2091.21	0.83	2087.71	0.66	2.5
r208	1583.65	1575.89	3.86	1605.75	1.40	1593.48	0.62	3.05
r209	1966.04	1953.65	3.84	2012.38	2.36	1973	0.35	2.99
r210	1764.44	1665.10	3.62	1773.08	0.49	1769.74	0.30	2.76
r211	1626.82	1620.31	4.80	1658.46	1.94	1647.19	1.25	4.02
Avg R	***1250.44***	***1209.47***	***2.35***	***1266.25***	***1.34***	***1257.98***	***0.66***	***1.85***
rc101	711.38	700.35	1.22	724.85	1.89	714.09	0.38	0.95
rc102	932.80	926.08	1.38	947.27	1.55	939.18	0.68	1.02
rc103	945.25	897.98	1.27	948.60	0.35	947.01	0.19	1.02
rc104	775.43	767.06	0.76	780.15	0.61	780.01	0.59	0.61
rc105	958.00	949.19	1.02	978.25	2.11	963.05	0.53	0.81
rc106	691.98	687.14	1.06	693.51	0.22	693.09	0.16	0.80
rc107	1337.74	1268.58	1.87	1346.10	0.62	1339.62	0.14	1.51
rc108	959.95	949.97	2.26	967.87	0.83	960.45	0.05	1.65

rc201	1533.50	1498.08	2.05	1567.37	2.21	1544.89	0.74	1.67
rc202	1163.90	1161.22	1.99	1171.37	0.64	1169.18	0.45	1.53
rc203	1073.61	1054.18	2.67	1103.09	2.75	1100.03	2.46	2.09
rc204	1531.25	1521.30	1.87	1572.36	2.68	1538.1	0.45	1.56
rc205	1456.04	1442.35	2.55	1489.17	2.28	1465.79	0.67	2.08
rc206	1450.69	1437.05	2.64	1467.28	1.14	1452.33	0.11	2.13
rc207	1358.46	1331.02	2.23	1370.56	0.89	1361.16	0.20	1.89
rc208	1658.43	1633.56	2.55	1668.73	0.62	1665.01	0.40	2.17
Avg RC	***1158.65***	***1139.07***	***1.84***	***1174.78***	***1.34***	***1164.56***	***0.51***	***1.47***
Avg	***1209.62***	***1182.80***	***2.02***	***1231.79***	***1.85***	***1221.81***	***1.04***	***1.60***

We observe that the use of drones contributes to limited total revenue, with a positive average gap of 1.85%. Our results suggest that the use of drones is particularly helpful when customers are clustered (i.e., type C) where all average gaps are positives with an average gap equal to 2.88% that range between 0.66% and 4.21%. This is reasonable as drones can serve customers who are geographically closed to each other. Thus, using drones can significatively improve the quality solution and increase the total expected revenue compared to use only trucks to serve the customers.

7.3. Single compartment versus multiple compartments

Table 13 shows the benefit of having a drone to deliver multiple packages (i.e., "Multi compartments") per trip. The results of the comparison are reported in Table 13. The column "Single compartment" refers to the case of using the drone with single compartment payload in tandem with the trucks. Other columns report the same measures as in Tables 11 and 12.

Table 13
Benefit of multi compartment

Inst	Multi compartment			Single compartment				
	Best	Avg	CPU (min)	Best	Best%	Avg	Avg%	CPU (min)
c101	887.90	879.23	1.24	886.69	-0.14	876.13	-1.33	886.69
c102	1195.99	1182.30	1.36	1184.04	-1.00	1158.19	-3.16	1184.04
c103	1109.86	1093.55	1.23	1041.71	-6.14	968.65	-12.72	1041.71
c104	1253.90	1180.38	0.78	1172.25	-6.51	1088.81	-13.17	1172.25
c105	825.55	812.20	1.04	769.79	-6.75	711.99	-13.76	769.79
c106	556.35	551.73	1.04	538.11	-3.28	513.22	-7.75	538.11
c107	1230.75	1176.34	2.11	1150.09	-6.55	1065.14	-13.46	1150.09
c108	1122.84	1087.83	2.07	1069.09	-4.79	1003.28	-10.65	1069.09
c109	1281.53	1266.11	2.05	1192.55	-6.94	1087.07	-15.17	1192.55
c201	1089.77	1058.43	2.06	1054.99	-3.19	1014.02	-6.95	1054.99
c202	1275.96	1223.24	2.84	1252.67	-1.83	1222.59	-4.18	1252.67
c203	1594.00	1583.19	1.94	1480.82	-7.10	1357.93	-14.81	1480.82
c204	1330.49	1320.20	3.06	1301.50	-2.18	1259.48	-5.34	1301.50
c205	1314.21	1294.86	2.57	1235.30	-6.00	1152.11	-12.33	1235.30
c206	1933.53	1921.67	2.42	1794.02	-7.22	1645.91	-14.88	1794.02
c207	1660.66	1648.03	3.18	1651.98	-0.52	1633.15	-1.66	1651.98
c208	1594.90	1567.81	2.67	1493.48	-6.36	1384.53	-13.19	1493.48
Avg C	***1250.48***	***1226.30***	***1.98***	***1192.30***	***-4.50***	***1126.01***	***-9.68***	***1192.30***
r101	899.08	895.21	0.82	889.59	-1.06	868.88	-3.36	889.59
r102	910.65	899.30	0.90	895.84	-1.63	873.34	-4.10	895.84
r103	778.50	765.14	1.44	763.11	-1.98	737.41	-5.28	763.11
r104	877.41	864.94	0.89	864.93	-1.42	845.23	-3.67	864.93
r105	747.90	738.11	1.49	737.82	-1.35	721.54	-3.52	737.82
r106	1449.29	1426.42	0.88	1416.36	-2.27	1370.17	-5.46	1416.36
r107	1391.46	1377.53	2.10	1388.68	-0.20	1370.54	-1.50	1388.68
r108	753.70	749.18	2.01	750.60	-0.41	736.59	-2.27	750.60
r109	1238.25	1234.48	1.86	1235.45	-0.23	1216.40	-1.76	1235.45

r110	1347.98	1337.17	2.25	1339.60	-0.62	1323.24	-1.84	1339.60
r111	920.74	906.72	2.41	908.84	-1.29	886.03	-3.77	908.84
r112	1213.54	1205.33	1.95	1200.36	-1.09	1180.81	-2.70	1200.36
r201	1209.47	1199.25	3.67	1191.71	-1.47	1164.52	-3.72	1191.71
r202	1092.47	1073.46	2.99	1085.28	-0.66	1063.74	-2.63	1085.28
r203	1154.86	1144.30	2.10	1147.86	-0.61	1127.45	-2.37	1147.86
r204	1148.71	1137.15	3.27	1135.38	-1.16	1102.59	-4.01	1135.38
r205	1642.81	1624.31	2.26	1626.20	-1.01	1596.17	-2.84	1626.20
r206	1032.19	1005.27	2.88	1026.36	-0.57	1013.81	-1.78	1026.36
r207	2027.92	1993.21	3.38	1988.11	-1.96	1931.07	-4.78	1988.11
r208	1631.45	1592.06	4.14	1601.41	-1.84	1563.58	-4.16	1601.41
r209	2033.34	2010.91	4.07	2011.18	-1.09	1973.95	-2.92	2011.18
r210	1734.09	1631.64	3.77	1722.77	-0.65	1690.42	-2.52	1722.77
r211	1648.75	1637.54	5.21	1623.83	-1.51	1588.76	-3.64	1623.83
Avg R	***1255.85***	***1236.90***	***2.47***	***1241.36***	***-1.13***	***1215.05***	***-3.24***	***1241.36***
rc101	718.49	707.82	1.28	713.01	-0.76	702.97	-2.16	713.01
rc102	942.41	927.25	1.41	917.96	-2.59	884.78	-6.12	917.96
rc103	906.71	896.03	1.30	887.98	-2.07	861.73	-4.96	887.98
rc104	775.43	762.91	0.78	762.25	-1.70	742.42	-4.26	762.25
rc105	974.47	959.33	1.05	937.52	-3.79	894.18	-8.24	937.52
rc106	691.98	673.80	1.05	670.95	-3.04	642.99	-7.08	670.95
rc107	1279.52	1240.29	1.93	1238.15	-3.23	1186.94	-7.24	1238.15
rc108	959.95	926.84	2.26	938.09	-2.28	903.00	-5.93	938.09
rc201	1558.91	1536.55	2.20	1542.06	-1.08	1510.96	-3.08	1542.06
rc202	1161.81	1132.85	2.03	1138.72	-1.99	1110.31	-4.43	1138.72
rc203	1101.82	1068.76	2.74	1072.13	-2.69	1027.60	-6.74	1072.13
rc204	1565.70	1531.58	1.99	1556.22	-0.61	1526.51	-2.50	1556.22
rc205	1482.88	1459.60	2.80	1452.99	-2.02	1413.90	-4.65	1452.99
rc206	1462.39	1438.46	2.75	1444.95	-1.19	1403.46	-4.03	1444.95
rc207	1349.42	1339.64	2.55	1303.03	-3.44	1236.32	-8.38	1303.03
rc208	1644.62	1626.12	2.83	1594.57	-3.04	1531.79	-6.86	1594.57
Avg RC	***1161.03***	***1139.24***	***1.93***	***1135.66***	***-2.22***	***1098.74***	***-5.42***	***1135.66***
Avg	***1222.45***	***1200.81***	***2.13***	***1189.77***	***-2.62***	***1146.60***	***-6.11***	***1189.77***

The results show that having multi-compartment drones can improve the total revenue as compared to having only single-compartment drones. This is reasonable as multiple packages can be delivered on the same trip such that the flying distance can be shortened. In fact, allowing the same drone to deliver multiple packages to many customers leads an increase in the total revenue by 2.62% on average. From the results, using drones with multiple compartments is important and highly beneficial when the drones are allowed to deliver packages within a cluster (data set C). We observe an average gap of 4.50%. On the other hand, the gaps are less significant when customer locations are randomly generated locations (i.e., data sets R and RC). We believe that this is due to the need to travel longer distances to satisfy the time windows constraints. As a result, trucks need to deliver these small packages instead of using drones. In summary, the variant of having multi-compartment drones has proven to be beneficial for both policies. Hence, practitioners are suggested to consider using multi-compartment drones, especially when customer demands are clustered.

8. Conclusions

In this paper, we introduced a new variant of the Vehicle Routing Problem with Drones (VRP-D), in which a fleet of trucks-drones is used to serve customers where each drone is equipped with multiple compartments. We considered the flexibility of having drones to return to any trucks to swap depleted

batteries and also to pick up packages. The problem is denoted as the Vehicle Routing Problem with Drones equipped with Multiple payload Compartment (VRP-D-MC). In addition, our model considers the current payload in order to calculate the energy consumption for the drones.

We designed an Adaptive Multi-Start Simulating Annealing (AMS-SA) that incorporates an efficient constructive heuristic and several novel diversification and intensification mechanisms. Extensive numerical experiments showed that our proposed AMS-SA provides high-quality solutions both on our generated instances and on benchmark instances from the literature. In particular, our AMS-SA outperforms on average the current state-of-the-art algorithm, including the Hybrid Genetic Algorithm of Euchi and Sadok (2020). We also showed the effects of newly added features: (i) the multi-payload compartment, (ii) the flexibility of having drones to return to any truck, and (iii) the benefits of using trucks with drones. All these new features allowed furthering improving the total revenue. Moreover, we provided the impact of using different configuration drones on the quality solution and also, and we conducted a sensitivity analysis to show the effect of using the time windows.

Further developments on this topic may consider, instead of assuming that drones are fully recharged (swapping battery), a more realistic and complex scenario with a partial recharging setting. Another interesting issue to be analysed is to consider a dynamic environment where requests dynamically appear over the day, or a stochastic environment where travel times for trucks are affected by uncertainty due to traffic congestion.

Acknowledgement

The research of the fourth author was partially supported by the Research Grants Council (RGC) (General Research Fund 17200820), the Germany/Hong Kong Joint Research Scheme Germany Academic Exchange Service (DAAD) and RGC of Hong Kong (G-HKU703/19), and the 2019 Guangdong Special Support Talent Program – Innovation and Entrepreneurship Leading Team (China) (2019BT02S593).

References

Amazon, 2016. Amazon Prime Air. URL: https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011.

Alinaghian, M., & Shokouhi, N. (2018). Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. *Omega*, *76*, 85-99.

Bezos, J., 2013. Interview with charlie rose. Broadcasted on: 60 Minutes, CBS. December 1, Transcript: https://www.cbsnews.com/news/amazons- jeff- bezos- looks- to- the- future/.

Cheng, C., Adulyasak, Y., & Rousseau, L. M. (2020). Drone routing with energy function: Formulation and exact algorithm. *Transportation Research Part B: Methodological*, *139*, 364-387.

Chung, S. H., Sah, B., & Lee, J. (2020). Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Computers & Operations Research*, *123*, 105004.

Chiang, W. C., Li, Y., Shang, J., & Urban, T. L. (2019). Impact of drone delivery on sustainability and cost: Realizing the UAV potential through vehicle routing optimization. *Applied energy*, *242*, 1164-1175.

Chung, S. H., Sah, B., & Lee, J. (2020). Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Computers & Operations Research*, 105004.

D'Andrea, R. (2014). Guest editorial can drones deliver?. *IEEE Transactions on Automation Science and Engineering*, *11*(3), 647-648.
Demir, E., Bektaş, T., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, *223*(2), 346-359.
Pugliese, L. D. P., & Guerriero, F. (2017, September). Last-mile deliveries by using drones and classical vehicles. In *International Conference on Optimization and Decision Science* (pp. 557-565). Springer, Cham.
Pugliese, L. D. P., Guerriero, F., & Macrina, G. (2020). Using drones for parcels delivery process. *Procedia Manufacturing*, *42*, 488-497.
Dorling, K., Heinrichs, J., Messier, G. G., & Magierowski, S. (2016). Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*(1), 70-85.
Euchi, J., & Sadok, A. (2021). Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones. *Physical Communication*, *44*, 101236.
FFA (2020). Federal Aviation Administration. Available on : https://www.faa.gov/news/updates/?newsId=97907
Figliozzi, M. A. (2017). Lifecycle modeling and assessment of unmanned aerial vehicles (Drones) CO2e emissions. *Transportation Research Part D: Transport and Environment*, *57*, 251-261.
Fosin, J., Carić, T., & Ivanjko, E. (2014). Vehicle routing optimization using multiple local search improvements. *automatika*, *55*(2), 124-132.
Gonzalez-R, P. L., Canca, D., Andrade-Pineda, J. L., Calle, M., & Leon-Blanco, J. M. (2020). Truck-drone team logistics: A heuristic approach to multi-drop route planning. *Transportation Research Part C: Emerging Technologies*, *114*, 657-680.
Ham, A. M. (2018). Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C: Emerging Technologies*, *91*, 1-14.
Han, Y. Q., Li, J. Q., Liu, Z., Liu, C., & Tian, J. (2020). Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones. *International Journal of Advanced Robotic Systems*, *17*(2), 1729881420920031.
Hemmelmayr, V. C., Doerner, K. F., & Hartl, R. F. (2009). A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, *195*(3), 791-802.
Henke, T., Speranza, M. G., & Wäscher, G. (2015). The multi-compartment vehicle routing problem with flexible compartment sizes. *European Journal of Operational Research*, *246*(3), 730-743.
Karagul, K., Sahin, Y., Aydemir, E., & Oral, A. (2019). A simulated annealing algorithm based solution method for a green vehicle routing problem with fuel consumption. In *Lean and green supply chain management* (pp. 161-187). Springer, Cham.
Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, *34*(5-6), 975-986.
Kirschstein, T. (2020). Comparison of energy demands of drone-based and ground-based parcel delivery services. *Transportation Research Part D: Transport and Environment*, *78*, 102209.
Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J. M., & Brunese, P. A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering*, *129*, 14-30.
Kitjacharoenchai, P., Min, B. C., & Lee, S. (2020). Two echelon vehicle routing problem with drones in last mile delivery. *International Journal of Production Economics*, *225*, 107598.
Leishman, G. J. (2006). *Principles of helicopter aerodynamics with CD extra*. Cambridge university press.
Lin, S. W., & Vincent, F. Y. (2012). A simulated annealing heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, *217*(1), 94-107.
Liu, Y., Liu, Z., Shi, J., Wu, G., & Pedrycz, W. (2020). Two-echelon routing problem for parcel delivery by cooperated truck and drone. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.
Luo, Z., Poon, M., Zhang, Z., Liu, Z., & Lim, A. (2021). The Multi-visit Traveling Salesman Problem with Multi-Drones. *Transportation Research Part C: Emerging Technologies*, *128*, 103172.
Macrina, G., Pugliese, L. D. P., Guerriero, F., & Laporte, G. (2020). Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, *120*, 102762.
Masmoudi, M. A., Hosny, M., Braekers, K., & Dammak, A. (2016). Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem. *Transportation Research Part E: Logistics and Transportation Review*, *96*, 60-80.
Masmoudi, M. A., Hosny, M., Demir, E., Genikomsakis, K. N., & Cheikhrouhou, N. (2018). The dial-a-ride problem with electric vehicles and battery swapping stations. *Transportation research part E: logistics and transportation review*, *118*, 392-420.
Masmoudi, M. A., Hosny, M., Demir, E., & Pesch, E. (2020). Hybrid adaptive large neighborhood search algorithm for the mixed fleet heterogeneous dial-a-ride problem. *Journal of Heuristics*, *26*(1), 83-118.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, *21*(6), 1087-1092.

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.

Poikonen, S., & Campbell, J. F. (2021). Future directions in drone routing research. *Networks*, *77*(1), 116-126.

Poikonen, S., & Golden, B. (2020). Multi-visit drone routing problem. *Computers & Operations Research*, *113*, 104802.

Poikonen, S., Wang, X., & Golden, B. (2017). The vehicle routing problem with drones: Extended models and connections. *Networks*, *70*(1), 34-43.

Potvin, J. Y., Rousseau, J. M., 1995. An exchange heuristic for routing problems with time windows. *Journal of the Operational Research Society*, **46**(12), 1433-1446.

Pugliese, L. D. P., Guerriero, F., & Macrina, G. (2020). Using drones for parcels delivery process. *Procedia Manufacturing*, *42*, 488-497.

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.

Salama, M., & Srinivas, S. (2020). Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. *Transportation Research Part C: Emerging Technologies*, *114*, 620-642.

Savelsbergh, M. W. (1992). The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, *4*(2), 146-154.

Schermer, D., Moeini, M., & Wendt, O. (2019). A matheuristic for the vehicle routing problem with drones and its variants. *Transportation Research Part C: Emerging Technologies*, *106*, 166-204.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, *35*(2), 254-265.

Stolaroff, J. K., Samaras, C., O'Neill, E. R., Lubers, A., Mitchell, A. S., & Ceperley, D. (2018). Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nature communications*, *9*(1), 1-13.

Tamke, F., & Buscher, U. (2021). A branch-and-cut algorithm for the vehicle routing problem with drones. *Transportation Research Part B: Methodological*, *144*, 174-203.

Tiwari, M. K., Kumar, S., & Shankar, R. (2006). Solving part-type selection and operation allocation problems in an FMS: An approach using constraints-based fast simulated annealing algorithm. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *36*(6), 1170-1184.

Troudi, A., Addouche, S. A., Dellagi, S., and Mhamedi, A., 2018. Sizing of the drone delivery fleet considering energy autonomy. *Sustainability*, *10*(9), 3344.

Wang, X., Poikonen, S., & Golden, B. (2017). The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, *11*(4), 679-697.

Wang, Z., & Sheu, J. B. (2019). Vehicle routing problem with drones. *Transportation research part B: methodological*, *122*, 350-364.

Wei, L., Zhang, Z., Zhang, D., & Leung, S. C. (2018). A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, *265*(3), 843-859.

Xiang, Z., Chu, C., & Chen, H. (2006). A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research*, *174*(2), 1117-1139.

Xiao, Y., & Konak, A. (2015). A simulating annealing algorithm to solve the green vehicle routing & scheduling problem with hierarchical objectives and weighted tardiness. *Applied Soft Computing*, *34*, 372-388.

Yu. F. V. & Lin, S. W. (2014). Multi-start simulated annealing heuristic for the location routing problem with simultaneous pickup and delivery. *Applied soft computing*, *24*, 284-290.

Yu, V. F., Purwanti, S. S., Redi, A. P., Lu, C. C., Suprayogi, S., & Jewpanya, P. (2018). Simulated annealing heuristic for the general share-a-ride problem. *Engineering Optimization*, *50*(7), 1178-1197.

Yurek, E. E., & Ozmutlu, H. C. (2018). A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, *91*, 249-262.

Zhang, J., Campbell, J. F., Sweeney II, D. C., & Hupman, A. C. (2021). Energy consumption models for delivery drones: A comparison and assessment. *Transportation Research Part D: Transport and Environment*, *90*, 102668.

Žulj, I., Kramer, S., & Schneider, M. (2018). A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem. *European Journal of Operational Research*, *264*(2), 653-664.