Expert Systems With Applications 191 (2022) 116264

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Vehicle routing problem with drones considering time windows

R.J. Kuo[a], Shih-Hao Lu[b], Pei-Yu Lai[a], Setyo Tri Windras Mara[c,*]

[a] Department of Industrial Management, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 106, Taiwan
[b] Department of Business Administration, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 106, Taiwan
[c] Department of Mechanical and Industrial Engineering, Faculty of Engineering, Universitas Gadjah Mada, Special Region of Yogyakarta 55284, Indonesia

ARTICLE INFO

Keywords:
Vehicle routing problem with drones
Time windows
Unmanned aerial vehicles
Logistics
Variable neighborhood search

ABSTRACT

The cooperation of trucks and unmanned aerial vehicles (UAV) has become a new delivery method in the area of logistics and transportation. In this form of cooperation, the trucks are not only able to provide services to the customers, but also serve as a 'launch pad' for the drones, in which the drones can be launched to service a customer and then recovered at the rendezvous node. This study intends to explore this cooperation by developing a model for the vehicle routing problem with drones that considers the presence of customer time windows (VRPTWD). A mixed-integer programming (MIP) model is presented to minimize the total travelling costs. Then, a simple yet effective variable neighborhood search (VNS) procedure with a novel solution representation is proposed as a solver. The numerical results indicate the ability of the proposed VNS to solve the VRPTWD, as well as the improvement of delivery performance using drones.

1. Introduction

The unmanned aerial vehicles (UAV) or drones represent aircraft without a pilot that automatically controls the balance of their flights through an onboard computer system. In recent years, drones have received huge attention from both logistics practitioners and researchers due to their potential implementation in logistics. Apart from the ability to be operated without a human pilot, the idea of deploying drones for delivery services stems from the fact that drones are generally faster than trucks, have lower transportation costs, and can avoid traffic congestions in traditional road networks (Wohlsen, 2014).

Today, several prominent companies have started to launch their remarkable pilot-projects for implementing drones in logistics. For instances, Amazon has launched their "Amazon's Prime Air UAV" program to directly deliver the customer packages from their warehouse (Rose, 2013). The third-party logistics (3PL) company, DHL, created a "Parcelcopter" program to deliver medical supplies to a car-free island in Germany (Bryan, 2014). Another 3PL company, UPS, has also become a part of this movement by creating a blood delivery program in Rwanda with UAV manufacturer Zipline (Tilley, 2016). In short, it can be observed that in the near future, drones may play an important role in the logistics industry.

However, drones still possess some limitations before being fully implemented in logistics. From the operational perspective, some challenges arise from the fact that the drones have limited flight distance and limited payload capacity of packages (Murray and Chu, 2015). The drones deployed in "Amazon's Prime Air UAV," for instances, have a flying range of 10 miles and 5-pound payload capacity, as reported by Gross (2013).

These limitations have led to several remarkable recent studies that promoted the implementation of trucks and drones in a tandem way. In this form of cooperation, the trucks are not only able to provide delivery services to the customers, but also serve as a 'launch pad' for the drones as well, in which the drones can be launched to service a customer and then be recovered at the rendezvous node (Murray and Chu, 2015). To this end, some industrial implementations have proven the applicability of truck-drone cooperation (Peterson and Dektas, 2017; Etherington, 2017). The applicability of this cooperation stems from the exploitation of the fact that trucks generally have a longer travel range and can carry much more payload than drones (Ha et al., 2018). Thus, they can be operated as a 'mobile operation center' to increase the proximity of drones. Murray and Chu (2015) first introduced the model of this cooperation as the flying sidekick traveling salesman problem (FSTSP), while Agatz et al. (2018) studied a similar form of cooperation with a model of traveling salesman problem with drone (TSP-D). Their studies have shown that even the cooperation of one truck and one drone can

* Corresponding author.
E-mail addresses: rjkuo@mail.ntust.edu.tw (R.J. Kuo), shlu@mail.ntust.edu.tw (S.-H. Lu), m10701009@gapps.ntust.edu.tw (P.-Y. Lai), setyotriw@gmail.com (S.T.W. Mara).

https://doi.org/10.1016/j.eswa.2021.116264
Received 25 August 2021; Received in revised form 30 October 2021; Accepted 19 November 2021
Available online 27 November 2021
0957-4174/

improve the service quality and reduce the completion time of delivery services. These promising findings then result in some studies on the scalability of truck-drone cooperation and some recent works have extended the FSTSP and TSP-D models into the vehicle routing problem with drones (VRPD) (e.g. Wang et al., 2017; Poikonen et al., 2017; Wang and Sheu, 2019; Sacramento et al., 2019), in which a fleet of trucks are equipped with one or more drones to deliver packages to customers.

This study aims to extend the exploration of truck-drone cooperation by considering the constraint of time windows. In this form of extension, each customer is subjected to a time window constraint, and these customers must be visited by either a truck or a drone within their specified time windows $[o_i, e_i]$, where o_i denotes the opening time of a given node i and e_i denotes the closing time of node i. A hard time windows constraint is imposed in this study. This concept can be visually explained as in Fig. 1, which presents three different scenarios: (i) the vehicle may arrive at i before the start of the time window, but the services will not be started until the opening time of the time window and waiting time is incurred, (ii) if the vehicle arrives at i in between o_i and e_i, the service starts immediately, and (iii) if the vehicle arrives after e_i, the corresponding solution is violating the time window constraints. Furthermore, the approach of hard time windows has been widely implemented by various previous studies on routing problems with time windows, such as Braaten et al. (2017), Chen et al. (2018), and Vahdani et al. (2018).

The presence of time windows is a classical extension of routing problems, which arises from the observation that the customers in numerous industries often require the delivery service to be performed within a certain period (Solomon, 1984; Desaulniers et al., 2014; Zhen et al., 2020). Additionally, addressing time windows in a logistics optimization problem increases the complexity of the considered problem, due to the presence of additional constraints to impose the earliest and/or latest time a service can be executed. To the best of our knowledge, few studies have discussed the issue of time windows in drone routing problems. A recent study from Lan (2020) proposed a model for the traveling salesman problem with time windows and drones (TSPTWD), which considers the presence of time windows in a cooperation of a single truck-drone tandem. On the other hand, we are only aware of two studies from Pugliese and Guerriero (2017) and Pugliese et al. (2020) that discussed the extension of TSPTWD to consider the cooperation of multiple trucks and drones in a vehicle routing problem with drones and time windows (VRPTWD) model. Nevertheless, both studies did not consider presenting an approach to solve the problem in

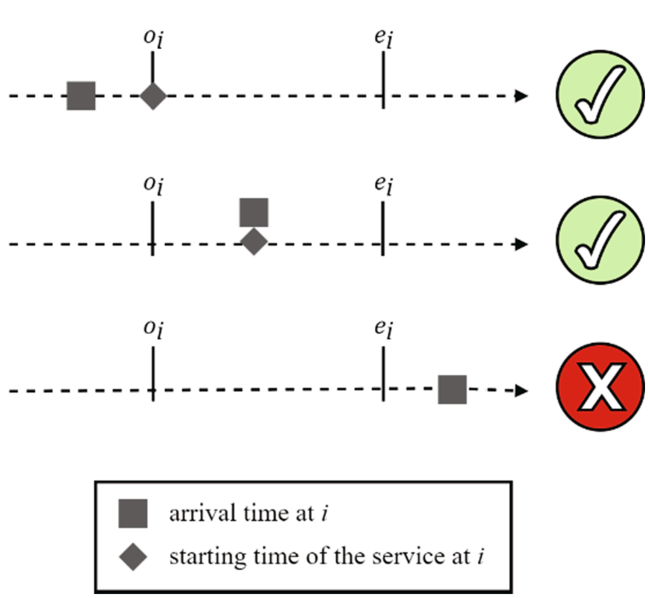

Fig. 1. Three different scenarios in hard time windows.

large-scale instances, which is the main contribution of this present study.

The objectives of this study are twofold:

1. *Formulating a mathematical model for VRPTWD*
   In this study, the mathematical formulation of VRPD from Sacramento et al. (2019) is extended and our effort results in a mixed-integer programming (MIP) formulation for the VRPTWD, with an objective function to minimize the total travel costs.
2. *Developing an effective solution approach for the VRPTWD*
   As the solution approach for the VRPTWD model, the current study proposes to use a variable neighborhood search (VNS) (Mladenović and Hansen, 1997) with novel solution representation for solving large-sized problems. The main goal of our VNS proposal is to develop a simple but effective solution approach for the VRPTWD, which itself is a complex combinatorial optimization problem. Afterwards, numerical analysis is done based on the results of our computational tests.

The rest of this study is organized as follows. Section 2 provides the literature review of the related works. Section 3 presents the proposed VRPTWD model and Section 4 described the proposed VNS to solve it. The computational results and analysis are discussed in Section 5. Finally, the concluding statements are made in Section 6.

2. Literature review

In this section, the related literature are reviewed to assess the contributions of this study. The main focuses of this review are the literature on drone routing as well as its solution approaches. The discussion starts with a review on routing problems with a cooperation of single truck-drone tandem (Section 2.1). Further discussion in Section 2.2 captures the articles on the cooperation of multiple trucks and multiple drones. Then, a review on the proposed VNS is given on Section 2.3, while Section 2.4 summarizes the contributions of this study.

2.1. Routing problems with a cooperation of single truck and drones

In recent years, there has been an increasing amount of attention on drone routing research, especially from the logistics and operational researcher community. A handful of dedicated surveys (Barmpounakis et al., 2016; Otto et al., 2018; Coutinho et al., 2018; Khoufi et al., 2019; Chung et al., 2020; Rojas Viloria et al., 2021; Macrina et al., 2020; Poikonen and Campbell, 2021) have been conducted in the last five years to review the related studies in logistics with drones, either from the application perspective (e.g. Barmpounakis et al., 2016; Otto et al., 2018) or from the modelling perspective (e.g. Chung et al., 2020; Rojas Viloria et al., 2021; Macrina et al., 2020). In this regard, Chung et al. (2020) and Macrina et al. (2020) provided an interesting perspective to classify the works in drone routing based on the form of vehicles that perform the delivery services: (1) using drones only and (2) using a cooperation of trucks and drones.

Here, this study focuses on the class of truck-drone combined operations. The research on this area gained momentum from the seminal work of Murray and Chu (2015) who introduced a MIP formulation for FSTSP, along with the parallel drone scheduling traveling salesman problem (PDSTSP). The FSTSP represents the case where a single truck is equipped with a single drone, in which the drone can perform a delivery operation called *sortie* and then return to the truck in a customer or depot node. Agatz et al. (2018) then proposed an integer programming formulation for a similar condition in their TSP-D model which allows the launching node to be similar to the rendezvous node. They showed that the TSP-D model is more efficient than the FSTSP from Murray and Chu (2015) and then proposed to use a 'truck first, drone second' heuristic approach to solve large-sized problem instances.

Since these two influential works, several studies have attempted to broaden the perspective on the coordination of trucks and drones from the operational perspective. For instance, Ha et al. (2018) extended the TSP-D model by considering a different objective function to minimize the logistics cost, instead of the completion time as in the FSTSP and TSP-D. Dell'Amico et al. (2021) proposed a new formulation for the FSTSP along with a set of valid inequalities. By using an exact branch-and-cut algorithm, they showed that their new formulation is superior to the original formulation of FSTSP. Bouman et al. (2018) proposed a dynamic programming approach for the TSP-D. They found that this approach can solve TSP-D instances with up to 20 customer nodes. Es Yurek and Ozmutlu (2018) proposed a solution approach for the TSP-D based on an iterative decomposition approach that is able to yield a high-quality solution for the TSP-D with more efficient computational time, while Poikonen et al. (2019) presented an exact branch-and-bound approach for the TSP-D. Marinelli et al. (2018) discussed the possibility to relax the basic assumption of FSTSP and TSP-D that the drone can only be launched and collected again in a node. This idea then resulted in the so-called *en route* operation. Further, Murray and Raj (2020) recently introduced another extension of the FSTSP by proposing a mathematical formulation of multiple flying sidekick traveling salesman problem (m-FSTSP), in which a single truck can operate simultaneously with multiple drones. Meanwhile, Gonzalez-R et al. (2020), Poikonen and Golden (2020), and Agárdi et al. (2019) proposed to consider a multi-drop condition that allows a drone to serve more than one customer node within a single *sortie* operation.

2.2. Routing problems with a cooperation of multiple trucks and multiple drones

While the aforementioned studies focus on the deployment of a single truck, some other studies discuss the scalability of truck-drone cooperation by considering the usage of multiple trucks. This form of cooperation is discussed in the form of VRPD model. Here, we review the articles related to VRPD and its extensions.

2.2.1. Vehicle routing problem with drones (VRPD)

The VRPD was first introduced by Wang et al. (2017). It considers a fleet of trucks, each is equipped with one or more drones to deliver the parcel directly or launch the drone to deliver the parcel. In their study, Wang et al. (2017) conducted a comprehensive analysis of the worst-case scenarios to propose bound on the time savings that can be achieved by the model. Since then, several published works extended the work of Wang et al. (2017). Poikonen et al. (2017) then continued the work performed by Wang et al. (2017) by considering the battery limitation of drones, the effect of two different distance matrices for trucks and drones, as well as taking the economic aspect as the objective function. Ulmer and Thomas (2018) proposed a dynamic VRPD in a same-day delivery setting with an objective function to maximize the expected number of customers served within a single working day. Schermer et al. (2018) presented two heuristics for solving the VRPD with two main stages: (1) initialization and (2) improvement phase. The initialization phase is conducted with *route first cluster second* heuristic and the solution is then improved with several local search moves. Wang and Sheu (2019) extended the VRPD to consider the multi-drop condition for drones and proposed an arc-based model for the VRPD with a branch-and-price algorithm to solve it. Meanwhile, Schermer et al. (2019a) presented a VRPD model with the possibility of *en route* operations (VRPDERO), similar to the idea of Marinelli et al. (2018) in the context of TSP-D. A hybrid variable neighborhood search with tabu search was proposed as the solution approach for the VRPDERO and the algorithm was equipped with a drone insertion operator embedded within a divide and conquer approach.

Several works also proposed their version of mathematical formulation for VRPD. Chiang et al. (2019) discussed the environmental impact of implementing drones in logistics by presenting a VRPD model. The model was presented with two different objectives: (1) to minimize the total cost and (2) to minimize the CO_2 emissions incurred from the logistics system. Their work resulted in an important finding that the usage of drones may lead to a cost-efficient and environmental-friendly logistics system. Popović et al. (2019) proposed a mixed-integer quadratic programming approach to model the VRPD based on the three-index formulation for vehicle routing problem. Their proposed model has an objective function to minimize the distribution cost, which was calculated based on the travelled distance and working time of vehicles. Schermer et al. (2019b) developed the MIP formulation of the VRPD model and proposed a matheuristic algorithm for several classes of the VRPD. They decomposed the model into an allocation and sequencing component and proposed to use the savings heuristic to solve the problem heuristically. Kitjacharoenchai et al. (2019) and Sacramento et al. (2019) investigated the development of VRPD formulation based on the formulation of FSTSP from Murray and Chu (2015). Kitjacharoenchai et al. (2019) proposed a model called as multiple traveling salesman problem with drones (MTSP-D). MTSP-D assumed that all trucks have unlimited capacity to carry multiple drones and packages, with a goal to minimize the delivery completion time. Then, Sacramento et al. (2019) developed a VRPD model with an objective function to minimize the operational cost under the restriction of a maximum duration for all routes. Different to the MTSP-D, Sacramento et al. (2019) considered the capacity of trucks and the maximum flight duration of drones. For the solution approach, they proposed an adaptive large neighborhood search (ALNS) with several destroy and repair operators.

2.2.2. VRPD with time windows (VRPTWD)

This study proposes a VRPTWD model by considering the presence of customer time windows within the VRPD model of Sacramento et al. (2019). In this regard, the VRPTWD can be viewed as an extension of the vehicle routing problem with time window (VRPTW) (Bräysy and Gendreau, 2005a; Bräysy and Gendreau, 2005b) and the traveling salesman problem with time windows and drones (TSPTWD) from Lan (2020) as well. Time windows arise naturally in the logistics system of numerous industries (Desaulniers et al., 2014; Zhen et al., 2020), and to date, few efforts have been dedicated to the issue of time windows in drone routing. The pioneering work in the consideration of time windows in the VRPD was Pugliese and Guerriero (2017). They developed the first VRPTWD model to minimize total transportation costs, in which the costs comprise the cost per unit of distance travelled by trucks and drones. In their model, Pugliese and Guerriero (2017) considered hard time windows for the customers and assumed the setup time for launching and recollecting the drones to be negligible. The proposed model was solved using the commercial software CPLEX 12.5.1 for two problem sizes with five and ten customers. Another study by Ham (2018) extended the PDSTSP with time windows constraints and the consideration of pickup-and-delivery demands. In the model proposed by Ham (2018), a fleet of *m*-trucks are accompanied by a set of *m*-drone that can perform delivery services from *m*-depots. The author developed the model using a constraint programming approach and proposed a variable ordering heuristics algorithm to improve the performance of their model. More recently, Pugliese et al. (2020) discussed the effectiveness of using drones for delivery systems in urban logistics environments. They performed a comparison analysis using three different models, namely the VRPTWD model of Pugliese and Guerriero (2017), the classical VRPTW with trucks only, and VRPTW with drones only. Their study highlighted that the VRPTWD has the best trade-off between efficiency and negative externalities of delivery tasks such as congestion and emission.

2.3. Solution method for VRPTWD

Moving on, in the current study, a metaheuristic VNS is applied as the solution approach for the VRPTWD model. VNS is a promising algorithm that was first discussed by Mladenovic and Hansen (1997).

Numerous documentations on the effectiveness of VNS-based algorithms in solving various complex optimization problems have also been published before. Within the context of routing problems, those documentations are ranging from a class of periodic routing problems (Hemmelmayr et al., 2009), team orienteering problem with time windows (Labadie et al., 2012), dial-a-ride problem (Parragh et al., 2010), production routing problem (Qiu et al., 2018), and vehicle routing problem with multiple time windows (Ferreira et al., 2018). Furthermore, several successful implementations of VNS-based algorithms for routing problems with drone are also available, such as de Freitas and Penna (2018, 2020) who presented an effective VNS procedures for the FSTSP. Overall, these facts show the applicability of VNS in this context and motivate us to develop a VNS-based approach for the proposed VRPTWD model.

In order to evaluate our proposed VNS in solving the VRPTWD, this study compares the performance of VNS to the performance of ALNS algorithm adapted from Sacramento et al. (2019). This approach to compare the proposed algorithm (VNS) to the relevant approach available in the literature (ALNS) is developed based on the ideal scenario in comparing metaheuristics discussed by Silberholz et al. (2019). This approach also has been employed by several recent studies, such as Ribeiro et al. (2014), Lee and Prabhu (2016), and Avci and Avci (2019). In this regard, the ALNS can be seen as a relevant benchmark due to the several reasons. First, the ALNS has been proven by Sacramento et al. (2019) to be effective to solve the problem most related to the VRPTWD, which is the VRPD. Thus, adapting ALNS is naturally the first practical option for dealing with VRPTWD. Second, ALNS has been widely known to be one of the most prominent algorithms for solving routing problems. Previous articles have deployed ALNS framework to develop a unified heuristic algorithm for a class of routing problems, such as vehicle routing problems (VRP) with backhauls (Ropke and Pisinger, 2006), capacitated VRP (Pisinger and Ropke, 2007), and periodic location-routing problems (Koç, 2016). Moreover, another work from Tu et al. (2018) has also proven the effectiveness of ALNS-based heuristic in solving another variant of routing problems with drones, namely the traveling salesman problem with multiple drones. Therefore, although we are aware of other metaheuristic approaches available for solving routing problems (see Prodhon and Prins, 2016; Elshaer and Awad, 2020), our decision to select the ALNS from Sacramento et al. (2019) as a benchmark approach can be justified appropriately.

2.4. Contributions of this study

Finally, the contributions of this study are threefold. First, this study proposes a mathematical formulation for the VRPTWD based on the VRPD model of Sacramento et al. (2019). This study aims to enhance the discussion on the presence of time windows in drone routing that seems to be neglected until now by the research community. Second, with respect to the study of Pugliese and Guerriero (2017), our proposed model differs from their model due to the presence of setup time consideration for launching and recollecting drones, while compared to the TSPTWD from Lan (2020), our model can be viewed as the extension of their MIP model from the presence of a multiple number of ground vehicles. Further, this model focuses on the minimization of the arrival time instead of the monetary aspect as in Pugliese and Guerriero (2017). Thus, it can be confirmed that our VRPTWD model is original. Third, a metaheuristic approach of VNS with a novel solution representation is proposed for the VRPTWD model. In this regard, to the best of our knowledge, this is the first heuristic solution developed for the VRPTWD and our numerical experiments in Section 4 confirms the applicability of VNS to solve instances with up to 50 customers.

3. VRPTWD problem formulation

This section provides a formal definition of VRPTWD and mathematical formulation for the VRPTWD. Our formulation can be seen as an extension of the MIP of the FSTSP presented by Murray and Chu (2015), TSPTWD of Lan (2020), as well as the MIP of the VRPD presented by Sacramento et al. (2019). This study takes the presence of customer time windows into account, along with the capacity of the truck when arranging the route. Let us first explain several important assumptions that hold as true in this study. First, the vehicle routes must start and end in the depot. Second, the routing distances associated with trucks and drones are calculated with Manhattan distance, since it is assumed that the drones may also find obstacles within its routes (e.g. the presence of tall buildings). Third, a drone cannot be launched and recollected at the same node. Fourth, within a single truck route, the drone can be launched multiple times and the recharging times are neglected. Fifth, during the drone delivery operation, the truck can also perform delivery services in other customer nodes.

By considering these assumptions, the VRPTWD can be defined in an undirected graph $G = (V,A)$. The set $V = N\cup D$ stands for the set of all nodes, where $N = \{1,\cdots,n\}$ is the set of n customer nodes that need to be served and $D = \{0,n+1\}$ denotes the depot nodes, while A is the set of all arcs between nodes. Given a v fleet of homogeneous trucks $U = \{1,\cdots,v\}$ with a load capacity Qt, each equipped with a single drone with a load capacity Qd, the main task of VRPTWD is to deliver the packages to the all customers in N, so that all customer demands q_i are satisfied. Each customer must be served in a duration s_i exactly once by either truck or drone. The services must be performed within the pre-defined hard time windows $[o_i,e_i]$, where o_i and e_i respectively correspond to the earliest and latest service time of customer $i\in N$. Further, in the cases where a vehicle arrives too early in node i, the vehicle must wait for the opening time before the service can be performed.

Then, let us also introduce $V_L = \{0,1,\cdots,n\}$ as the subset of launching nodes, $V_R = \{1,\cdots,n,n+1\}$ as the subset of rendezvous nodes, and $V_D\subseteqq N$ as the subset of customer nodes that can be served by a drone (if $q_i\leq Qd$). During the delivery operations, the drone $v\in U$ can be launched from its corresponding truck $v\in U$ in node $i\in V_L$ to serve one customer $j\in V_D$. Then, after performing a service in customer j, the drone must fly back to return to the truck in rendezvous node $k\in V_R$ before the flight endurance E is violated. For the purpose of operating the drones, setup times SL and SR are respectively incurred for the launching and recollecting tasks. Additionally, due to the presence of external factors such as congestion and the contour of road networks, the trucks and drones may require different travel times to traverse the same arc $(i,j)\in A$. These travel times are respectively denoted by t_{ij} and t'_{ij}.

The objective function of the VRPTWD model is to minimize the total travel cost of trucks and drones, which correspond to the fuel price incurred to operate them. The calculation of travel cost in this study is following the explanation of Sacramento et al. (2019). Given a distance matrix d_{ij} where $i,j\in A$, the cost to travel on arc (i,j) using truck (C_{ij}) and drones (C'_{ij}) can be respectively calculated using Eqs. (1) and (2),

$$C_{ij} = (d_{ij}\bullet MC)FP\bullet FC \tag{1}$$

$$C'_{ij} = \alpha\bullet C_{ij} \tag{2}$$

where MC stands for the miles converter, FP denotes the fuel price per litre, FC denotes the fuel consumption of truck, and α is the cost factor of drone.

Finally, Table 1 provide a summary of all sets and notations, while Table 2 presents six decision variables required to construct the MIP model for the VRPTWD. The full formulations are presented by Equations (3)-(33).

Objective function:

$$\text{Minimize}Z = \sum_{v\in U}\Big(\sum_{i\in V\backslash\{j\}}\sum_{j\in V\backslash\{i\}} C_{ij}x^{v}_{ij} + \sum_{i\in V_L\backslash\{j,k\}}\sum_{j\in N\backslash\{i,k\}}\sum_{k\in V_R\backslash\{i,j\}}(C'_{ij}+C'_{jk})y^{v}_{ijk}\Big) \tag{3}$$

Subject to:

Table 1
Summary of notations for mathematical model.

Symbol	Description
Sets and notations	
n	Number of customers to be served
v	Number of available truck-drone tandems
D	Depot node and its dummy duplicate, $D = \{0, n+1\}$
N	Set of all customer nodes, $N = \{1, \cdots, n\}$
V	Set of all nodes, $V = N \cup D = \{0, 1, \cdots, n, n+1\}$
U	Set of available truck-drone tandems, $U = \{1, \cdots, v\}$
A	Set of (i,j) arcs between all nodes, $i, j \in V, i \neq j$
V_L	Subset of launching nodes, $V_L = \{0, 1, \cdots, n\}$
V_R	Subset of rendezvous nodes, $V_R = \{1, \cdots, n, n+1\}$
V_D	Subset of customer nodes that can be served by a drone, $V_D \subseteqq N$
Parameters	
d_{ij}	Travel distance of a truck from nodes i to j
d'_{ij}	Travel distance of a drone from nodes i to j
t_{ij}	Travel time of a truck from nodes i to j
t'_{ij}	Travel time of a drone from nodes i to j
C_{ij}	Cost incurred to travel from nodes i to j using a truck
C'_{ij}	Cost incurred to travel from nodes i to j using a drone
Qt	The payload capacity of a truck
Qd	The payload capacity of a drone
q_i	Demand of customer i
s_i	Service time incurred to serve node i
$[o_i, e_i]$	Opening and closing time window of node i, respectively
SL	Setup time required to launch a drone in the launch node
SR	Setup time required to retrieve a drone in the rendezvous node
E	Flight endurance of drone
MC	Miles converter
FP	Fuel price
FC	Fuel consumption of truck
α	Cost factor of drone
$Tmax$	Maximum duration time of a tour, $Tmax = max_{i \in V}(l_i)$
M	A sufficiently large number, $M = \lvert V \rvert$
x^v_{ij}	A binary variable that takes the value of 1 if arc (i,j) is traversed by truck v, and 0 otherwise.
y^v_{ijk}	A binary variable that takes the value of 1 if drone v launches from node i, deliver a package to node j, and returns to the truck v at node k, 0 otherwise.
P^v_{ij}	A binary variable that takes the value of 1 if node i is served before, but not necessarily adjacent to, node j within the route of truck v, and 0 otherwise.
u^v_i	A non-negative integer variable that indicates the position of node i within the route of truck v.
a^v_i	A continuous variable to indicate the arrival time of truck v at node i
a'^v_i	A continuous variable to indicate the arrival time of drone v at node i

$$\sum_{v \in U} \sum_{i \in V_L \setminus \{j\}} x^v_{ij} + \sum_{v \in U} \sum_{i \in V_L \setminus \{j,k\}} \sum_{k \in V_R \setminus \{i,j\}} y^v_{ijk} = 1 \forall j \in N \tag{4}$$

$$\sum_{j \in N} x^v_{0j} \leq 1 \forall v \in U \tag{5}$$

$$\sum_{i \in N} x^v_{i,n+1} \leq 1 \forall v \in U \tag{6}$$

$$\sum_{i \in V_L \setminus \{j\}} x^v_{ij} - \sum_{k \in V_R \setminus \{j\}} x^v_{jk} = 0 \forall v \in U, j \in N \tag{7}$$

$$u^v_i - u^v_j + 1 \leq M(1 - x^v_{ij}) \forall v \in U, i \in V_L \setminus \{j\}, j \in V_R \setminus \{i\} \tag{8}$$

$$u^v_j \leq M \sum_{i \in V_L \setminus \{j\}} x^v_{ij} \forall v \in U, j \in V_R \tag{9}$$

$$u^v_j - u^v_i \leq MP^v_{ij} \forall v \in U, i \in V_L \setminus \{j\}, j \in N \setminus \{i\} \tag{10}$$

Table 2
Values of problem parameters.

Problem parameters	Value
n	$\{5, 10, 15\}$ for small-size instances, $\{20, 30, 40, 50\}$ for larger-size instances.
Size of grid area	$\{5 \times 5, 10 \times 10\}$
SL	1 min
SR	1 min
s_i	10 min
Truck speed	35 mph
Drone speed	50 mph
E	30 min
Qt	1400 kg
Qd	5 kg
MC	1.61 km/miles
FP	1.13 €/litre
FC	0.07 L/km
α	10% of the truck travel cost
$Tmax$	8 h

$$u^v_j - u^v_i \geq M(P^v_{ij} - 1) + 1 \forall v \in U, i \in V_L \setminus \{j\}, j \in N \setminus \{i\} \tag{11}$$

$$\sum_{j \in N} \left(\sum_{k \in V_R \setminus \{j\}} q_j x^v_{jk} + \sum_{i \in V_L \setminus \{j,k\}} \sum_{k \in V_R \setminus \{i,j\}} q_j y^v_{ijk} \right) \leq Q \forall v \in U \tag{12}$$

$$y^v_{ijk} = 0 \forall v \in U, i \in V_L \setminus \{j,k\}, j \in N \setminus \{V_D, i, k\}, k \in V_R \setminus \{i,j\} \tag{13}$$

$$\sum_{j \in N \setminus \{i,k\}} \sum_{k \in V_R \setminus \{i,j\}} y^v_{ijk} \leq 1 \forall v \in U, i \in V_L \tag{14}$$

$$\sum_{i \in V_L \setminus \{j,k\}} \sum_{j \in N \setminus \{i,k\}} y^v_{ijk} \leq 1 \forall v \in U, k \in V_R \tag{15}$$

$$2(y^v_{ijk}) \leq \sum_{j \in N \setminus \{i\}} x^v_{ij} + \sum_{j \in V_L \setminus \{k\}} x^v_{jk} \forall i \in V_L \setminus \{j,k\}, j \in N \setminus \{i,k\}, k \in V_R \setminus \{i,j\}, v \in U \tag{16}$$

$$y^v_{0jk} \leq \sum_{i \in V_L \setminus \{j,k\}} x^v_{ik} \forall j \in N \setminus \{k\}, k \in V_R \setminus \{j\}, v \in U \tag{17}$$

$$a^v_0 = 0 \forall v \in U \tag{18}$$

$$a'^v_0 = 0 \forall v \in U \tag{19}$$

$$a^v_i + t_{ik} + s_i + SL\left(\sum_{j \in N \setminus \{i,k\}} y^v_{ijk} \right) + SR\left(\sum_{j \in N \setminus \{i,k\}} y^v_{ijk} \right) \leq a^v_k + Tmax(1 - x^v_{ik}) \forall v \in U, i \in V_L \setminus \{k\}, k \in V_R \setminus \{i\} \tag{20}$$

$$a^v_i + t'_{ij} + SL - Tmax\left(1 - \sum_{k \in V_R \setminus \{i,j\}} y^v_{ijk}\right) \leq a'^v_j \forall v \in U, i \in V_L\{j\}, j \in N \setminus \{i\} \tag{21}$$

$$a'^v_j + t'_{jk} + s_j + SR - Tmax\left(1 - \sum_{i \in V_L \setminus \{j,k\}} y^v_{ijk}\right) \leq a'^v_k \forall v \in U, j \in N \setminus \{k\}, k \in V_R \setminus \{j\} \tag{22}$$

$$a'^v_i \geq a^v_i - Tmax\left(1 - \sum_{j \in N \setminus \{i,k\}} \sum_{k \in V_R \setminus \{i,j\}} y^v_{ijk}\right) \forall v \in U, i \in V_L \tag{23}$$

$$a'^v_i \leq a^v_i + Tmax\left(1 - \sum_{j \in N \setminus \{i,k\}} \sum_{k \in V_R \setminus \{i,j\}} y^v_{ijk}\right) \forall v \in U, i \in V_L \tag{24}$$

$$a'^{v}_{k} \geq a^{v}_{k} - Tmax(1 - \sum_{i \in V_L \backslash \{j,k\}} \sum_{j \in N \backslash \{i,k\}} y^{v}_{ijk}) \forall v \in U, k \in V_R \tag{25}$$

$$a'^{v}_{k} \leq a^{v}_{k} + Tmax(1 - \sum_{i \in V_L \backslash \{j,k\}} \sum_{j \in N \backslash \{i,k\}} y^{v}_{ijk}) \forall v \in U, k \in V_R \tag{26}$$

$$t'_{ij} + s_j + t'_{jk} - Tmax(1 - y^{v}_{ijk}) \leq E \forall v \in U, i \in V_L \backslash \{j,k\}, j \in N \backslash \{i,k\}, k \in V_R \backslash \{i,j\} \tag{27}$$

$$a'^{v}_{k} - Tmax(3 - \sum_{j \in N \backslash \{i,k\}} y^{v}_{ijk} - \sum_{l \in N \backslash \{b,m\}} \sum_{m \in V_R \backslash \{b,l\}} y^{v}_{blm} - P^{v}_{ib}) \leq a'^{v}_{b} \forall v \in U, i \in V_L \backslash \{b,k\}, k \in V_R \backslash \{i,b\}, b \in N \backslash \{i,k\} \tag{28}$$

$$o_i \leq a^{v}_{i} \leq e_i \forall v \in U, i \in V \tag{29}$$

$$o_i \leq a'^{v}_{i} \leq e_i \forall v \in U, i \in V \tag{30}$$

$$x^{v}_{ij} \in \{0,1\} \forall v \in U, i,j \in A \tag{31}$$

$$y^{v}_{ijk} \in \{0,1\} \forall v \in U, i \in V_L \{j,k\}, j \in N \backslash \{i,k\}, k \in V_R \backslash \{i,j\} \tag{32}$$

$$P^{v}_{ij} \in \{0,1\} \forall v \in U, i \in V_L \backslash \{j\}, j \in N \backslash \{i\} \tag{33}$$

$$u^{v}_{i}, a^{v}_{i}, a'^{v}_{i} \geq 0 \forall v \in U, i \in V \tag{34}$$

The objective function (3) aims to minimize total travel costs incurred. The calculation of travel costs involves the travel cost of trucks and the travel cost of drones. Constraint (4) guarantees that each customer is only visited once, either by a truck or a drone. This is imposed by set the sum of variables x^{v}_{ij} and y^{v}_{ijk} that visits a node $j \in N$ to be 1. Therefore, a single node will not be visited by a drone (truck) if it has been visited by a truck (drone). Constraint (5) defines that each truck must start from the depot, while Constraint (6) states that a route must also end at the depot. Both constraints also impose that each truck can only be used once. Constraint (7) is the flow conservation constraint for the trucks, which ensures that the inflow and outflow of a node $j \in N$ will always be equal. The elimination of subtours for the trucks is provided by Constraints (8) and (9), which ensure that the vehicles will not travel through the arcs that have been traversed. Constraint (10) and (11) define the sequence of vehicle tours to avoid a given customer node to appear multiple times in a single truck route. Constraint (12) is the capacity constraint for trucks, while the payload limitation of drones has been imposed by the creation of set V_D, so that the drones will only serve customer nodes belong to V_D as in Constraint (13). Constraints (14) and (15) set that the drones can be launched and returned once from each node. Meanwhile, Constraint (16) ensures that a drone v is only launched and recollected respectively at two different nodes i and k. These nodes i and k must be visited by its corresponding truck v during its tour, where i and k are not necessary to be adjacent. Additionally, when the drone is launched from the depot, Constraint (17) guarantees that the corresponding truck must depart from any node to visit the rendezvous node k.

The following Constraints (18)-(28) are related to the establishment of time windows. Constraints (18) and (19) set the arrival time of the trucks and drones at the beginning of each route (a^{v}_{0} and a'^{v}_{0}) to zero. These constraints ensure that all vehicle routes will always be started from depot at the beginning of the working period. Constraint (20) ensures the arrival time continuity for the trucks. This provides a guarantee that the arrival time of truck v at a given node k is higher than the arrival time of the same truck at a given node i, if truck v visits i before k. Similarly, Constraints (21) and (22) define the arrival time continuity for the drones. Let us denote a *sortie* operation that consists of two arcs traversed by a drone v : (i,j) and $\{j,k\}$. In this regard, Constraint (21) imposes that the arrival time of drone v at node j is larger than the arrival time of drone v at node i. Likewise, Constraint (22) imposes that $a'^{v}_{k} > a'^{v}_{j} > a'^{v}_{i}$ for the same drone v.

Constraints (23)-(26) guarantee the synchronization of the arrival times of vehicles. Constraints (23)-(24) correspond to the arrival time of trucks and drones at the launch nodes in V_L. These Constraints (23)-(24) set that in order to launch a drone v from a given node i, both of the truck and drone v must be arrived at the node i at the same time (synchronized). Accordingly, Constraints (25)-(26) impose the arrival time of truck and drone v at the rendezvous node k after executing a *sortie* to be the same. This implies that if one of the vehicles arrive that k earlier, it must wait for the arrival of the other one. Constraint (27) ensures that the delivery process executed by a drone must not violate the drone flight endurance time E. The calculation of flight time considers the value of travel times t'_{ij} and t'_{jk}, alongside the time incurred to service customer j (s_j). Constraint (28) guarantees that a truck v cannot launch the drone if its corresponding drone v is still executing a delivery process. Constraints (29) and (30) are the hard time window constraints that ensure that the services at all nodes $i \in V$, either by a truck or a drone, are only performed within the corresponding time window of node i. This is imposed by restrict the value of a^{v}_{i} and a'^{v}_{i} to be in between $[\ o_i, e_i\ ]$. Finally, the domains of the decision variables are defined by Equations (31)-(34).

4. Variable neighborhood search for VRPTWD

In this section, the solution approach for the VRPTWD will be completely presented. The discussion starts with the description of solution representation in Section 4.1. Then, the presentation of the proposed VNS and its parts are given in Section 4.2, Section 4.3, and Section 4.4. Lastly, Section 4.5 discusses the calculation of penalized objective function.

4.1. Solution representation

First, the encoding scheme of the solution representation will be explained. In general, our scheme consists of two arrays: (1) the upper part and (2) the lower part. The upper part of the solution vector reflects both routes and the sequence of customer nodes served by trucks. Non-negative integer numbers are used here, in which the value of '0' indicates the beginning of a new vehicle route and/or the end of the current route, while the non-zero values represent the customer nodes. Therefore, the total length of the upper part of the solution vector is $(n + v + 1)$, where the $(v+1)$ nodes take the value of '0'. On the other hand, the lower part of the solution vector has the same length as the upper part, but comprises binary numbers. These numbers indicate whether the customer node is visited by a truck or a drone. Given a non-negative integer value of $i \in \{2, .., n + v\}$, the lower part works in the following way:

(i) If the i -th number of the lower part takes the value of '0', then the customer node depicted in the i -th number of the upper part will be served by a truck.
(ii) Else, if the i -th number of the lower part takes the value of '1', when the $(i-1)$ -th number of the lower part (the previous

number) takes the value of '0', then the customer node depicted in the i -th number of the upper part will be served by a drone.

(iii) Else, if the i -th number of the lower part takes the value of '1', when the $(i-1)$ -th number of the lower part takes the value of '1', then the customer node depicted in the i -th number of the upper part will be served by a truck.

(iv) In this way, the lower part of the solution vector also controls the launching and rendezvous nodes of the drone. Given a sequence of numbers (with length $>= 1$) that take the value of '1′ in the lower part, the launching and rendezvous nodes are respectively indicated by the upper parts of the predecessor and the successor numbers of the sequence.

(v) Lastly, when the upper part of the i -th number depicts the '0′ value, the active truck and drone must return to the depot.

A simple example of the solution scheme is depicted in Fig. 2. In this example, there are three different routes constructed to serve 15 customer nodes. Starting from the depot, the vehicles in the first route travel together to serve Node 4 and then move to the Node 2. Then, since the next number in the lower part of the vector takes the value of '1', the drone is launched from the Node 2 to serve the Node 11. Meanwhile, during the drone operation, the truck moves to serve Node 9 and Node 6 consecutively. Then, both the truck and drone travel to Node 13 to perform the recollecting process in Node 13. After the recollection process is finished, the vehicles then return to the depot to finish this route. Further, based on these explanations, Fig. 3 visually illustrates the deciphering results of the encoding scheme in Fig. 2.

4.2. Main framework of the VNS

Here, the main framework of the proposed VNS approach is presented. The VNS itself is constructed based on a systematic change between pre-defined neighborhood moves. These systematic moves are executed to progress towards a local optimum solution. Afterwards, a shaking procedure is required to escape from the corresponding local optimum solution, in order to move to the better solution (Hansen and Mladenović, 2018). The particular design of VNS is developed based on a simple notion that (i) a global minimum point can be considered as a local minimum point of all possible structures of neighborhood, and (ii) the local minimum points of one or several neighborhoods of many problems are relatively close to each other (Jarboui et al., 2013). We refer the readers to Hansen and Mladenovic (2018) who provided a comprehensive discussion on the basic concept and recent developments of VNS algorithm.

The detailed procedure of our proposed VNS is presented in Algorithm 1. The algorithm is started by the creation of a neighborhood list $NL = \{1, \cdots, |NL|\}$, which comprises all the neighborhood moves that will be explained later in Section 4.4. An initialization method is then executed with a greedy approach to create a good-quality initial solution. After the initial solution $S_{initial}$ is generated, this solution is then deployed as the current solution S for the main iteration of VNS. This main iteration phase consists of three main parts: (i) a shaking procedure to create a perturbed solution vector S', (ii) a local search procedure based on the classical variable neighborhood descent (VND) algorithm, and (iii) the evaluation of new solution S'' based on the VRPTWD model. The main iteration is continuously executed as long as the stopping criterion of the algorithm has not been met. Finally, the solution with the smallest value of objective function (according to Equation (3)) is returned as the best solution S_{best}.

Algorithm 1. Variable neighborhood search for VRPTWD

Input: VRPTWcase
Create neighborhood list $NL = \{1, \cdots, |NL|\}$
$S_{initial}, f(S_{initial}) \leftarrow$ InitializationMethod (VRPTWcase)
Set $nonImprovementCount \leftarrow 0$, $kMax = |NL|$
Set $S \leftarrow S_{initial}$, $f(S) \leftarrow f(S_{initial})$
Set $S_{best} \leftarrow S_{initial}$, $f(S_{best}) \leftarrow f(S_{initial})$
While stopping criterion is not reached **do** :
  Set $k \leftarrow 1$
  While $k \leq kMax$ **do** :
$r \leftarrow rand[0, 1]$
$S' \leftarrow$ Shaking(S, NL, r)
  $S'', f(S'') \leftarrow$ RandomizedVND(S')
    If $f(S'') < f(S)$ **then** :
      $S \leftarrow S''$, $f(S) \leftarrow f(S'')$
      Set $k \leftarrow 1$
    If $f(S) < f(S_{best})$ **then** :
      $S_{best} \leftarrow S$, $f(S_{best}) \leftarrow f(S)$
    End if
    Else:
    $k \leftarrow k+1$, $nonImprovementCount \leftarrow 0$
    End If
  End While
End While
Output: $S_{best}, f(S_{best})$

4.3. Initialization of solution

The VNS procedure in Algorithm 1 is started with the generation of an initial solution. Our observation during the preliminary design phase showed that the deployment of a pre-defined initialization approach has an advantage to the commonly used randomized approach, as the searching process is started at a better position in the search space.

Algorithm 2 displays the initialization method developed to create an initial VRPTWD solution. This procedure starts with a heuristic approach to decide the number of vehicles v required to create the solution vector. As stated in Section 4.1, our solution representation scheme comprises of two blocks of array, each with a total length of $(n+v+1)$ and the $(v+1)$ nodes of the upper part take the value of '0′ to denote the creation of a new tour. In the case of VRPTWD with limited number of vehicles, the value of v has been pre-defined and the horizontal length of the solution vector can be determined accordingly. However, in another case when the number of available vehicles is unlimited or simply not stated (see Sacramento et al., 2019 who considered a VRPD with unlimited number of vehicles), the value of v must be decided first in order to determine the length of arrays. By considering the worst possible case when each tour is only serving one customer node, the value of v should be defined as n. However, our observation during the design phase revealed that this approach substantially reduces the efficiency of our heuristic approach, as it results in an enormous search space. Therefore, a heuristic approach is developed here to determine the value of v. This approach is generated based on the observation of Sacramento et al. (2019) who stated that, in the context of urban logistics with small-size packages, the capacity constraints may not play a major role even for the large-size instances and the creation of tour may more be related to the maximum duration of tour $Tmax$. Overall, this heuristic approach consists of four steps as follow: (i) calculate $\bar{t}_{ij}$, the mean value of $t_{ij} \in A$, (ii) calculate $\bar{s}_i$, the mean value of $s_i \in N$, (iii) the average number of customer nodes per tour $\overline{L} = \lceil \frac{Tmax}{\bar{t}_{ij}+\bar{s}_i} \rceil$, and (iv) calculate $v = \lceil \frac{n}{L} \rceil . + 1$.

After the solution vector is ready, the next steps are deployed as a

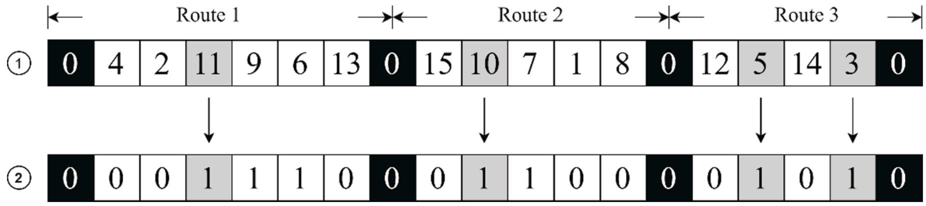

Fig. 2. Solution representation.

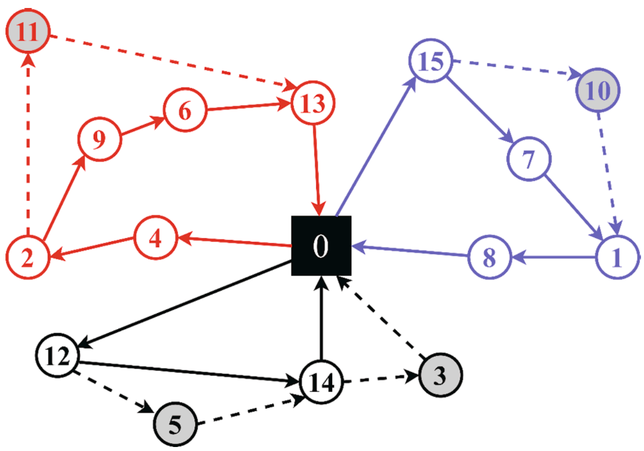

Fig. 3. The illustration of the encoding scheme in Fig. 2.

simple greedy approach to construct the content of the initial solution. The basic ideas of this greedy approach are to:

(i) create a basic truck-only vehicle routing problem (VRP) tour using a simple nearest neighbor approach based on the distance proximity of nodes in V, since the total cost of VRPTWD is highly related to the distance traveled by the vehicles. This nearest neighbor approach is executed by iteratively moves the vehicle to the nearest node from its current location. The process starts from the depot and a new tour is created when the currently created tour violates the capacity and duration constraints.

(ii) optimize the generated VRP tour using the classical 2-Opt approach (Croes, 1958) as defined in Algorithm 3.

(iii) iteratively relocate the customer nodes in the optimized VRP tour into a drone node with the *FindSortie* approach from Sacramento et al. (2019).

Algorithm 2. InitializationMethod

Input: VRPTWcase
Decide the number of vehicles v heuristically
$S_{initial} \leftarrow$ NearestNeighbor (n, v, d_{ij})
$S_{initial}, f(S_{initial}) \leftarrow$ Fast-2Opt $(S_{initial})$
$S_{initial} \leftarrow$ DroneSavings $(S_{initial}, f(S_{initial}))$
$f(S_{initial}) \leftarrow$ ObjectiveCalculation $(S_{initial})$
Output: $S_{initial}, f(S_{initial})$

Algorithm 3. Fast-2Opt

Input: $S_{initial}$
$f(S_{initial}) \leftarrow$ ObjectiveCalculation $(S_{initial})$
Set $i \leftarrow 0$
While $i < 10000(n+v)$:
  $S'_{initial} \leftarrow$ 2-Opt move $(S_{initial})$
  $f(S'_{initial}) \leftarrow$ ObjectiveCalculation $(S'_{initial})$
  If $f(S'_{initial})) < f(S_{initial})$ **then** :
$S_{initial} \leftarrow S'_{initial} f(S_{initial}) \leftarrow f(S'_{initial})$
  Set $i \leftarrow 0$
 End If
$i \leftarrow i + 1$
End While
Output: $S_{initial}, f(S_{initial})$

4.4. *Neighborhood moves, shaking procedure, and local search procedure*

The VNS involves the incorporation of multiple neighborhood moves. In this regard Mladenovic and Hansen (1997) derived three important questions on the VNS framework: (i) "What N_n should be used and how many of them?, (ii) What should be their order in the search?, and (iii) What strategy should be used in changing neighborhood?". These questions will be answered in this subsection.

- *Neighborhood moves*

Overall, there are eight (8) different neighborhood moves to be explored in this study. These are: (i) Random swap node, (ii) Random swap whole, (iii) Random insertion node, (iv) Random insertion whole, (v) Random reverse node, (vi) Random reverse whole, (vii) Remove sortie node, and (viii) Add sortie node. The procedure and illustration of the neighborhood moves are visually presented in Fig. 4. Among these eight moves, three moves (random swap node, random insertion node, random reverse node) are fully dedicated to improving the upper part of solution vector, which corresponds to the improvement of truck tour. Then, for each of them, a modification is proposed within the form of "Whole" move. In these modified moves, the upper and lower parts of the selected array(s) are perturbed simultaneously, resulting in a higher magnitude of adjustment.

Subsequently, the last two moves remove sortie node and add sortie node are presented to directly modify the indication of transportation mode by altering the value of the lower part of the solution vector. On the one hand, the remove sortie node attempts to discard the involvement of a certain node in a *sortie,* by changing the value of the lower part of the vector from '1' to '0'. Note that in our solution scheme, a node with the value of '1' in the lower part does not guarantee that this node is visited by drone, since this node might be visited by a truck while its corresponding drone is executing a *sortie.* In this kind of situation, this neighborhood move will shift the rendezvous node. On the other hand, the add sortie node attempts to involve a certain node into a *sortie* by altering the lower part array from '0' to '1'. This move is similar to the 'Relocate customer' presented by de Freitas and Penna (2020), in which the main aim of this move is to reduce the total cost since the travel cost of drones is generally lower than the trucks.

- *Shaking procedure*

Shaking is a crucial procedure of VNS. This part is presented to assist the searching process by ensuring that the next local search procedure will not be started at a local optimum solution (Hansen and Mladenović, 2018). Algorithm 4 displays the proposed shaking procedure of our VNS

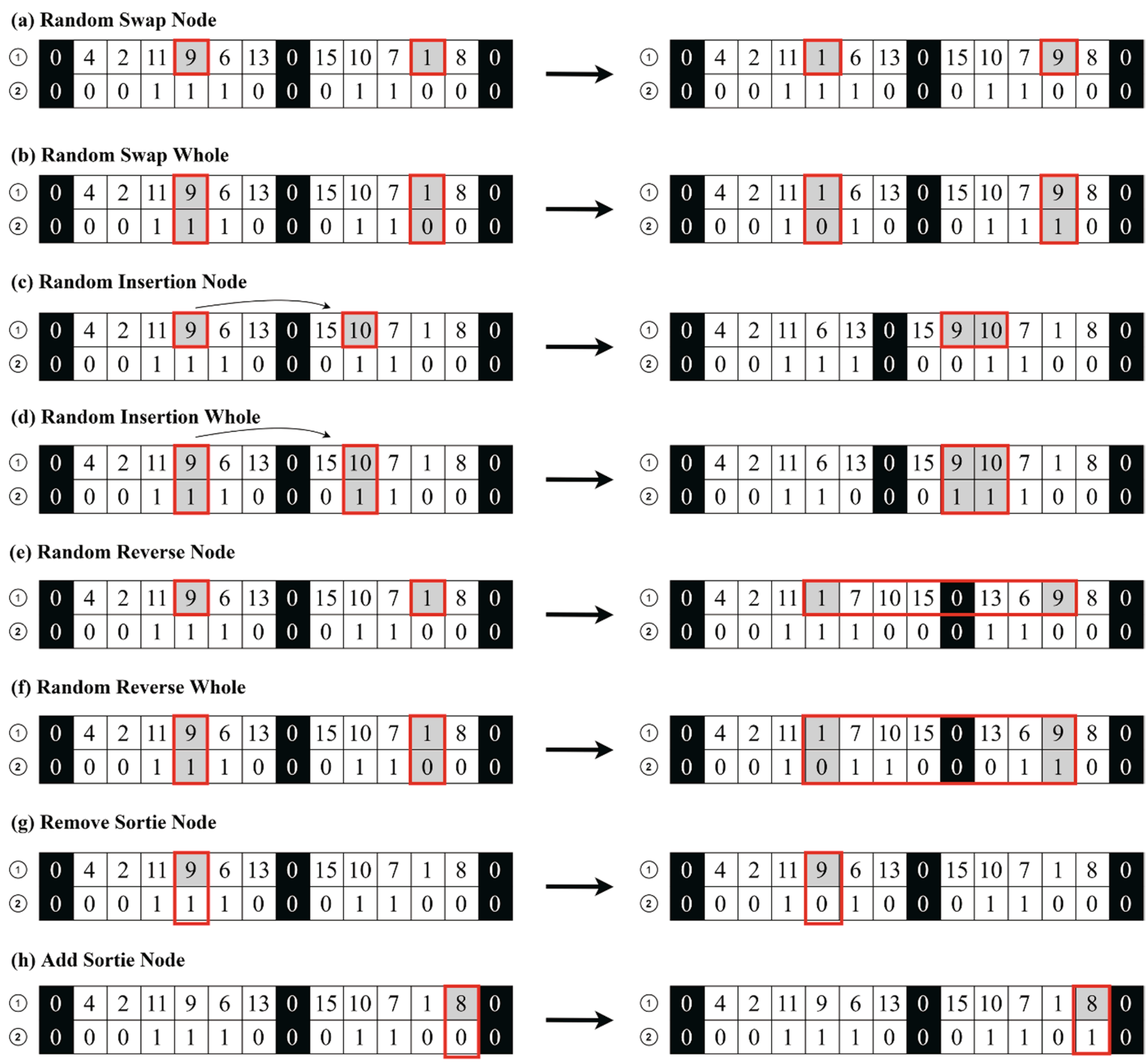

Fig. 4. Neighborhood moves.

algorithm, in which all the neighborhood moves discussed before are included to avoid any complicated modification of the algorithm, so that the replicability of our algorithm can be increased.

Algorithm 4: Shaking procedure

1.	**Input:** S, NL, r
2.	**If** $0 \leq r < 1/\|NL\|$, **then:**
3.	$S' \leftarrow$ RandomSwapNode(S)
4.	**Elseif** $1/\|NL\| \leq r < 2/\|NL\|$, **then:**
5.	$S' \leftarrow$ RandomSwapWhole(S)
6.	**Elseif** $2/\|NL\| \leq r < 3/\|NL\|$, **then:**
7.	$S' \leftarrow$ RandomInsertionNode(S)
8.	**Elseif** $3/\|NL\| \leq r < 4/\|NL\|$, **then:**
9.	$S' \leftarrow$ RandomInsertionWhole(S)
10.	**Elseif** $4/\|NL\| \leq r < 5/\|NL\|$, **then:**
11.	$S' \leftarrow$ RandomReverseNode(S)
12.	**Elseif** $5/\|NL\| \leq r < 6/\|NL\|$, **then:**
13.	$S' \leftarrow$ RandomReverseWhole(S)
14.	**Elseif** $6/\|NL\| \leq r < 7/\|NL\|$, **then:**
15.	$S' \leftarrow$ RemoveSortieNode(S)
16.	**Else:**
17.	$S' \leftarrow$ AddSortieNode(S)
18.	**End If**
19.	**Output:** S'

- *Local search procedure*

In each iteration of VNS, the algorithm calls a local search procedure that systematically explore the search space of the problem. Here, the randomized VND is employed, as presented in Algorithm 5. The selection of randomized VND in our VNS is largely based on de Freitas and Penna (2020), who successfully implemented a VNS-based algorithm with randomized VND for FSTSP. Randomized VND is a modification of the general VND, which itself is the primitive version of VNS (Mladenović and Hansen, 1997). The main characteristic of randomized VND is the inclusion of shuffle procedure (Step 10 of Algorithm 6) to rearrange the neighborhood list NL. This approach is intended to increase the exploration performance of the algorithm by continuously altering the order of the neighborhood moves.

$$p_Z = Z + Endurancepenalty + Truckloadpenalty + Droneloadpenalty + Durationpenalty + Timewarpspenalty \tag{35}$$

Algorithm 5. RandomizedVND

1.	**Input:** S'
2.	$NL \leftarrow \{Allneighborhoodmoves\}$
3.	$f(S') \leftarrow$ ObjectiveCalculation (S')
4.	Set $k \leftarrow 1$
5.	**While** $k \leq kMax$ **do** :
6.	$S'' \leftarrow$ dothemoveinNL(k)
7.	$f(S'') \leftarrow$ ObjectiveCalculation (S'')
8.	**If** $f(S'') < f(S')$ **then** :
9.	$S' \leftarrow S'', f(S') \leftarrow f(S'')$
10.	Shuffle NL
11.	Set $k \leftarrow 1$
12.	**Else** :
13.	$k \leftarrow k + 1$
14.	**End If**
15.	**End While**
16.	$S'' \leftarrow S', f(S'') \leftarrow f(S')$
17.	**Output:** $S'', f(S'')$

4.5. Objective calculation

Moving on, the discussion here is dedicated to explaining the calculation of objective function in our heuristic approach. Since VRPTWD belongs to the class of constrained optimization problem, one issue that must be handled thoroughly in the algorithm is the management of constraints itself, and a popular way to handle constraints in heuristics is by penalizing the infeasible solutions. Penalization of infeasible solutions also holds an importance in the effort of improving the performance of algorithm (Vidal et al., 2013). This penalty function separates the infeasible search space from the feasible one, allowing the efficient exploration of search space.

The search space of VRPTWD, as indicated in Equations (3)-(34), involves several prominent constraints that must be complied. These constraints can be classified as (i) endurance constraints, (ii) truck load constraints, (iii) drone load constraints, (iv) travel duration constraints, and (v) time window constraints. Therefore, the penalized objective function p_Z must be calculated based on the presence of these constraints. Here, a major focus is given to the time window constraints which constitute to our main contribution in this study.

This study borrows a time window relaxation concept called "Time warps" (Nagata et al., 2010). This concept has been successfully implemented in VRPTW by Vidal et al. (2013). Consider $[o_i, e_i]$ as the opening and closing time window of node i, this time warps relaxation concept works by two simple steps as follow: (i) When a vehicle v arrives earlier than the opening time o_i, it needs to wait for the opening time without any penalty, so that $a_i^v = o_i$, while (ii) if a vehicle v arrives after e_i, it pays for a time warp $W(v)$ with a value of $a_i^v - e_i$, so that $a_i^v = e_i$.

All in all, the calculation of penalized objective function can then be described as follows. First, let p as the pre-defined penalty value which can be set as a large number (i.e. $p = \sum_{i,j \in A} C_{ij}$). Then, for each active set of vehicles v, the vehicle corresponds to a route which starts from $\{0\}$ and finishes at $\{n+1\}$. Between these nodes $\{0, n+1\}$, the truck v may visits all nodes in V that is indicated by $a_i^v \geq 0$, while the corresponding drone v may visits all nodes in V_D that is indicated by $a'^v_i > 0$. Thus, the following sets can be constructed to characterize the vehicle routes. Let $r_v^t = \{i \in V | a_i^v \geq 0\}$ be a set of nodes that are visited by truck v and $r_v^d = \{i \in V_D | a'^v_i > 0\}$ as a set of nodes that are visited by drone v. Then, the calculation of penalty function and the penalized objective function can be presented as in Equations (35) and (36).

$$\begin{aligned} p_Z = \ & Z + \sum_{v \in U} p \bullet \max(0, \sum_{i,k \in r_v^t, j \in r_v^d} \sum y_{ijk}^v (t'_{ij} + t'_{jk}) - E) \\ & + \sum_{v \in U} p \bullet \max(0, \sum_{i \in r_v^t} q_i - Qt) + \sum_{v \in U} p \bullet \max(0, \sum_{i \in r_v^d} q_i - Qd) \\ & + \sum_{v \in U} p \bullet \max(0, a_{n+1}^v - Tmax, a'^v_{n+1} - Tmax) + \sum_{v \in U} p \bullet W(v) \end{aligned} \tag{36}$$

5. Numerical experiments

This section provides the results of our attempt to evaluate the performance of the proposed VNS. Our proposal is tested to two benchmark options, namely the exact solution from the MIP model and the ALNS algorithm adapted from Sacramento et al. (2019), which is the state-of-the-art algorithm for VRPD.

All algorithms are coded in MATLAB R2018b and implemented on a personal computer with AMD Ryzen 5 2600 Six-Core Processor 3.4 GHz, 16 GB of memory, NVIDIA GeForce GTX 1650 Super GPU, and a Windows 10 operating system. On the other hand, the MIP model is solved using optimization software GUROBI version 9.0.1 in the same personal computer. The discussion of results starts from the description of test instance generation in Section 5.1 and the parameter settings of algorithms in Section 5.2. Then, the complete experiment results are discussed in Section 5.3. Afterwards, some managerial insights are drawn for the readers in Section 5.4.

5.1. Test instances

In order to assess the feasibility of the proposed VNS, a dataset for computational testing is required. However, since this is the first study to consider the presence of time windows in the VRPD, there is no dataset available yet in the literature. Therefore, a new dataset is generated based on the VRPD dataset of Sacramento et al. (2019) for the purpose of this study.

This dataset consists of 112 different instances with various sizes. This dataset can be separated into two distinct classes: (1) small-size and (2) larger-size instances. The first set of instances comprises 48 instances with 5, 10, or 15 customer nodes, while the second set comprises 64 instances with 20, 30, 40, or 50 customer nodes. In all of these instances, a single central depot is located at coordinates (0,0). Then, the customer nodes are randomly located within a *dxd* square grid, using a uniform distribution $U(-d, d)$, where d stands for the size of the grid area and takes a value between $\{5, 10\}$. For each combination of the number of customer nodes and grid size, there are 16 different scenarios based on the density of time windows (%*TW*) and the width of the time window (w). The value of %*TW* is selected according to the set $\{25\%, 50\%, 75\%, 100\%\}$, while the value of w is selected from the set $\{30min, 120min\}$.

Afterwards, the time windows of each instance are created by inserting a randomly generated hard time window into a pre-defined portion of customer nodes. To ensure the feasibility of the modified instances, the classical study of VRPTW from Solomon (1987) is closely followed to generate the time window, where a simple three-step strategy as follow is deployed:

1. Decide the value of %*TW* and w for this instance based on the aforementioned sets.
2. Generate a random permutation of n customer nodes.
3. Generate a random time window $[o_i, e_i]$ for each of the first %*TW* customers of the generated permutation from the previous step. First, the center of time window for each considered customer i is randomly generated within the interval of $[(o_0 + t^{'}_{0,i}),(e_0 - t^{'}_{i,0} - s_i)]$, where $o_0 = 0$ and $e_0 = Tmax$. Then, for each of these considered customer nodes, incur a time window according to the selected value of w.

Finally, all instances of this modified dataset have been made available at https://sites.google.com/view/setyotw/resources, while Table 2 summarizes the selected value of parameters of the modified instances. For the values of most of these parameters, we refer the readers to the study of Sacramento et al. (2019).

Table 3
Parameter settings of the algorithms.

Algorithm	Parameter	Meaning	Value
VNS	*nonImprovementLimit*	Stopping criterion 1: non improvement limit of VNS	$10000(n+v)$
	timeLimit	Stopping criterion 2: time limit of VNS	10 minutes
ALNS	*nonImprovementLimit*	Stopping criterion 1: non improvement limit of ALNS	$10000(n+v)$
	timeLimit	Stopping criterion 2: time limit of ALNS	10 minutes
	T_0	Initial temperature multiplier	0.04
	δ	Degree of destruction	0.15
	ρ	Reaction factor	0.90
	$\sigma_1, \sigma_2, \sigma_3, \sigma_4$	Weight adjustment parameters	{33,9,13,0}

5.2. Parameter settings

To ensure a fair comparison between algorithms, we discuss the settings of the considered algorithms in this subsection. For the VNS, since this algorithm is designed as a parameter-less algorithm, therefore, the only settings to be disclosed are the stopping criteria of the VNS. On the other hand, for the benchmark algorithm ALNS, the presentation of Sacramento et al. (2019) is closely followed to obtain the optimal parameter setting. In this regard, Table 3 provides the summary of parameter settings.

5.3. Experiment results – analysis of algorithms

Here, the experiment results are presented. The presentation will be separated into two parts based on the class of instances.

5.3.1. Small-size instances

In this stage, the performance of VNS will be evaluated by comparing it with the solutions from GUROBI commercial software. Since the proposed VNS belongs to the class of stochastic search and employs random process within its procedure, then the VNS is executed 10 times for each instance to ensure the robustness of the results. Additionally, we also compare the results of VNS to the ALNS proposed by Sacramento et al. (2019) for VRPD. This additional comparison is aimed to verify that the ALNS can obtain near-optimal solutions for VRPTWD so that the VNS and ALNS can be compared afterwards for the larger-size instances.

The computational results on small-size instances are presented in Table 4. The first part of Table 4 describes the number of customer nodes (n), the size of grid area (d), the density of time windows (%*TW*), and the width of time windows (w). Then the second part presents the obtained optimal solutions from the GUROBI (z^*) alongside the optimality gap (*GAP*) and its running time (r^*). Similarly, the third and fourth parts present the results of VNS and ALNS, which comprise the best objective value obtained (${z_{best}}^{VNS}$, ${z_{best}}^{ALNS}$), the average objective value (${z_{avg}}^{VNS}$, ${z_{avg}}^{ALNS}$), the standard deviation (${z_{stdev}}^{VNS}$, ${z_{stdev}}^{ALNS}$), and the running time required of each algorithm (r^{VNS}, r^{ALNS}). Finally, the last part of Table 4 presents the gaps between the best (Δz_{best}) and average (Δz_{avg}) objective values of VNS to the results from GUROBI, which formally calculated as in Equations (37)-(38). Since the considered VRPTWD has a minimization objective, therefore, a positive value in Δz_{best} or Δz_{avg} indicates that the VNS obtains better result (smaller objective value).

$$\Delta z_{best} = \left(\frac{z^* - {z_{best}}^{VNS}}{z^*}\right)x100\% \tag{37}$$

$$\Delta z_{avg} = \left(\frac{z^* - {z_{avg}}^{VNS}}{z^*}\right)x100\% \tag{38}$$

Overall, the results in Table 4 shows the feasibility of our proposed VNS to be implemented as a solution approach for VRPTWD. It is observed that the VNS, alongside the ALNS, can always find solutions with equivalent quality to the GUROBI for the small-size instances. Further, our experiments also find that the MIP is not applicable to solve larger size instances as the commercial solver GUROBI cannot obtain the optimal solutions for instances with $n = 15$. Therefore, from the results of this experiment, this study endorses the implementation of VNS as the solution approach for VRPTWD.

5.3.2. Larger-size instances

After confirming the effectiveness of the VNS, the next experiments with larger-size instances aim to confirm the applicability of VNS for larger-size instances. Table 5 presents the complete computational results for this second set of instances, in which we compare the performance of VNS with the ALNS from Sacramento et al. (2019). For clarity, we have verified and validated the performance of the ALNS in accordance with Sacramento et al. (2019) by testing its performance in VRPD

Table 4
Experiment results for small-size instances.

Inst.	n	$d(km^2)$	$\%TW$	w (min)	MIP			VNS				ALNS (Sacramento et al., 2019)				Δz_{best}	Δz_{avg}
					z^*	GAP	r^*	z_{best}^{VNS}	z_{avg}^{VNS}	z_{stdev}^{VNS}	r^{VNS}	z_{best}^{ALNS}	z_{avg}^{ALNS}	z_{stdev}^{ALNS}	r^{ALNS}		
1	5	5x5	25	30	0.69	0.00	4.17	0.69	0.69	0.00	25.18	0.69	0.76	0.15	44.84	0,00	0,00
2	5	5x5	50	30	0.58	0.00	2.15	0.58	0.58	0.00	26.44	0.58	0.70	0.13	40.86	0,00	0,00
3	5	5x5	75	30	0.62	0.00	1.11	0.62	0.62	0.00	28.69	0.64	0.71	0.12	44.08	0,00	0,00
4	5	5x5	100	30	1.14	0.00	0.22	1.16	1.16	0.00	27.77	1.17	1.56	0.42	39.69	−0,02	−0,02
5	5	5x5	25	120	0.94	0.00	4.38	0.94	0.94	0.00	24.93	0.94	1.01	0.10	43.00	0,00	0,00
6	5	5x5	50	120	0.97	0.00	1.77	0.97	0.97	0.00	25.27	0.97	1.08	0.16	41.96	0,00	0,00
7	5	5x5	75	120	0.57	0.00	1.03	0.57	0.57	0.00	27.03	0.57	0.70	0.20	39.55	0,00	0,00
8	5	5x5	100	120	0.96	0.00	1.21	0.96	0.96	0.00	29.05	0.96	1.08	0.11	39.42	0,00	0,00
9	5	10x10	25	30	1.98	0.00	1.08	1.98	1.98	0.00	27.25	1.98	2.02	0.09	41.58	0,00	0,00
10	5	10x10	50	30	1.96	0.00	0.56	1.96	1.96	0.00	26.60	1.96	2.20	0.15	41.76	0,00	0,00
11	5	10x10	75	30	1.44	0.00	1.16	1.44	1.44	0.00	28.50	1.44	1.71	0.18	39.42	0,00	0,00
12	5	10x10	100	30	0.91	0.00	0.73	0.94	0.94	0.00	27.71	0.94	1.36	0.31	40.38	−0,03	−0,03
13	5	10x10	25	120	2.07	0.00	0.91	2.07	2.07	0.00	25.22	2.07	2.25	0.17	39.20	0,00	0,00
14	5	10x10	50	120	1.39	0.00	1.55	1.43	1.43	0.00	27.74	1.43	1.53	0.23	45.44	−0,03	−0,03
15	5	10x10	75	120	1.94	0.00	1.88	1.94	1.94	0.00	28.37	2.00	2.03	0.03	37.85	0,00	0,00
16	5	10x10	100	120	1.35	0.00	0.72	1.35	1.35	0.00	27.71	1.35	1.69	0.78	43.16	0,00	0,00
17	10	5x5	25	30	1.10	54.44	7200.00	1.11	1.12	0.01	52.01	1.11	1.12	0.02	79.02	−0,01	−0,02
18	10	5x5	50	30	1.36	0.00	247.07	1.36	1.36	0.00	51.56	1.36	1.38	0.03	82.25	0,00	0,00
19	10	5x5	75	30	1.08	0.00	93.74	1.10	1.10	0.00	55.77	1.33	1.39	0.11	112.86	−0,02	−0,02
20	10	5x5	100	30	0.98	11.52	7200.00	1.00	1.00	0.00	66.75	1.09	1.27	0.18	105.17	−0,02	−0,02
21	10	5x5	25	120	1.14	46.71	7200.00	1.14	1.14	0.00	57.55	1.17	1.18	0.01	116.79	0,00	0,00
22	10	5x5	50	120	0.86	43.41	7200.00	0.86	0.87	0.02	62.83	0.90	1.00	0.10	101.21	0,00	−0,02
23	10	5x5	75	120	1.39	4.34	7200.00	1.39	1.39	0.00	54.51	1.40	1.52	0.08	114.06	0,00	0,00
24	10	5x5	100	120	1.21	34.62	7200.00	1.22	1.22	0.00	53.18	1.41	1.47	0.04	89.13	−0,01	−0,01
25	10	10x10	25	30	2.37	36.49	7200.00	2.40	2.45	0.02	45.90	2.46	2.46	0.00	92.58	−0,01	−0,03
26	10	10x10	50	30	2.46	49.09	7200.00	2.50	2.51	0.02	64.27	2.64	2.71	0.13	91.90	−0,02	−0,02
27	10	10x10	75	30	2.54	36.36	7200.00	2.54	2.54	0.00	55.79	3.15	3.55	0.23	73.27	0,00	0,00
28	10	10x10	100	30	2.95	0.00	193.75	2.97	2.97	0.00	65.92	3.32	3.77	0.45	84.30	−0,01	−0,01
29	10	10x10	25	120	2.43	53.99	7200.00	2.45	2.45	0.00	46.96	2.67	2.69	0.06	76.95	−0,01	−0,01
30	10	10x10	50	120	2.45	16.27	7200.00	2.43	2.44	0.01	56.26	2.43	2.73	0.27	87.94	0,01	0,01
31	10	10x10	75	120	1.95	49.65	7200.00	1.95	1.95	0.00	69.81	2.24	2.33	0.05	103.10	0,00	0,00
32	10	10x10	100	120	3.29	26.22	7200.00	3.33	3.36	0.04	70.29	3.39	3.50	0.07	84.31	−0,01	−0,02
33	15	5x5	25	30	1.32	28.24	7200.00	1.31	1.34	0.07	82.72	1.31	1.44	0.12	146.75	0,01	−0,01
34	15	5x5	50	30	1.49	54.11	7200.00	1.48	1.51	0.04	97.97	1.60	1.69	0.09	184.84	0,01	−0,02
35	15	5x5	75	30	1.53	49.56	7200.00	1.59	1.61	0.01	85.33	1.81	1.92	0.09	256.66	−0,04	−0,06
36	15	5x5	100	30	1.75	45.35	7200.00	1.93	1.99	0.10	113.39	2.00	2.22	0.20	218.32	−0,10	−0,14
37	15	5x5	25	120	1.85	54.20	7200.00	1.70	1.79	0.05	67.87	1.70	1.82	0.07	135.21	0,09	0,03
38	15	5x5	50	120	1.62	63.65	7200.00	1.35	1.40	0.07	95.44	1.38	1.46	0.11	134.62	0,17	0,14
39	15	5x5	75	120	1.48	55.40	7200.00	1.54	1.57	0.05	89.15	1.62	1.73	0.11	168.21	−0,04	−0,07
40	15	5x5	100	120	1.50	59.08	7200.00	1.53	1.64	0.06	94.28	1.65	1.75	0.08	195.50	−0,02	−0,10
41	15	10x10	25	30	2.71	42.97	7200.00	2.48	2.49	0.01	83.51	2.48	2.67	0.41	124.23	0,08	0,08
42	15	10x10	50	30	3.11	41.29	7200.00	2.85	2.92	0.09	102.09	3.04	3.10	0.10	139.09	0,08	0,06
43	15	10x10	75	30	3.99	36.98	7200.00	3.95	4.24	0.32	101.09	4.63	5.25	0.50	162.28	0,01	−0,06
44	15	10x10	100	30	3.77	63.48	7200.00	3.82	3.89	0.03	98.37	4.28	4.49	0.17	155.09	−0,01	−0,03
45	15	10x10	25	120	8.85	86.79	7200.00	2.98	3.01	0.03	84.06	2.98	3.07	0.12	129.94	0,66	0,66
46	15	10x10	50	120	3.90	66.58	7200.00	3.19	3.25	0.05	87.87	3.19	3.31	0.16	141.03	0,18	0,17
47	15	10x10	75	120	3.54	66.38	7200.00	3.03	3.27	0.19	95.46	3.49	3.73	0.27	144.13	0,14	0,08
48	15	10x10	100	120	2.80	59.62	7200.00	2.93	3.07	0.11	87.01	3.29	3.48	0.24	173.13	−0,05	−0,10
Average					1,96	27,85	4361,65	1.81	1.84	0.03	58.93	1.92	2.08	0.17	99.29	0,02	0,01

Table 5
Experiment results for larger-size instances.

Inst.	n	$d(km^2)$	%TW	w (min)	VNS				ALNS (Sacramento et al., 2019)				Δz_{best}	Δz_{avg}	$\Delta z_{avg-best}$
					z_{best}^{VNS}	z_{avg}^{VNS}	z_{stdev}^{VNS}	r^{VNS}	z_{best}^{ALNS}	z_{avg}^{ALNS}	z_{stdev}^{ALNS}	r^{ALNS}			
49	20	5x5	25	30	1.76	1.88	0.14	141.62	1.77	1.88	0.11	233.39	0.01	0.03	−0.03
50	20	5x5	50	30	2.02	2.12	0.14	151.95	2.28	2.41	0.13	233.00	0.12	0.12	0.07
51	20	5x5	75	30	1.83	1.96	0.21	141.23	2.23	2.68	0.28	298.76	0.18	0.26	0.11
52	20	5x5	100	30	1.70	1.80	0.07	107.95	2.17	2.55	0.28	222.06	0.22	0.29	0.17
53	20	5x5	25	120	1.67	1.77	0.05	110.15	1.66	1.77	0.11	302.27	0.07	0.04	−0.03
54	20	5x5	50	120	1.60	1.69	0.08	164.53	1.63	1.73	0.10	286.40	0.02	0.02	−0.03
55	20	5x5	75	120	2.19	2.26	0.10	121.86	2.20	2.42	0.23	192.36	0.03	0.06	−0.03
56	20	5x5	100	120	1.79	1.88	0.06	121.52	2.06	2.26	0.13	277.48	0.16	0.16	0.08
57	20	10x10	25	30	3.38	3.45	0.06	112.69	3.32	3.62	0.26	253.38	−0.02	0.03	−0.06
58	20	10x10	50	30	3.81	4.38	0.34	159.39	4.43	4.54	0.12	318.53	0.14	0.06	0.04
59	20	10x10	75	30	3.29	3.46	0.14	176.77	3.76	4.47	0.49	276.14	0.13	0.22	0.07
60	20	10x10	100	30	5.40	5.82	0.49	105.71	6.82	8.44	1.45	258.60	0.22	0.32	0.16
61	20	10x10	25	120	2.91	2.94	0.06	105.99	2.92	3.12	0.20	294.42	0.01	0.05	−0.02
62	20	10x10	50	120	3.90	4.13	0.26	96.64	3.96	4.12	0.21	246.31	0.02	−0.02	−0.07
63	20	10x10	75	120	3.51	3.70	0.36	194.41	3.68	4.21	0.40	234.77	0.05	0.10	−0.03
64	20	10x10	100	120	3.58	3.83	0.17	157.07	4.47	4.67	0.16	220.20	0.20	0.17	0.13
65	30	5x5	25	30	2.28	2.41	0.12	325.66	2.42	2.92	0.31	430.09	0.10	0.17	−0.01
66	30	5x5	50	30	2.53	2.56	0.05	316.01	2.72	3.08	0.33	525.25	0.07	0.14	0.03
67	30	5x5	75	30	2.63	3.13	0.34	229.43	4.02	4.29	0.33	448.79	0.35	0.27	0.22
68	30	5x5	100	30	3.48	3.72	0.20	201.60	5.18	1563.82	2370.06	465.29	0.35	–	0.25
69	30	5x5	25	120	2.53	2.59	0.06	378.31	2.33	2.65	0.36	540.84	−0.09	−0.01	−0.15
70	30	5x5	50	120	2.41	2.58	0.11	337.45	2.50	2.89	0.33	394.50	0.05	0.10	−0.04
71	30	5x5	75	120	2.94	3.22	0.21	303.79	3.30	3.73	0.45	439.27	0.11	0.12	0.01
72	30	5x5	100	120	2.66	2.98	0.25	292.99	3.16	3.47	0.28	474.98	0.18	0.16	0.08
73	30	10x10	25	30	4.06	4.25	0.22	364.98	4.08	4.68	0.63	479.85	0.01	0.08	−0.05
74	30	10x10	50	30	4.90	5.18	0.23	289.77	6.04	6.52	0.28	565.50	0.19	0.20	0.14
75	30	10x10	75	30	5.41	5.85	0.51	261.00	7.19	7.79	0.64	484.11	0.31	0.26	0.19
76	30	10x10	100	30	6.38	6.82	0.55	266.07	8.22	362.59	787.41	499.85	0.24	–	0.18
77	30	10x10	25	120	4.15	4.28	0.15	227.78	3.93	4.24	0.35	522.61	0.03	0.02	−0.05
78	30	10x10	50	120	5.26	5.80	0.36	255.42	5.54	6.09	0.59	468.80	0.05	0.03	−0.07
79	30	10x10	75	120	5.34	5.64	0.34	270.60	6.36	6.55	0.21	427.49	0.16	0.15	0.12
80	30	10x10	100	120	5.52	5.63	0.10	277.06	6.86	8.34	0.90	397.66	0.20	0.32	0.18
81	40	5x5	25	30	2.72	3.01	0.20	496.60	2.79	3.36	0.39	566.17	0.05	0.14	−0.04
82	40	5x5	50	30	2.67	3.01	0.24	527.25	3.36	4.41	0.60	514.02	0.21	0.32	0.11
83	40	5x5	75	30	3.73	3.96	0.17	410.40	4.92	5.98	0.92	600.00	0.24	0.32	0.18
84	40	5x5	100	30	3.45	3.95	0.41	330.04	5.48	2803.90	2349.43	600.00	0.37	–	0.28
85	40	5x5	25	120	2.68	2.84	0.20	498.99	2.74	2.81	0.06	598.16	0.04	−0.01	−0.03
86	40	5x5	50	120	2.77	2.93	0.15	357.39	3.14	3.56	0.44	600.00	0.12	0.15	0.04
87	40	5x5	75	120	2.84	3.10	0.23	364.15	3.31	4.12	0.64	579.06	0.14	0.23	0.04
88	40	5x5	100	120	2.56	2.73	0.12	522.24	4.06	4.75	0.49	533.64	0.37	0.41	0.31
89	40	10x10	25	30	5.53	6.16	0.43	448.86	6.35	6.88	0.67	600.00	0.13	0.12	0.04
90	40	10x10	50	30	6.02	7.13	0.69	499.88	8.79	9.56	0.75	600.00	0.31	0.25	0.19
91	40	10x10	75	30	7.38	8.00	0.71	382.96	8.97	248.39	532.32	600.00	0.18	–	0.13
92	40	10x10	100	30	8.11	8.98	1.08	333.59	12.22	107.20	130.41	568.34	0.34	–	0.24
93	40	10x10	25	120	5.71	6.04	0.28	405.30	5.39	6.13	0.89	588.27	0.01	0.03	−0.10
94	40	10x10	50	120	5.57	5.80	0.24	568.16	5.80	6.73	0.59	582.31	0.04	0.11	−0.04
95	40	10x10	75	120	6.51	6.72	0.17	507.22	7.69	8.49	0.86	588.14	0.20	0.22	0.14
96	40	10x10	100	120	5.82	6.47	0.49	374.13	8.53	9.46	0.75	519.32	0.32	0.31	0.24
97	50	5x5	25	30	3.49	3.87	0.23	486.30	4.44	4.74	0.33	600.00	0.25	0.21	0.16
98	50	5x5	50	30	3.32	3.45	0.09	580.70	3.56	4.90	0.80	600.00	0.14	0.32	0.06
99	50	5x5	75	30	4.22	4.81	0.46	583.06	6.49	7.01	0.49	600.00	0.35	0.32	0.27
100	50	5x5	100	30	4.57	4.88	0.32	438.39	7.66	3100.16	4584.08	600.00	0.42	–	0.38
101	50	5x5	25	120	3.27	3.42	0.25	537.43	3.19	3.56	0.31	600.00	0.03	0.04	−0.08
102	50	5x5	50	120	3.56	3.74	0.29	496.59	4.58	4.81	0.25	600.00	0.27	0.23	0.20
103	50	5x5	75	120	3.64	3.87	0.19	555.56	4.79	5.42	0.62	600.00	0.24	0.27	0.18
104	50	5x5	100	120	3.98	4.15	0.21	510.23	4.18	6.25	1.19	600.00	0.08	0.33	0.01
105	50	10x10	25	30	5.92	6.49	0.61	587.11	6.60	7.65	0.88	600.00	0.10	0.13	0.00
106	50	10x10	50	30	7.62	8.41	0.62	565.14	10.03	11.36	1.29	600.00	0.26	0.28	0.18
107	50	10x10	75	30	9.95	10.22	0.18	412.59	12.03	1259.23	1415.28	600.00	0.23	–	0.16
108	50	10x10	100	30	9.71	10.21	0.44	485.97	–	–	–	600.00	–	–	–
109	50	10x10	25	120	5.77	6.09	0.41	521.67	5.94	6.35	0.40	600.00	0.03	0.01	−0.06
110	50	10x10	50	120	6.84	7.41	0.47	517.98	7.21	9.20	1.43	600.00	0.08	0.22	0.00
111	50	10x10	75	120	6.77	7.65	1.16	588.35	9.13	10.88	1.01	600.00	0.26	0.31	0.18
112	50	10x10	100	120	8.91	9.60	0.65	520.06	15.49	3227.21	6627.93	600.00	0.42	–	0.36
Average					4.75	5.12	0.34	416.92	5.80	274.43	400.42	550.05	0.16	0.17	0.08

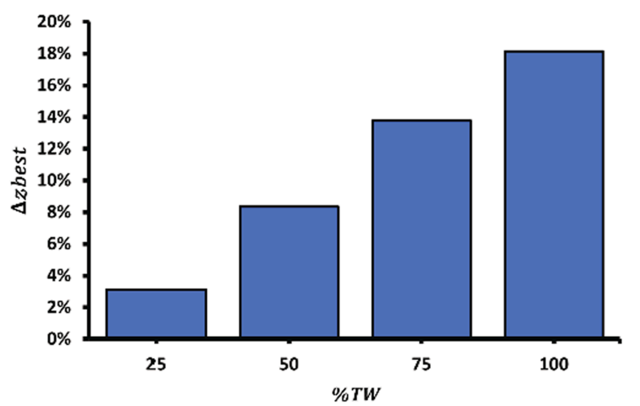

Fig. 5. Comparison of average gaps between VNS and ALNS according to.%TW

environment (without time windows). Similar to the previous experiment, we now calculate the gap between the best solutions of both VNS and ALNS (Δz_{best}), the gap between the average solutions of VNS and ALNS (Δz_{avg}), and the gap between the average solutions of VNS to the best solutions of ALNS ($\Delta z_{avg-best}$). These gaps are formally calculated as in Equations (39)-(41). Likewise, positive values indicate that the VNS obtains better result than ALNS.

$$\Delta z_{best} = \left(\frac{z_{best}{}^{ALNS} - z_{best}{}^{VNS}}{z_{best}{}^{ALNS}}\right)x100\% \tag{39}$$

$$\Delta z_{avg} = \left(\frac{z_{avg}{}^{ALNS} - z_{avg}{}^{VNS}}{z_{avg}{}^{ALNS}}\right)x100\% \tag{40}$$

$$\Delta z_{avg-best} = \left(\frac{z_{best}{}^{ALNS} - z_{avg}{}^{VNS}}{z_{best}{}^{ALNS}}\right)x100\% \tag{41}$$

Overall, the results in Table 5 show that the proposed VNS can always obtain the feasible solutions for all instances. The results also show that the performance of VNS is better than the ALNS, with $\Delta z_{best} = 0.16$, $\Delta z_{avg} = 0.17$, and $\Delta z_{avg-best} = 0.08$. These results confirm the feasibility of our VNS as a solution approach for VRPTWD. Note that the calculation of Δz_{avg} and $\Delta z_{avg-best}$ omit several instances where ALNS delivers penalized objective values to ensure a fair comparison.

Regarding the results of ALNS, it can also be observed that the objective gap between ALNS and VNS grows considerably large as the size of the instance increases. As seen in the columns $z_{avg}{}^{ALNS}$ and $z_{stdev}{}^{ALNS}$, the ALNS failed to obtain any feasible solution in several repetitions of multiple instances, and therefore, the penalized objective functions are listed with value > 100. Another interesting observation can be seen in Fig. 5. It is shown that the average gap between VNS and ALNS generally escalates as the value of %TW enlarges. Additionally, the ALNS also did not return any feasible solution in all the repetitions performed in instance 108, which can be considered as the most complex instance within the dataset due to the maximum value of %TW and the tightness of the time window width w. These findings indicates that the performance of ALNS deteriorates when facing instances with complex time window configuration (high %TW, low w). This observation is reasonable enough, since the ALNS was proposed by Sacramento et al. (2019) for VRPD and is not designed to deal with the presence of time windows. Nevertheless, we note that the ALNS is also not equipped with any specific mechanism to tackle the time window constraints, this fact indicates that the searching process of VNS through the search space of VRPTWD is more effective. Moreover, we also note that the VNS performs more efficient, since it obtains the feasible solutions in shorter runtime than ALNS.

5.4. Managerial discussions

Based on the results of our computational experiments, some interesting managerial insights from the experiments are drawn here. For clarity purpose, we are aware that Sacramento et al. (2019) have presented a comprehensive discussion on several problem parameters of VRPD which are related to the features of drones, namely the impact of altering the cost of battery, the battery endurance, the payload capacity, and the speed of drones. In this regard, we would like to present a summary on the results of their experiments as follow:

- *Battery cost*: From the economic perspective, the usage of drones in a delivery mission is to exploit the assumption that drones generally incur a cheaper travel cost than a traditional truck. Correspondingly, in the study of Sacramento et al. (2019), as well as in this study, the travel cost of drones is set to be 10% of the fuel consumption of trucks ($\alpha = 10\%$). Obviously, this substantial saving will provide the decision-maker an incentive to deploy the drones as often as. Therefore, one might find it interesting to see what would happen if the cost of using a drone becomes more expensive. In this regard, the experiment of Sacramento et al. (2019) has indicated that as the variable cost of drones increases, the number of sorties executed is reduced, and the potential savings from deploying drones may become negligible as the travel cost of drones increases to 50% of the truck cost.
- *Battery endurance and payload capacity*: When compared to the traditional truck, the main disadvantage of drones are perhaps its limited battery endurance and load capacity. In terms of the endurance of drone battery, Sacramento et al. (2019) observed that if the battery endurance enlarges, the potential cost savings from drone deployments may also improve. With a larger battery capacity, the drone will be able to travel through the arcs with larger distance, which in turns will lessen the total travel cost as well due to reduction of distance traveled by trucks. Subsequently, the similar trend is also found in the payload capacity as well. Sacramento et al. (2019) noted that the increasing payload capacity of drones may led to a more cost-efficient delivery mission, since the increasing number of potential drone customers will result in the ability to launch the drones more often.
- *Drone speed*: Related to the battery endurance, Sacramento et al. (2019) discussed that the higher speed of drones will expand the set of feasible sorties since the travel time required to travel through the corresponding arcs becomes lower. Thus, as the speed of drones increases, the higher cost-savings might be obtained.

With respect to the similarity of VRPD and VRPTWD, we argue that those observations from Sacramento et al. (2019) for VRPD remain valid for VRPTWD as well. Therefore, in the rest of this subsection, we will give more attention to the insights derived from the problem characteristics related to the time windows.

5.4.1. Impact of the presence of time windows and drones

From the operational research perspective, the VRPTWD can be seen as a generalization of the classical VRPTW (Solomon, 1987) and the VRPD (Sacramento et al., 2019). The former problem constitutes a delivery task with a set of trucks to a set of customer nodes which have predetermined time windows, meanwhile, the VRPD considers a delivery task with a set of truck-drone tandems to a set of customer nodes without time windows. From there, it can be concluded that the main aspects that constitute the VRPTWD are the presence of time windows of the customers and the presence of drones to assist the delivery tasks. Therefore, two additional experiments are executed to study the effect of these aspects.

By adapting the proposed VNS, we solve all the 112 instances under the situation of VRPTW and VRPD, respectively. Afterwards, the results of VRPTW and VRPD are compared to the previous results from

Table 6
Summary of comparison between VRPD, VRPTW, and VRPTWD (results are in the average value).

	z	T	$dist$	$dist_t$	$dist_d$	$\#v$	$\#d$	$\%d$
VRPD	2.63	58.34	58.34	16.43	41.91	1.17	11.34	0.50
VRPTW	5.61	408.02	44.05	44.05	–	2.08	–	–
VRPTWD	3.21	414.03	80.18	20.80	59.37	2.00	11.72	0.51

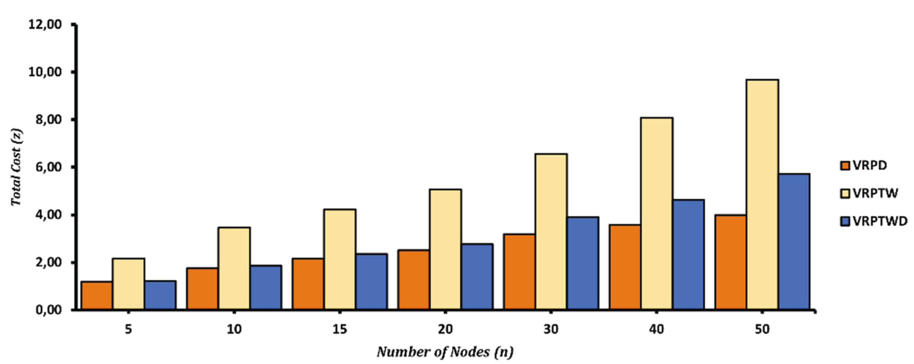

Fig. 6. Comparison of total costs between VRPD, VRPTW, and VRPTWD.

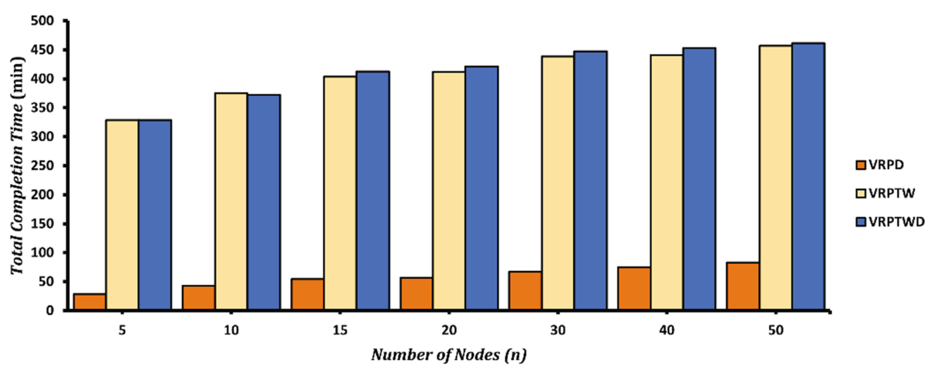

Fig. 7. Comparison of total completion times between VRPD, VRPTW, and VRPTWD.

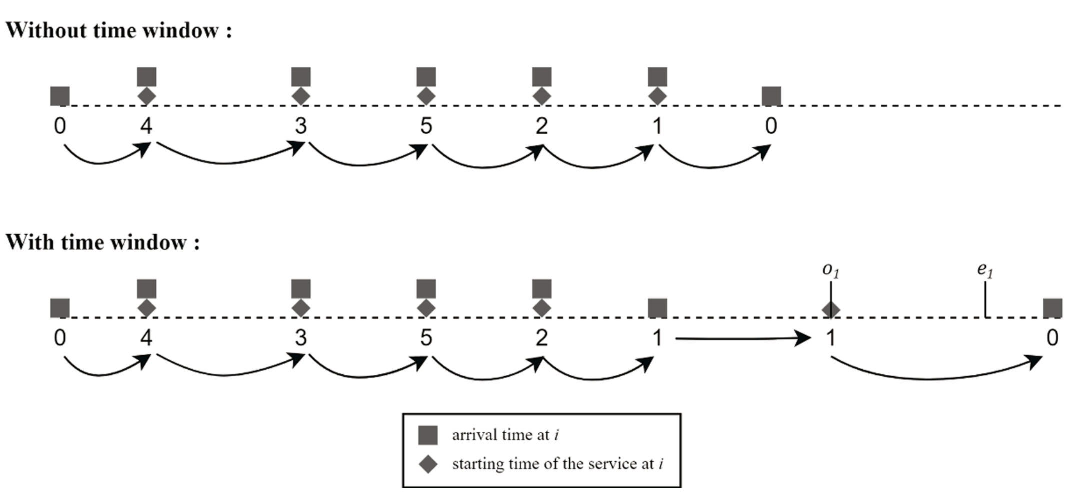

Fig. 8. The effect of time windows on the total completion times of a delivery mission.

VRPTWD. Table 6 presents the summary of comparison between these three problems, where z corresponds to the total cost, T is the total completion time of the delivery mission, $dist$ represent the total distance traveled by all vehicles, $dist_t$ is the total distance traveled by trucks, $dist_d$ is the total distance traveled by drones, $\#v$ is the total sets of truck-drone tandem deployed, $\#d$ presents the total sorties, and $\%d$ is the portion of customers served with drones. or clarity, all numbers are presented in the average value of the corresponding parameters.

Table 7
Impact of the width of time windows to the solutions.

n	w (min)	VRPTWD					VRPD				
		z	$dist$	$dist_t$	$dist_d$	$\#v$	z	$dist$	$dist_t$	$dist_d$	$\#v$
5	30	1.17	27.11	7.20	19.91	1.25	1.11	26.93	6.69	20.24	1.13
	120	1.28	31.21	7.69	23.52	1.13	1.26	29.99	7.69	22.30	1.13
10	30	1.87	46.59	11.25	35.34	1.83	1.76	42.06	10.71	31.34	1.25
	120	1.85	49.42	10.66	38.75	1.70	1.75	44.32	10.33	33.99	1.50
15	30	2.43	61.67	14.96	46.71	1.76	2.17	54.16	12.91	41.25	1.50
	120	2.28	59.26	14.15	45.11	1.63	2.16	55.08	12.70	42.38	1.00
20	30	2.88	76.95	18.44	58.50	1.85	2.52	58.79	15.47	43.32	1.38
	120	2.61	71.27	16.55	54.73	1.60	2.50	55.55	15.63	39.92	1.25
30	30	3.86	97.22	26.48	70.73	2.35	3.06	65.67	19.39	46.28	1.00
	120	3.79	91.52	25.57	65.95	1.79	3.31	68.35	21.33	47.02	1.13
40	30	4.94	122.98	34.54	88.44	2.88	3.71	72.45	24.34	48.12	1.00
	120	4.21	107.71	28.32	79.39	2.60	3.45	76.67	21.56	55.10	1.13
50	30	5.92	147.12	40.44	106.68	2.98	3.90	83.02	24.82	58.20	1.00
	120	5.25	129.56	35.94	93.63	2.83	4.10	83.70	26.47	57.23	1.00
Average	30	*3.30*	*82.81*	*21.90*	*60.90*	*2.13*	2.60	57.58	16.33	41.25	*1.18*
	120	3.04	77.14	19.84	57.30	1.89	*2.65*	*59.09*	*16.53*	*42.56*	1.16

Note: *italic* numbers indicate the highest value.

From the comparison between VRPTW and VRPTWD, the impact of deploying drones in a delivery mission with time windows can be observed. In Table 6, it is obvious that the presence of drones generally led to the declining total costs. Although the total distance traveled by all vehicles enlarges in the VRPTWD, a large portion of those distances are traveled by drones which possess the smaller transportation cost than trucks. Thus, it can be concluded that, even with the presence of customer time windows, deploying drones remains a beneficial option to improve the efficiency of a delivery mission. This observation is confirmed in Fig. 6, which shows that as the number of nodes grows larger, the total cost of VRPTW enlarges considerably when compared to the VRPD and VRPTWD.

Subsequently, the effect of customer time windows can be distinguished by comparing the results of VRPTWD with VRPD. Table 6 and Fig. 6 inform us that the total costs of VRPTWD for the same instances are generally higher than VRPD, even though the number of sorties executed in VRPTWD is generally higher. This can be explained by considering that the customer time windows may prevent the vehicles to travel through certain arcs i, j with lesser distance, since the nodes i and j might only be served during the similar time periods. On the other hand, Fig. 7 reveals another interesting insight that the completion time of solutions for VRPD is substantially faster than the solutions of VRPTWD and VRPTW. The presence of time windows forces the vehicles which arrive early to wait for the opening time of the customer nodes. As such, depending on the spread of the time windows within the working hours, the value of total completion time will always be close to the time windows of the customer served last ($T > \max_{j \in N}(o_j)$). For illustration, consider a simple TSP route with five customers that can be served within 8 h of working time, where the following optimal route *0-4-3-5-2-1-0* can finish the delivery mission in 5 h. Suppose that, under the presence of hard time windows constraints, node $\{1\}$ can only be served 6 h after the starting time. In this regard, the total completion time of the delivery mission will always be imposed to be more than 6 h since the vehicle must always wait for the opening time of node $\{1\}$, regardless of its arrival time at node $\{1\}$. This example can be visually illustrated in Fig. 8. Considering this, we note that optimizing drone routing problems with time windows should not consider the minimization of total completion time as in the classical FSTSP (Murray and Chu, 2015) and TSP-D (Agatz et al., 2018), but rather the minimization of total travel costs (Sacramento et al., 2019).

5.4.2. Impact of the width and density of time windows

In order to explain the impact of customer time windows better, we are also interested to see the effect of two features: the width (w) and the density ($\%TW$) of time windows. As explained in Section 5.1, the value of $\%TW$ was selected according to the set $\{25\%, 50\%, 75\%, 100\%\}$, while the value of w is selected from the set $\{30min, 120min\}$.

In terms of the width of time windows, Table 7 numerically presents the difference between the solutions for a class of instances with $w = 30$ and $w = 120$. From here, two main observations are derived. First, when facing a case with tighter time windows, the decision-maker tends to use a larger number of truck-drone tandems. This is based on a consideration that having more vehicles may assist the delivery process when several customer nodes need to be served at a tight period of time (i.e. both customers 1 and 2 must be served from 2.00 pm to 2.30 pm) and there are not enough time to serve them sequentially using a single vehicle. Correspondingly, the tighter time windows also increase the distance traveled by both trucks and drones. This phenomenon happens since tighter time windows may prevent the vehicles to use arcs with lower distance. As a result, instances with tighter time windows tend to suffer from higher total travel cost due to the relatively high travel distance of trucks. This can be proven by the additional observation of the VRPD solutions in Table 7. When the presence of time windows is relaxed, it turns out that the group of instances with $w = 120$ incur higher total cost and total distance than the group of instances with $w = 30$, which indicates the magnitude of the width of time windows to the characteristic of the optimal solutions.

Afterwards, the impact of time windows' density can be studied from

Table 8
Impact of the density of time windows to the solutions.

n	$\%TW$ (%)	VRPTWD					VRPD				
		z	$dist$	$dist_t$	$dist_d$	$\#v$	z	$dist$	$dist_t$	$dist_d$	$\#v$
Average	25	2.78	68.45	18.23	50.22	1.78	2.64	*58.98*	16.52	*42.46*	*1.18*
	50	3.03	76.89	19.96	56.93	2.00	*2.65*	58.16	*16.70*	41.47	1.11
	75	3.25	84.06	21.16	62.91	2.03	2.58	57.63	16.16	41.47	*1.18*
	100	*3.57*	*89.15*	*23.76*	*65.39*	*2.18*	2.58	57.42	16.13	41.29	*1.18*

Note: *italic* numbers indicate the highest value.

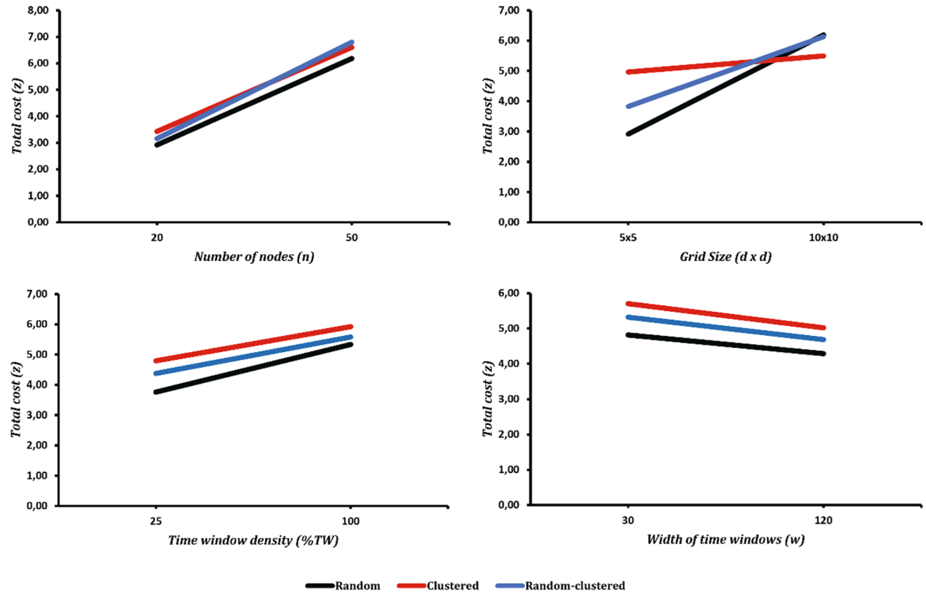

Fig. 9. The effect of customer distribution and problem parameters to the heuristic solutions.

Table 8. Similar to the width of time windows, as the number of customer nodes which have a time window enlarges, the decision-maker tend to use more vehicle sets since the possibility of having two or more nodes with similar time windows rises. As noted previously, having more sets of vehicles will assist in serving these customer nodes simultaneously. Moreover, it can also be seen that the higher density of time windows may result in a greater distance incurred by the vehicles in VRPTWD, while correspondingly, the total costs generally increase as well. On the contrary, the similar trends are not found in the results of VRPD, which shows the significance of time window density.

5.4.3. Impact of the distribution of customers

Another problem characteristic that might affect the generation of solutions in vehicle routing problems is the spatial distribution of customers. Following Uchoa et al. (2017), the distribution of customer nodes in vehicle routing-related instances can be classified into one of the following classes: (i) *random*, where the customer nodes are randomly located, (ii) *clustered*, where the location distribution of the customer nodes forms several clusters, and (iii) *random-clustered*, where a half of customer nodes are randomly distributed, while the other portion are clustered. In this regard, we are interested to study the effect of the customer distribution in VRPTWD.

In order to create instances where customer nodes are clustered, the presentation of Sacramento et al. (2019) is closely followed. The generation of instance starts with the creation of θ focal points, where the value of θ is set as a random number in the interval 1–5. Then, a permutation of all customers is created, and each customer is randomly assigned to one of the θ focal points. For each of these customers, its location (x, y) is generated with a normal distribution centered on the location of its corresponding focal point and with a standard deviation of 2 miles. Furthermore, for the random-clustered instances, half of the customers are assigned with the above steps, while the other half are randomly generated.

Using the above steps, we generate 48 additional instances to evaluate the effect of customer distribution. Each of these additional instances corresponds to a change in the following problem parameters: number of customer nodes (n), the size of grid area (d), the density of time windows ($\%TW$), and the width of time windows (w). Then, using the proposed VNS approach, all of these instances are solved, and the results are completely presented in Appendix 1 and Appendix 2.

Overall, similar to the conclusion from Sacramento et al. (2019) in the context of VRPD, our evaluation of the results shows that the distribution of customer nodes does not display a remarkable effect to the heuristic solutions. As seen in Fig. 9, it can be noted that as the value of n, d, $\%TW$, and w altered, the average objective values (z) of both random, clustered, and random-clustered instances shift to the same direction. Additionally, we also observe that the distribution of customer nodes does not incur a notable difference to the standard deviation of solutions produced by VNS, with the average values of z_{stdev} of each instance class are: 0.30 for random instances, 0.31 for clustered instances, and 0.26 for random-clustered instances. From here, we can conclude that there is no notable impact of the spatial distribution of customer nodes to the total travel costs of the delivery mission and the quality of solutions from VNS remain stable.

6. Conclusions

This study aims to extend the exploration of truck-drone cooperation by considering the constraint of time windows. The mathematical formulation of VRPD by Sacramento et al. (2019) is extended to develop the MIP formulation for VRPTWD. The objective function is to minimize the total cost of the delivery mission. In this model, hard time window constraints are considered, in which the customers must be visited by either truck or drone within their specified time windows. Besides, some realistic characteristics of the drone routing are also taken into account, such as the limitation of vehicle capacity, drone battery endurance, and setup times for launching and recollecting the drones in the rendezvous node. As the solution approach for the VRPTWD model, this study proposes a VNS-based heuristic with novel chromosome representation for solving various sizes of problem. Then, numerical experiments are conducted to prove the effectiveness and efficiency of the proposed VNS as well as to derive some interesting managerial insights from the model.

Though the result is promising, yet there are still some encouraging directions for future works. Future studies may extend the model to include other realistic variants of routing problem to develop a richer model. For instance, Gonzalez-R et al. (2020) and Poikonen and Golden (2020) considered a model with one tandem of truck and drone where the drone can serve multiple customer nodes in a single sortie. Considering the fast technological advancement nowadays, this relaxation of single-customer sortie seems to be a promising extension to be included in VRPTWD model. Another possible extension is to scale-up the size of our numerical experiments. In this study, we only have considered instances with size up to 50 customers due to the complexity of the considered problem. Naturally, extending this study for larger-size instances, and possibly evaluating the implementation of other algorithms from different domains (e.g. evolutionary computation) will be interesting for researchers in operational research and scientific computing.

CRediT authorship contribution statement

R.J. Kuo: Conceptualization, Supervision, Resources, Writing – review & editing. **Shih-Hao Lu:** Conceptualization, Supervision. **Pei-Yu Lai:** Conceptualization, Software, Investigation, Formal analysis, Writing – original draft. **Setyo Tri Windras Mara:** Methodology, Software, Data curation, Investigation, Formal analysis, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2021.116264.

References

Agárdi, A., Kovács, L., & Bányai, T. (2019). Vehicle routing in drone-based package delivery services. In *Solutions for Sustainable Development* (pp. 151–159). CRC Press.

Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science, 52*(4), 965–981. https://doi.org/10.1287/trsc.2017.0791

Avci, M. G., & Avci, M. (2019). An adaptive large neighborhood search approach for multiple traveling repairman problem with profits. *Computers and Operations Research, 111*, 367–385. https://doi.org/10.1016/j.cor.2019.07.012

Barmpounakis, E. N., Vlahogianni, E. I., & Golias, J. C. (2016). Unmanned Aerial Aircraft Systems for transportation engineering: Current practice and future challenges. *International Journal of Transportation Science and Technology, 5*(3), 111–122. https://doi.org/10.1016/j.ijtst.2017.02.001

Bouman, P., Agatz, N., & Schmidt, M. (2018). Dynamic programming approaches for the traveling salesman problem with drone. *Networks, 72*(4), 528–542. https://doi.org/10.1002/net.21864

Bräysy, O., & Gendreau, M. (2005a). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science, 39*(1), 104–118. https://doi.org/10.1287/trsc.1030.0056

Bräysy, O., & Gendreau, M. (2005b). Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science, 39*(1), 119–139. https://doi.org/10.1287/trsc.1030.0057

Braaten, S., Gjønnes, O., Hvattum, L. M., & Tirado, G. (2017). Heuristics for the robust vehicle routing problem with time windows. *Expert Systems with Applications, 77*, 136–147. https://doi.org/10.1016/j.eswa.2017.01.038

Bryan, V., 2014. *Drone delivery: DHL 'parcelcopter' flies to German isle*. Retrieved from http://www.reuters.com/article/2014/09/24/us-deutsche-post-drones-idUSKCN0HJ1ED20140924/. Accessed March 5, 2021.

Chen, S., Chen, R., Wang, G. G., Gao, J., & Sangaiah, A. K. (2018). An adaptive large neighborhood search heuristic for dynamic vehicle routing problems. *Computers and Electrical Engineering, 67*, 596–607. https://doi.org/10.1016/j.compeleceng.2018.02.049

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. https://doi.org/10.1016/j.apenergy.2019.03.117

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. https://doi.org/10.1016/j.cor.2020.105004

Coutinho, W. P., Battarra, M., & Fliege, J. (2018). The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Computers & Industrial Engineering, 120*, 116–128. https://doi.org/10.1016/j.cie.2018.04.037

Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations Research, 6*(6), 791–812. https://doi.org/10.1287/opre.6.6.791

de Freitas, J. C., & Penna, P. H. V. (2018). A randomized variable neighborhood descent heuristic to solve the flying sidekick traveling salesman problem. *Electronic Notes in Discrete Mathematics, 66*, 95–102. https://doi.org/10.1016/j.endm.2018.03.013

de Freitas, J. C., & Penna, P. H. V. (2020). A variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research, 27*(1), 267–290. https://doi.org/10.1111/itor.12671

Dell'Amico, M., Montemanni, R., & Novellani, S. (2021). Drone-assisted deliveries: New formulations for the flying sidekick traveling salesman problem. *Optimization Letters, 15*(5), 1617–1648. https://doi.org/10.1007/s11590-019-01492-z

Desaulniers, G., Madsen, O. B., & Ropke, S. (2014). Chapter 5: The vehicle routing problem with time windows. In *Vehicle Routing: Problems, Methods, and Applications* (second ed., pp. 119–159). Society for Industrial and Applied Mathematics.

Elshaer, R., & Awad, H. (2020). A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering, 140*, 106242. https://doi.org/10.1016/j.cie.2019.106242

Es Yurek, 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. https://doi.org/10.1016/j.trc.2018.04.009

Etherington, D., 2017. *Check out the real world version of the Mercedes-Benz drone delivery van*. Retrieved from https://techcrunch.com/2017/01/10/check-out-the-real-world-version-of-the-mercedes-benz-drone-delivery-van/. Accessed March 10, 2021.

Ferreira, H. S., Bogue, E. T., Noronha, T. F., Belhaiza, S., & Prins, C. (2018). Variable neighborhood search for vehicle routing problem with multiple time windows. *Electronic Notes in Discrete Mathematics, 66*, 207–214. https://doi.org/10.1016/j.endm.2018.03.027

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. https://doi.org/10.1016/j.trc.2020.02.030

Gross, D., 2013. *Amazon's Drone Delivery: How Would it Work?* Retrieved from http://www.cnn.com/2013/12/02/tech/innovation/amazon-drones-questions/. Accessed April 12, 2021.

Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2018). On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies, 86*, 597–621. https://doi.org/10.1016/j.trc.2017.11.015

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. https://doi.org/10.1016/j.trc.2018.03.025

Hansen, P., & Mladenović, N. (2018). In *Handbook of heuristics* (pp. 759–787). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-07124-4_19.

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. https://doi.org/10.1016/j.ejor.2007.08.048

Jarboui, B., Derbel, H., Hanafi, S., & Mladenović, N. (2013). Variable neighborhood search for location routing. *Computers & Operations Research, 40*(1), 47–57. https://doi.org/10.1016/j.cor.2012.05.009

Khoufi, I., Laouiti, A., & Adjih, C. (2019). A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles. *Drones, 3*(3), 66. https://doi.org/10.3390/drones3030066

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. https://doi.org/10.1016/j.cie.2019.01.020

Koç, Ç. (2016). A unified-adaptive large neighborhood search metaheuristic for periodic location-routing problems. *Transportation Research Part C: Emerging Technologies, 68*, 265–284. https://doi.org/10.1016/j.trc.2016.04.013

Labadie, N., Mansini, R., Melechovský, J., & Calvo, R. W. (2012). The team orienteering problem with time windows: An lp-based granular variable neighborhood search. *European Journal of Operational Research, 220*(1), 15–27. https://doi.org/10.1016/j.ejor.2012.01.030

Lan, B. (2020). *Traveling salesman problem with time windows and drones (TSPTWD)* (Doctoral dissertation, Iowa State University). https://doi.org/10.31274/etd-20200902-84.

Lee, S., & Prabhu, V. V. (2016). Just-in-time delivery for green fleets: A feedback control approach. *Transportation Research Part D: Transport and Environment, 46*, 229–245. https://doi.org/10.1016/j.trd.2016.04.005

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., & Laporte, G. (2020). Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies, 120*, 102762. https://doi.org/10.1016/j.trc.2020.102762

Marinelli, M., Caggiani, L., Ottomanelli, M., & Dell'Orco, M. (2018). En route truck–drone parcel delivery for optimal vehicle routing strategies. *IET Intelligent Transport Systems, 12*(4), 253–261. https://doi.org/10.1049/iet-its.2017.0227

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research, 24*(11), 1097–1100. https://doi.org/10.1016/S0305-0548(97)00031-2

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. https://doi.org/10.1016/j.trc.2015.03.005

Murray, C. C., & Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies, 110*, 368–398. https://doi.org/10.1016/j.trc.2019.11.003

Nagata, Y., Bräysy, O., & Dullaert, W. (2010). A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research, 37*(4), 724–737. https://doi.org/10.1016/j.cor.2009.06.022

Otto, A., Agatz, N., Campbell, J., Golden, B., & Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks, 72*(4), 411–458. https://doi.org/10.1002/net.21818

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2010). Variable neighborhood search for the dial-a-ride problem. *Computers & Operations Research, 37*(6), 1129–1138. https://doi.org/10.1016/j.cor.2009.10.003

Peterson, K., Dektas, M., 2017. *UPS tests residential delivery via drone launched from atop package car*. Online accessed at https://www.pressroom.ups.com/pressroom/ContentDetailsViewer.page?ConceptType=PressReleases&id=1487687844847-162/. Accessed March 6, 2021.

Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research, 34*(8), 2403–2435. https://doi.org/10.1016/j.cor.2005.09.012

Poikonen, S., & Campbell, J. F. (2021). Future directions in drone routing research. *Networks, 77*(1), 116–126. https://doi.org/10.1002/net.21982

Poikonen, S., & Golden, B. (2020). Multi-visit drone routing problem. *Computers & Operations Research, 113*, 104802. https://doi.org/10.1016/j.cor.2019.104802

Poikonen, S., Golden, B., & Wasil, E. A. (2019). A branch-and-bound approach to the traveling salesman problem with a drone. *INFORMS Journal on Computing, 31*(2), 335–346. https://doi.org/10.1287/ijoc.2018.0826

Poikonen, S., Wang, X., & Golden, B. (2017). The vehicle routing problem with drones: Extended models and connections. *Networks, 70*(1), 34–43. https://doi.org/10.1002/net.21746

Popović, D., Kovač, M., & Bjelić, N. A. (2019). May). MIQP model for solving the vehicle routing problem with drones. In *Proceedings of 4th Logistics International Conference–LOGIC* (pp. 52–62).

Prodhon, C., & Prins, C. (2016). Metaheuristics for vehicle routing problems. In *Metaheuristics* (pp. 407-437). Springer, Cham.

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. https://doi.org/10.1007/978-3-319-67308-0_56.

Pugliese, L. D. P., Guerriero, F., & Macrina, G. (2020). Using drones for parcels delivery process. *Procedia Manufacturing, 42*, 488–497. https://doi.org/10.1016/j.promfg.2020.02.043

Qiu, Y., Wang, L., Xu, X., Fang, X., & Pardalos, P. M. (2018). A variable neighborhood search heuristic algorithm for production routing problems. *Applied Soft Computing, 66*, 311–318. https://doi.org/10.1016/j.asoc.2018.02.032

Ribeiro, G. M., Desaulniers, G., Desrosiers, J., Vidal, T., & Vieira, B. S. (2014). Efficient heuristics for the workover rig routing problem with a heterogeneous fleet and a finite horizon. *Journal of Heuristics, 20*(6), 677–708. https://doi.org/10.1007/s10732-014-9262-1

Rojas Viloria, D., Solano-Charris, E. L., Muñoz-Villamizar, A., & Montoya-Torres, J. R. (2021). Unmanned aerial vehicles/drones in vehicle routing problems: A literature review. *International Transactions in Operational Research, 28*(4), 1626–1657. https://doi.org/10.1111/itor.12783

Ropke, S., & Pisinger, D. (2006). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research, 171*(3), 750–775. https://doi.org/10.1016/j.ejor.2004.09.004

Rose, C. (2013). *Amazon's Jeff Bezos looks to the future*. Retrieved from http://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/. Accessed March 1, 2021.

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. https://doi.org/10.1016/j.trc.2019.02.018

Schermer, D., Moeini, M., & Wendt, O. (2018, March). Algorithms for solving the vehicle routing problem with drones. In *Asian Conference on Intelligent Information and Database Systems* (pp. 352-361). Springer, Cham. https://doi.org/10.1007/978-3-319-75417-8_33.

Schermer, D., Moeini, M., & Wendt, O. (2019a). A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Computers & Operations Research, 109*, 134–158. https://doi.org/10.1016/j.cor.2019.04.021

Schermer, D., Moeini, M., & Wendt, O. (2019b). A matheuristic for the vehicle routing problem with drones and its variants. *Transportation Research Part C: Emerging Technologies, 106*, 166–204. https://doi.org/10.1016/j.trc.2019.06.016

Silberholz, J., Golden, B., Gupta, S., & Wang, X. (2019). Computational comparison of metaheuristics. In *Handbook of Metaheuristics* (pp. 581–604). Cham: Springer.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research, 35*(2), 254–265. https://doi.org/10.1287/opre.35.2.254

Tilley, A., 2016. *UPS experiments with drone delivery in partnership with Zipline*. Retrieved from https://www.forbes.com/sites/aarontilley/2016/05/09/ups-experiments-with-drone-delivery-in-partnership-with-zipline/#58f1c1d145a4/. Accessed May 29, 2021.

Tu, P. A., Dat, N. T., & Dung, P. Q. (2018). December). Traveling salesman problem with multiple drones. In *Proceedings of the Ninth International Symposium on Information and Communication Technology* (pp. 46–53). https://doi.org/10.1145/3287921.3287932

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., & Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research, 257*(3), 845–858. https://doi.org/10.1016/j.ejor.2016.08.012

Ulmer, M. W., & Thomas, B. W. (2018). Same-day delivery with heterogeneous fleets of drones and vehicles. *Networks, 72*(4), 475–505. https://doi.org/10.1002/net.v72.410.1002/net.21855

Vahdani, B., Veysmoradi, D., Noori, F., & Mansour, F. (2018). Two-stage multi-objective location-routing-inventory model for humanitarian logistics network design under uncertainty. *International Journal of Disaster Risk Reduction, 27*, 290–306. https://doi.org/10.1016/j.ijdrr.2017.10.015

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research, 40*(1), 475–489. https://doi.org/10.1016/j.cor.2012.07.018

Wang, X., Poikonen, S., & Golden, B. (2017). The vehicle routing problem with drones: Several worst-case results. *Optimization Letters, 11*(4), 679–697. https://doi.org/10.1007/s11590-016-1035-3

Wang, Z., & Sheu, J. B. (2019). Vehicle routing problem with drones. *Transportation Research Part B: Methodological, 122*, 350–364. https://doi.org/10.1016/j.trb.2019.03.005

Wohlsen, M. (2014). The Next Big Thing You Missed: Amazon's Delivery Drones Could Work—They Just Need Trucks. *Wired: Business, Jun, 10*.

Zhen, L.u., Ma, C., Wang, K., Xiao, L., & Zhang, W. (2020). Multi-depot multi-trip vehicle routing problem with time windows and release dates. *Transportation Research Part E: Logistics and Transportation Review, 135*, 101866. https://doi.org/10.1016/j.tre.2020.101866