Expert Systems With Applications 221 (2023) 119777

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Applying NSGA-II to vehicle routing problem with drones considering makespan and carbon emission

R.J. Kuo[a,*], Evan Edbert[b], Ferani E. Zulvia[c], Shih-Hao Lu[d]

[a] Department of Industrial Management, National Taiwan University of Science and Technology, No. 43, Section 4, Kee-Lung Road, Taipei 106, Taiwan
[b] Department of Product and Manufacturing Engineer, Top Union Electronics Corporation, No. 480, Nioupi East Rd., Hsinchu City 300, Taiwan
[c] School of Industrial Engineering and Engineering Management, Mapua University, No. 658, Muralla Street, Intramuros, Manila 1002, Philippines
[d] Department of Business Administration, National Taiwan University of Science and Technology, No. 43, Section 4, Kee-Lung Road, Taipei, Taiwan

ARTICLE INFO

Keywords:
Vehicle routing problem with drones
Makespan
Carbon emission
NSGA-II

ABSTRACT

Drones or unmanned aerial vehicles (UAV) are aerial vehicles capable of sustained flight independent of a human operator onboard. At present, many companies are developing their own delivery system with drones, in which the usage of drones in delivery is expected to develop substantially in the next few years. Although the speediness of drone delivery has been investigated, its feasibility remains in question and needs to be studied further. In addition, despite the advantages of using drones for delivery from an economic point of view, its environmental benefits must be tested. This study proposes the vehicle routing problem with drones (VRPD) model aimed at minimizing the makespan and carbon emissions of the delivery route. In this model, each truck is equipped with one drone to collaborate in delivery. In particular, this study aims to propose the mathematical formulation for the VRPD with two objectives—solve it with the non-dominated sorting genetic algorithm II (NSGA-II) and test the proposed algorithm with different-scale problems. Hypervolume and spacing are employed to check the quality of the solutions produced through the algorithm. The result shows that the algorithm can produce a remarkable result with good hypervolume and spacing values. The benefits of drones to reduce makespan and carbon emissions are tested through a significance test. Furthermore, this study finds that the difference when using drones is significant.

1. Introduction

Drones or unmanned aerial vehicles (UAV) are aerial vehicles capable of sustained flight independence of a human operator onboard (Bento, 2008). Drones have become a center of attention, given their advantages, such as in filming, war, agriculture, surveillance, and even deliveries. Jeff Bezos, CEO and founder of Amazon, disclosed in the “60 min” interview in December 2013 that drones could be used to speed up the delivery of packages to customers (Andrea, 2014). His statements have instilled curiosity and stimulated more discussion around the world. Besides Amazon, several companies also attempted to develop their own delivery systems using drones. DHL launched its fully-automated and improved parcelcopter to deliver medication to the North Sea Island of Juist in 2014 (GmbH, 2014). Google also tested its drone deliveries in Project Wing trials to deliver defibrillators in Australia in 2014 (Stewart, 2014).

The usage of drones in shipping is anticipated to develop substantially in the next few years. Due to the pointy boom in online shopping, buyers have been stressed about the speed of shipping services (Macrina et al., 2020). However, drones also have some limitations that must be addressed, one of which is their battery life. Nevertheless, delivering with both trucks and drones has become an efficient way of making the best use of their advantages (Wohlsen, 2014). Optimizing the usage of drones in delivery can gain many benefits, such as reducing operational costs and speeding up deliveries.

Murray and Chu (2015) and Poikonen et al. (2019) combined trucks and drones to deliver goods for traveling sales problems (TSPs). Later, Sacramento et al. (2019) proposed the vehicle routing problem with the drones (VRPD) model, which has been widely studied. In particular, Chung et al. (2020) and Moshref-Javadi and Winkenbach (2021) provided a comprehensive survey for VRPD.

Although much research has been conducted to study the problems in drone delivery, there are still some issues that must be solved. For instance, most of the current research on drones only considered one

* Corresponding author.
E-mail address: rjkuo@mail.ntust.edu.tw (R.J. Kuo).

https://doi.org/10.1016/j.eswa.2023.119777
Received 26 March 2022; Received in revised form 2 August 2022; Accepted 27 February 2023
Available online 5 March 2023

objective. However, in real applications, more than one objective should be considered simultaneously. Therefore, this study intends to focus on the possible gains of utilizing drones in the vehicle routing problem and propose a mathematical model for multi-objective vehicle routing problems with drones. In addition, this study aims to achieve lower delivery costs and carbon emissions. Further, two objectives considered in this study include minimizing the makespan and the total carbon emission of the routes. Environment issues have become a serious problem for most companies. If green delivery can be taken into account, logistics companies can provide better service for their customers. Thus, a non-dominated sorting genetic algorithm is employed to solve the proposed problem, and different-scale problems are applied to test the proposed model to verify its feasibility.

The rest of this study is organized as follows. Section 2 presents the literature review for the related research, while Section 3 explains the proposed mathematical model. The experimental results are presented in Section 4. Finally, the concluding remarks are discussed in Section 5.

2. Literature review

2.1. Vehicle routing problem

Murray and Chu (2015) created the famous model that combined the usage of trucks and drones in delivery for the flying sidekick traveling salesman problem (FSTSP). In this problem, they created a model to find an optimal delivery route combining the delivery truck and UAV. They also developed the parallel drone scheduling TSP model (PDSTSP). The second model states different conditions where customers are located within the UAV's flying range; meanwhile, in the first model, all customers are located far from the delivery centers. Poikonen et al. (2019) proposed a branch-and-bound heuristic approach to solve the TSP with drones. Sacramento et al. (2019) also proposed an adaptive large neighborhood search metaheuristic for the VRPD, slightly similar to the FSTSP (Murray & Chu, 2015) but used capacitated multi-truck and time limit constraints to minimize total transportation cost. Kitjacharoenchai et al. (2020) used formulated mixed integer programming (MIP) for small-size problems and designed two adequate heuristic algorithms to solve large-size problems—that is, the drone truck route construction (DTRC) and large neighborhood search (LNS).

Poikonen and Golden (2020) considered a *k*-multi-visit drone routing problem (*k*-MVDRP), where each drone is launched from the truck with one or more packages to deliver to customers and employs the route, transform, shortest path (RTS) heuristic. Kyriakakis et al. (2022) introduced an electric VRPD (EVRPD) by combining electric ground vehicles (EVs) with UAVs. The objective of this routing problem is to minimize the total energy consumption. Further, Wang and Sheu (2019) proposed a mixed integer programming model and developed a branch-and-price algorithm for VRPD. Karak and Abdelghany (2019) also presented a mathematical model for the hybrid vehicle-drone routing problem for pick-up and delivery services. A method was developed which extends the classic Clarke and Wright algorithm to solve the model. In addition, Huang et al. (2022) employed the ant colony optimization algorithm to solve VRPD.

The FSTSP and PDSTSP models formulated by Murray and Chu (2015) were aimed at minimizing the makespan of the route, in which extensive synchronization among the truck and the drone is required in FSTSP. The maximum traveling time of all the routes designed for fulfilling the demands is the definition of the makespan for finishing all services. Boysen et al. (2018) proposed the drone scheduling problem (DSP) to determine the optimal timetable of drones such that the makespan is minimized. In particular, they proposed two diverse MILPs for solving the DSP. Schermer et al. (2019) formulated the VRPD as a MILP and presented a few sets of valid inequalities (VIEQ). They also proposed two MILP formulations for the drone arrangement problem. In addition, Chung et al. (2020) and Moshref-Javadi and Winkenbach (2021) provided a comprehensive review of drone and drone-truck combined operations.

The classic VRP model assumes the usage of trucks for deliveries and considers the objectives regarding time or distance traveled. However, with the rising environmental concerns, many alternatives were sought to achieve environment-friendly VRP, called the green VRP (G-VRP). Social awareness of the high environmental impact caused by transportation, the depletion of petroleum and oil reserves, and the quest for energy security have driven governments around the world to strengthen legislation on the use of environmentally unfriendly vehicles and provide incentives for adopting alternative fuel vehicles (AFV) (Pollet et al., 2012). Chiang et al. (2019) studied the influence of UAVs on CO_2 emission and expenditure and proposed a mixed-integer green routing model for UAVs, in which a genetic algorithm was employed. They found that not only optimally routing and delivering packages with UAVs are cost-effective but also help in energy conservation and carbon emission reduction.

The vehicle routing problem with time window (VRPTW) is a generalization of the well-known VRP. It can be viewed as a combination of vehicle routing and scheduling problems. In VRPTW, the fleet of vehicles must serve all customers to minimize the cost, subject to vehicle capacity and service time restrictions (Elabib et al., 2002). In VRPTW, the service for each customer must start within an associated time window, and the vehicle must remain at the customer's location during service (Cordeau et al., 1999). There are two types of time windows in VRPTW—soft time windows that can be violated at a cost and hard time windows that do not allow for a vehicle arriving at a customer after the latest time to begin service. In VRPTW, the usage of soft time windows has one disadvantage; violation of time windows does not directly incur any penalty cost but may lead to a drop in customer satisfaction level and profit loss in the long term (Tang et al., 2009). Zhang et al. (2013) proposed a stochastic programming model to formulate on-time shipment delivery problems with soft time windows under travel and service time uncertainties and minimize carriers' total cost. In particular, they used an iterated tabu search heuristic incorporating a route reduction mechanism. Bulhões et al. (2018) considered a vehicle routing problem that minimized the cost but was subject to service level constraints and solved it with a branch-and-price algorithm.

In VRP, sometimes there is a need to have more than one objective function to broaden traditional academic problems in order to enhance their viable application, generalize traditional issues, and observe actual cases in which the goals had been surely prominent through the decision-makers and were committed to a particular real-life problem or application (Jozefowiez et al., 2008). Molina et al. (2014) proposed a multi-objective VRPTW based on the Tchebycheff methods with a heterogeneous fleet and three objective functions— that is, to minimize the total internal costs, CO2 emissions, and the emission of air pollutants such as NOx. They also developed an algorithm based on the C&W savings heuristic to solve the model when time windows are not considered. Zhang et al. (2019) proposed a multi-objective VRP with flexible time windows, where the total distribution costs (including travel and fixed vehicle costs) are minimized and the overall customer satisfaction is maximized. A solution strategy based on ant colony optimization and three mutation operators was also proposed, incorporating the concept of Pareto optimality for multi-objective optimization. The performance was evaluated using the well-known benchmark, Solomon's problems. Han et al. (2020) proposed a VRPTW with the drone model to minimize the weighted sum of the overall energy consumption of the trucks, total energy consumption of the drones, and the total number of trucks. The problem is solved using an improved artificial bee colony algorithm, and a novel scout bee strategy is presented to enhance the global search capabilities of the proposed algorithm. In addition, Kuo et al. (2022) also proposed a novel VRPTW model with drones.

2.2. Multi-objective evolutionary algorithm

MOEAs are an expanding research area in the evolutionary computation field. The first MOEA was Schaffer's vector evaluated genetic algorithm proposed in 1985 (Schaffer, 1985). Subsequently, numerous EA-based approaches to solving multi-objective optimization problems (MOPs) have appeared in the literature. In this research, the multi-objective genetic algorithm proposed by Deb et al. (2002) will be employed.

The non-dominated sorting genetic algorithm II (NSGA-II) was first proposed by Deb et al. (2002) to improve and solve the issues of the non-dominated sorting genetic algorithm (NSGA). Over the years, the main criticisms of the first NSGA proposed by Srinivas and Deb (1994) centered on the high computational complexity of non-dominated sorting, lack of elitism, and the need for specifying the parameter σ_{share}.

In NSGA-II, there is a procedure called the non-dominated sorting approach used to sort the population by following the non-dominated rule. In implementing the non-dominated sorting process, the chromosomes' objective function values are compared to each other; hence, this process requires $O(MN)$ comparisons for every solution, in which M is the number of objectives. In addition, Deb et al. (2002) replaced the sharing function method in NSGA with a crowded-comparison method that eliminates the abovementioned difficulties to a few extents. The new approach does not require user-defined parameters to maintain diversity among population members. However, it makes use of a density-estimation metric and the crowded-comparison operator. The density-estimation value of solutions surrounding a specific solution in the population is calculated through the average distance of two points on both aspects of this point alongside the objectives. The crowded-comparison operator guides the selection process at the various stages of the algorithm closer to a uniformly spread-out Pareto-optimal front.

3. Methodology

This section defines the VRPD model and proposes the mathematical formulation. The delivery is done using a fleet of the same kind of delivery trucks where each truck is equipped with a single drone. It is only done once for each customer and can be done by either truck or drone. The trucks and the drone will coordinate in doing the delivery work. Some assumptions are made for the proposed model to minimize two objectives: the makespan of the delivery and the total carbon emission of the route.

3.1. Mathematical formulation for the VRPD

The mathematical formulation of this model is an extension of the mixed integer programming (MIP) formulation by (Murray & Chu, 2015) and the VRPD model proposed by Sacramento et al. (2019). This study aims to minimize both the total makespan and the total carbon emissions of the route. The carbon emission calculation method is adopted from Chiang et al. (2019), in which the carbon emission of trucks is calculated based on the total distance traveled and carbon emission by the drones' charging station. The truck and the drone must also cooperate during the delivery process.

3.2. Notations

The notations for sets for this formulation are listed as follows:

- T = The set of all trucks = {0, 1, …, v}
- C = The set of all customers = {1, 2, …, n}
- V = The set of starting and returning points, 0 and n + 1, both represent the same depot = {0, 1, …, n, n + 1}
- V_D = The subset $V\_D \subseteq C$ represents the customers that the drone can serve
- V_L = The subset of the nodes that the drone can be launched from = {0, 1, …, n}
- V_R = The subset of the nodes that the drone can be returned to = {1, 2, …, n + 1}

The parameters required for the formulation are listed as follows:

- t_{ij}, t'_{ij} = The travel time of the truck and the drone from node i to node j
- $a_i^v, a_j^{\prime v}$ = The arrival time of the truck at node i and the arrival time of the drone at node j
- d_{ij} = The distance from node i to node j
- d_{ij}^{UAV} = The distance traveled by UAV from node i to node j
- TW_i^S = Start of the time window of customer i
- TW_i^E = End of the time window of customer i
- EET_i = Endurable earliness time – the earliest service time that customer i can endure when a service starts earlier than TW_i^S
- ELT_i = Endurable lateness time – the latest service time that customer i can endure when a service starts later than TW_i^E
- q_i = Demand of node i
- Q = The capacity of the truck
- s_i = The service time of node i
- s_L = The setup time required for launching the drone
- s_R = The setup time required for receiving the drone
- E = The flight endurance of the battery of the drone
- M = Given a large value
- S_i = Service level for each customer
- WAER = Weighted average emission rate of the vehicle
- PGFER = The amount of CO_2 emitted at power generation facilities per watt-hour (Wh) used by UAVs
- AER = Average energy requirement of UAVs in Wh per mile
- τ = Maximum arrival time of all trucks on all routes

The decision variables in the formulation are listed as follows:

- b_{ij}^v = Truck v travels from node i to node j; $i \neq j$, A binary variable that states whether arc (i,j) is used by truck v
- y_{ijk}^v = Drone v (with truck v) launches from node i, travels to node j (visiting a customer without a truck), and returns to a truck or the ending depot at node k
- B_{ij}^v = Binary variables that indicate whether node i is served before but not necessarily adjacent to node j in the route of truck v
- u_i^v = Integer variables with a lower bound of 0 that state the position of node i in the route of truck v

3.3. Mathematical formulation

The mathematical formulation of this VRPD can be presented as follows:

Minimize

$$\tau \tag{1}$$

Minimize

$$\sum_{v\in T}\sum_{\substack{i\in V_L\\ i\neq j}} WAER \times d_{ij} \times b_{ij}^v + \sum_{v\in T}\sum_{\substack{i\in V_L\\ i\neq j}}\sum_{\substack{k\in V_R\\ j\neq k}} PGFER \times AER \times (d_{ij}^{UAV} + d_{jk}^{UAV}) \times y_{ijk}^v \tag{2}$$

Subject to:

$$\sum a_{n+1}^v \leq \tau \forall v \in T \tag{3}$$

$$\sum_{v\in T}\sum_{\substack{i\in V_L\\ i\neq j}}b_{ij}^{v}+\sum_{v\in T}\sum_{\substack{i\in V_L\\ i\neq j}}\sum_{\substack{k\in V_R\\ j\neq k}}y_{ijk}^{v}=1\forall j\in C \tag{4}$$

$$\sum_{j\in V_R}b_{0j}^{v}\leq 1\forall v\in T \tag{5}$$

$$\sum_{i\in V_L}b_{i,n+1}^{v}\leq 1\forall v\in T \tag{6}$$

$$b_{0,n+1}^{v}=0\forall v\in T \tag{7}$$

$$\sum_{\substack{i\in V_L\\ i\neq j}}b_{ij}^{v}-\sum_{\substack{k\in V_R\\ j\neq k}}b_{jk}^{v}=0\forall v\in T,j\in C \tag{8}$$

$$u_i^v-u_j^v+1\leq(n-1)\left(1-b_{ij}^{v}\right)\forall v\in T,i\in V_L,j\in V_R,i\neq j \tag{9}$$

$$u_j^v\leq M\sum_{i\in V_L}b_{ij}^{v}\forall v\in T,j\in V_R,i\neq j \tag{10}$$

$$\sum_{j\in C}(\sum_{k\in V_R}q_jb_{jk}^{v}+\sum_{i\in V_L}\sum_{k\in V_R}q_jy_{ijk}^{v})\leq Q\forall v\in T \tag{11}$$

$$\sum_{\substack{j\in C\\ i\neq j}}\sum_{\substack{k\in V_R\\ j\neq k}}y_{ijk}^{v}\leq 1\forall v\in T,i\in V_L \tag{12}$$

$$\sum_{\substack{i\in V_L\\ i\neq j}}\sum_{\substack{j\in C\\ j\neq k}}y_{ijk}^{v}\leq 1\forall v\in T,k\in V_R \tag{13}$$

$$y_{ijk}^{v}\leq b_{ik}^{v}\forall i\in V_L,j\in C,k\in V_R,v\in T,i\neq k,j\neq k \tag{14}$$

$$a_0^v=0\forall v\in T \tag{15}$$

$$a_0^{'v}=0\quad\forall v\in T \tag{16}$$

$$a_i^v+t_{ik}+s_i+s_L\left(\sum_{\substack{j\in C\\ i\neq j}}\sum_{k\in V_R}y_{ijk}^{v}\right)+s_R\left(\sum_{\substack{i\in V_L\\ i\neq k}}\sum_{j\in C}y_{ijk}^{v}\right)\leq a_k^v+M\left(1-b_{ik}^{v}\right) \tag{17}$$

$$\forall v\in T,i\in V_L,k\in V_R$$

$$a_i^v+t'_{ij}+s_L-M\left(1-\sum_{k\in V_R}y_{ijk}^{v}\right)\leq a_j^{'v}\forall v\in T,i\in V_L,j\in C \tag{18}$$

$$a_j^{'v}+t'_{jk}+s_j+s_R-M\left(1-\sum_{i\in V_L}y_{ijk}^{v}\right)\leq a_k^{'v}\forall v\in T,j\in C,k\in V_R \tag{19}$$

$$a_i^{'v}\geq a_i^v-M(1-\sum_{\substack{j\in C\\ i\neq j}}\sum_{\substack{k\in V_R\\ j\neq k}}y_{ijk}^{v})\forall v\in T,i\in V_L \tag{20}$$

$$a_i^{'v}\leq a_i^v+M(1-\sum_{\substack{j\in C\\ i\neq j}}\sum_{\substack{k\in V_R\\ j\neq k}}y_{ijk}^{v})\forall v\in T,i\in V_L \tag{21}$$

$$a_k^{'v}\geq a_k^v-M(1-\sum_{\substack{i\in V_L\\ i\neq k}}\sum_{\substack{j\in C\\ j\neq k}}y_{ijk}^{v})\forall v\in T,k\in C \tag{22}$$

$$a_k^{'v}\leq a_k^v+M(1-\sum_{\substack{i\in V_L\\ i\neq k}}\sum_{\substack{j\in C\\ j\neq k}}y_{ijk}^{v})\forall v\in T,k\in C \tag{23}$$

$$E+M\left(1-\sum_{\substack{j\in C\\ j\neq k}}y_{ijk}^{v}\right)\geq a_k^{'v}-a_i^{'v}\forall v\in T,i\in V_L,k\in V_R \tag{24}$$

$$\left(u_i^v-u_j^v\right)\leq MB_{ij}^{v}\forall v\in T,i\in V_L,j\in C \tag{25}$$

$$\left(u_i^v-u_j^v\right)\geq M\left(B_{ij}^{v}-1\right)+1\forall v\in T,i\in V_L,j\in C \tag{26}$$

$$a_k^{'v}-M\left(3-\sum_{j\in C}y_{ijk}^{v}-\sum_{l\in C}\sum_{m\in V_R}y_{blm}^{v}-B_{ib}^{v}\right)\leq a_b^{'v} \tag{27}$$

$$\forall v\in T,i\in V_L,k\in V_R,b\in C$$

$$y_{0jk}^{v}\leq\sum_{\substack{i\in V_L\\ i\neq k\\ i\neq j}}b_{ik}^{v}\forall j\in C,k\in V_R,v\in T \tag{28}$$

$$\sum_{i\in N}S_i(t)\geq S \tag{29}$$

$$b_{ij}^{v}\in\{0,1\}\forall v\in T,i,j\in V \tag{30}$$

$$y_{ijk}^{v}\in\{0,1\}\forall v\in T,i\in V_L,j\in C,k\in V_R \tag{31}$$

$$B_{ij}^{v}\in\{0,1\}\forall v\in T,i\in V_L,j\in C \tag{32}$$

$$u_i^v,a_i^v,a_i^{'v}\geq 0\forall v\in T,i\in V \tag{33}$$

The objective function (1) tries to minimize the makespan using constraints (3) while the other (2) attempts to minimize the total carbon emission, where WAER is the weighted average emission rate of the vehicle, PGFER is the amount of CO_2 emitted at power generation facilities per watt-hour (Wh) used by UAVs, and AER is the average energy requirement of UAVs in Wh per mile. The WAER is set to 1.2603 kg per mile, the PGFER to 3.773×10^{-4} kg per Wh, and the AER to 3.3333 Wh per mile (Chiang et al., 2019). The purpose of constraint (4) is to ensure that each customer will be served exactly once, either by the truck or the drone. Constraint (5) is used to ensure that each truck is used at least once and must depart from the depot, and all trucks must return to the depot where they are constrained by Constraint (6). Constraint (7) is used to restrict the truck movement from depot to depot. Constraint (8) defines the flow conservation of the truck where each truck can only visit the customer once and depart from the same point. Constraints (9) and (10) eliminate the subtour for the truck. It refers to where customer locations are visited on truck routes. The capacity constraint for the truck is provided by Constraint (11). Constraints (12) and (13) set the drone that can be launched and returned once from each node. Constraint (14) ensures that the drone is launched from node i, served node j, and returned to node k. Then, the truck travels from node i to node k. This constraint ensures that the launched and returned points of the drone are at the front and back positions of the service point.

Constraints (15) and (16) set the arrival time for the truck and the drone at the beginning of each route. Constraint (17) defines the arrival time constraint for the truck movement. Constraints (18) and (19) define the arrival time constraints for the drone movement where the drone visits a customer according to the truck position. Constraints (20)–(23) indicate the time coordination of the truck and the drone—the launch

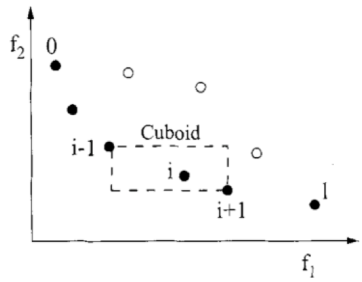

Fig. 1. Crowding distance calculation (Deb et al., 2002).

and return operations time of the drone must be coordinated with the truck. Constraint (24) ensures that if the drone is launched from node i to node j and returns to node k, the process must satisfy the endurance time of the drone's battery. Constraints (25) and (26) define the sequence in which trucks make a visit and determine if they have visited another customer on the route before. Constraint (27) can avoid the drone from being launched again when the drone is already working on the route. Constraint (28) explains that if the drone is launched from the depot to node k, the truck must depart from any point and visit node k to recover the drone. Constraints (30)-(33) define the domain of the decision variables used in this model.

Constraint (29) refers to the service level constraint for this model, which must be greater than S and the target service level for the route. In this model, the service level will be calculated using the soft time windows system. The t will become either the arrival time of the truck (a_i^v) or the arrival time of the drone ($a_j^{\prime v}$) at the customer. The service level will become 1 (or 100 %) if the delivery is within customer time windows. However, the customer's time windows may sometimes be violated for economic and operational issues. If the delivery is done outside of the time windows, then the service level must be calculated since the delivery made between [EET_i, TW_i^S] and [TW_i^E, ELT_i] can still be considered "it's all right" (Nahum et al., 2014). Using this concept of EET_i and ELT_i, the service level for each customer can be defined by a fuzzy membership function as follows:

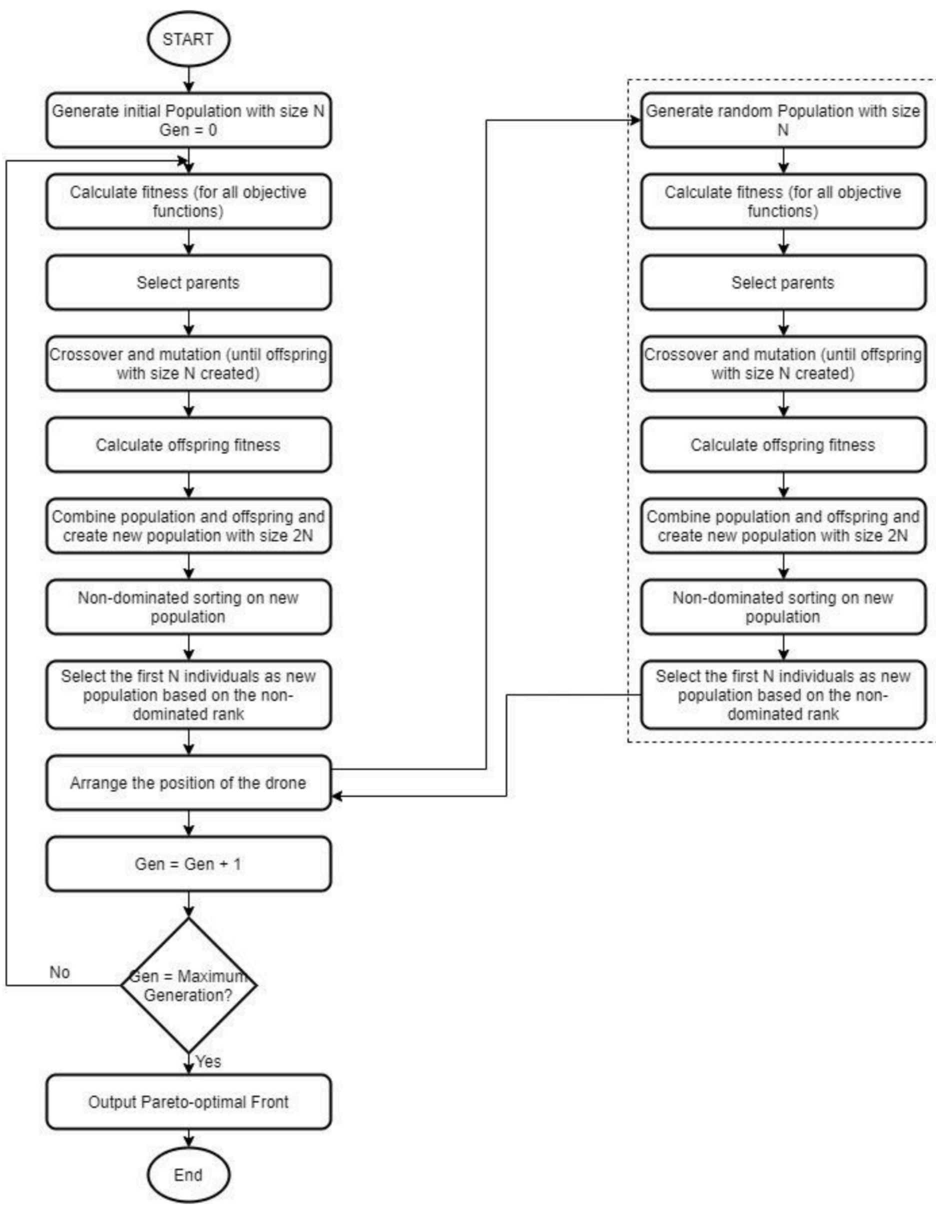

Fig. 2. Flowchart of NSGA-II for VRPD.

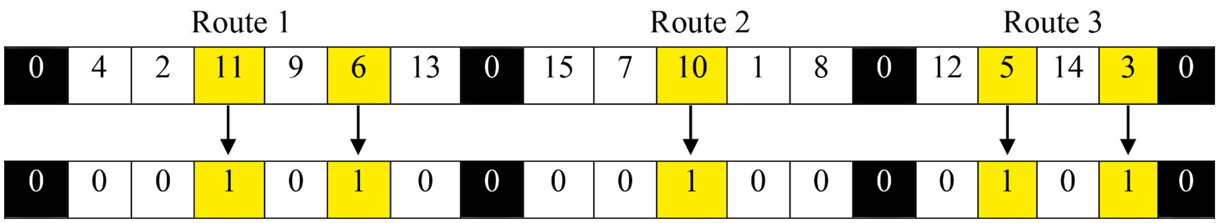

Fig. 3. Chromosome for the drones and the trucks.

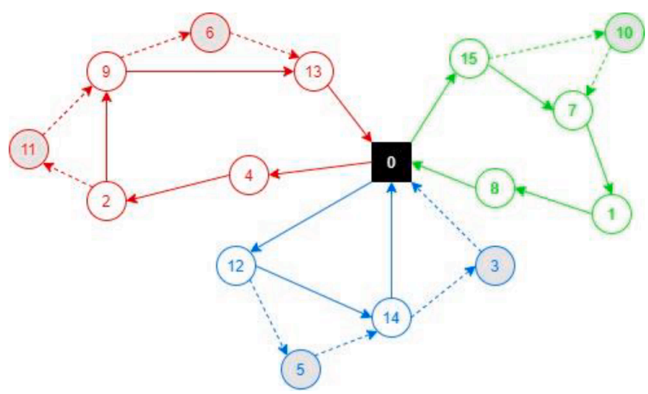

Fig. 4. Route for delivery according to chromosome in Fig. 7.

$$S_i(t) = \begin{cases} 0, t < EET_i \\ f_i(t), EET_i \le t < TW_i^S \\ 1, TW_i^S \le t < TW_i^E \\ g_i(t), TW_i^E \le t < ELT_i \\ 0, ELT_i < t \end{cases} \tag{34}$$

(34) shows that if the customer is served within their time window, the service level will become 1. However, if it is outside the EET_i or ELT_i, the service level will become 0. Notably, there are functions—$f_i(t)$ and $g_i(t)$—that can obtain the service level for delivery made between [EET_i, TW_i^S] and [TW_i^E, ELT_i], respectively. This function is also called the dissatisfaction function because it is assumed that all customers are not pleased when the supplier arrives either early or late. The said function is defined in Eqs. (35) and (36) as follows:

$$f_i(t) = 1 - \left(\frac{t - EET_i}{TW_i^S - EET_i}\right)^5 \tag{35}$$

$$g_i(t) = 1 - \left(\frac{ELT_i - t}{ELT_i - TW_i^E}\right)^5 \tag{36}$$

In this model, the arrival time of the trucks and the drones for the customers when delivering is an important indicator since the carbon emission function in Eq. (2) uses distance to define the amount of carbon emission emitted in the route. It is necessary to transform the travel time of the trucks and the drones to distance. The formula is defined using Eq. (37) as follows:

$$d_{ij} = t_{ij} \times vehiclespeed \tag{37}$$

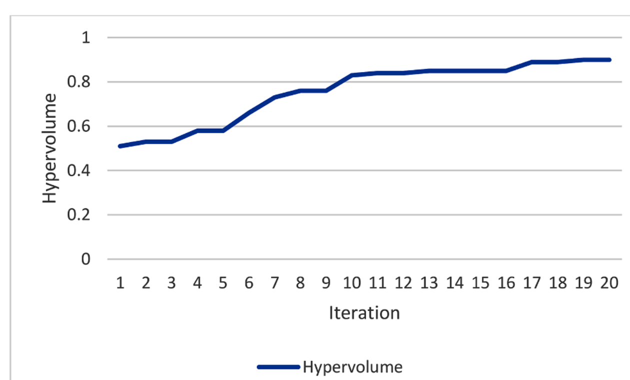

Fig. 5. Algorithm convergence.

Table 1
Parameter setup.

Parameters	Value or Range
Number of customers (N)	{10, 20, 50, 100}
Size of area (A)	{5 × 5, 10 × 10, 20 × 20, 40 × 40}
Drone endurance (E)	{30, 60}

Table 2
Parameter test.

Mutation rate	0.7			0.6		
Crossover rate	0.7	0.8	0.9	0.7	0.8	0.9
Hypervolume	0.84	0.87	0.86	0.82	0.87	0.86
Spacing	0.23	0.16	0.15	0.24	0.22	0.15
Running time (min)	7.71	7.74	7.93	7.93	7.92	7.98

Table 3
Service level boundary.

	0.8	0.9
Truck makespan (min)	301.0887	3854.997
Truck carbon emission (kg)	19.41575	251.1281
Truck and drone makespan (min)	271.8402	3467.338
Truck and drone carbon emission (kg)	14.92839	193.6967

Table 4
One sample solution.

Solution	Truck chromosome	Truck time (*sec*)	Truck emission (kg)	Drone chromosome	Truck and drone time (*sec*)	Truck and drone emission (kg)
1	1	263.03	19.78	1	226.15	11.67
		263.03	19.78	2	238.42	11.31
	2	250.14	20.42	1	227.17	16.63

Table 5
Hypervolume and spacing resuls.

Instance	Average hypervolume	Standard deviation hypervolume	Average spacing	Standard deviation spacing
10-05-30	82.34 %	0.0934	0.224906	0.2302
10-05-60	84.64 %	0.1507	0.131699	0.2019
10-10-30	81.66 %	0.1526	0.175236	0.1868
10-10-60	83.36 %	0.1311	0.205262	0.2086
10-20-30	85.92 %	0.1181	0.253339	0.2701
10-20-60	86.02 %	0.1345	0.192178	0.3145
20-10-30	81.13 %	0.1352	0.120495	0.207
20-10-60	89.74 %	0.101	0.26115	0.1239
20-20-30	92.45 %	0.1204	0.131544	0.1951
20-20-60	85.65 %	0.1436	0.269378	0.2843
20-40-30	88.52 %	0.1283	0.263971	0.2185
20-40-60	87.94 %	0.126	0.170973	0.2807
50-10-30	85.56 %	0.1488	0.196921	0.3203
50-10-60	85.16 %	0.1242	0.144757	0.2193
50-20-30	84.31 %	0.1167	0.188375	0.1883
50-20-60	81.91 %	0.1339	0.234207	0.1966
50-40-30	90.77 %	0.1211	0.256365	0.226
50-40-60	81.53 %	0.1414	0.210216	0.208
100-10-30	81.75 %	0.1113	0.186439	0.1332
100-10-60	85.00 %	0.1183	0.24021	0.2094
100-20-30	85.82 %	0.1353	0.233587	0.2125
100-20-60	91.56 %	0.1369	0.239184	0.1426
100-40-30	83.88 %	0.1059	0.238426	0.2739
100-40-60	87.82 %	0.1036	0.264694	0.1918
150-10-30	84.17 %	0.1097	0.155063	0.1371
Average	85.54 %		0.207543	
Standard Deviation	0.03226		0.04641	

The vehicle speed in Eq. (37) refers to both the truck and the drone speed, which will be defined later in the experiment.

3.4. NSGA-II for multi-objective VRPD

Research in VRP has applied metaheuristics, such as GA and NSGA-II. However, although NSGA-II has been applied in VRP, it has never been used for solving VRPD problems. The NSGA-II algorithm is quite similar to GA. The process starts with generating the initial population and calculating both the objective function values for each chromosome. In this study, two objective functions are considered the makespan and the carbon emission of the route. After the initial population is generated, the parents are chosen from the initial population to conduct a crossover and create offspring.

In this study, the two-point crossover is used for the crossover operator, in which the algorithm chooses two points in both parents' chromosomes whose genetic materials in between are exchanged. Since the chromosomes are in the delivery sequence and some cases later use more than 1 vehicle, the depot position in the crossover process will be reserved, and only the customer position will be swapped.

After the crossover process, the mutation is implemented. In this study, the swap mutation is employed as the mutation operator, in which we randomly choose two points in the chromosome and exchange the position. In addition, the insertion method is employed, in which we randomly choose two points and insert one of the chromosomes into the position of the other. The inversion mutation is also used, in which two points are randomly chosen, and the sequence between these two points is reversed.

Further, this study will randomize the usage among these three methods with the same probability. After the mutation process, the objective function values for the offspring will be calculated. In the NSGA-II, it is necessary to generate the offspring with the same number of the population *N*, since the initial population and the offspring will later be combined to create a new population with the size of 2 *N*. After the new population is generated, the next process in NSGA-II involves conducting non-dominated sorting for the new population.

The non-dominated sorting process starts with checking which chromosome is not dominated by any other chromosome based on the chromosome's objective function values. Chromosome *X* dominates Chromosome *Y* if all the objective function values from chromosome *X* are not worse than that from Chromosome *Y* and at least one of the objective function values from Chromosome *X* is better than that from Chromosome *Y*. Through this rule, the algorithm will collect all chromosomes that are not dominated by any other chromosome and place all of them in a front called Front 1. Subsequently, all the chromosomes in Front 1 will be deleted from the population, and the process will be repeated to create the next front. The front creation process will be done until the algorithm has the same number of *N*, and the other chromosomes will be rejected. The crowding distance of each chromosome will also be calculated by calculating the average side length in cuboid form between two points (chromosomes) on either side of that chromosome. The shape of the cuboid is drawn with a dashed line in Fig. 1, and the points marked in filled circles are solutions of the same non-dominated front.

After the new population is generated with the number of *N*, the next process involves implementing the same thing for the drone assignment. Given that this study uses VRPD, another NSGA-II process for the drone is necessary before continuing to the next iteration. Since the drone assignment is done after the route for the truck is created, the algorithm will attempt to find which customer, if served with the drone, can get a better result. Fig. 2 illustrates the flowchart of the process. The left side is the NSGA-II process for the truck's route, and the right side is the NSGA-II process to determine which customer will be served by the drone.

The chromosome in the NSGA-II for the VRPD is, as mentioned before, the sequence of the customers for the delivery. Integer numbers are used to represent the customers, and 0 is used to represent the depot. The sequence of 0 indicates the starting of the route; if there are several 0 s in the chromosome, it means that after the second, third, and so on, 0 is a different truck with its respective route. In the drone assignment, after the truck route is done, the customer before and after the customer that the drone will serve is considered the launching and returning point for the drone, respectively. An example of a chromosome with 3 trucks and 15 customers is illustrated in Fig. 3.

The upper part shows the 3 truck routes—that is, (0–4-2–11-9–6-13–0), (0–15-7–10-1–8-0), and (0–12-5–14-3–0). On the other hand, the

Table 6
Average and standard deviation of makespan and carbon emission.

Instance	Average makespan truck (min)	Average makespan truck and drone (min)	Standard deviation makespan truck (min)	Standard deviation makespan truck and drone (min)	Average carbon emission truck (kg)	Average carbon emission truck and drone (kg)	Standard deviation carbon emission truck (kg)	Standard deviation carbon emission truck and drone (kg)
10-05-30	301.09	271.84	22.21	19.97	19.42	14.93	1.01	1.71
10-05-60	298.43	268.16	22.78	19.8	19.45	14.79	1.01	1.81
10-10-30	329.48	294.14	22.28	20.82	39.69	30.35	3	3.66
10-10-60	327.21	291.99	20.98	20.33	39.81	30.45	3.07	3.76
10-20-30	427.4	427.09	30.89	25.59	82.98	63.52	4.41	8.99
10-20-60	486.32	440.84	27.05	30	82.36	66.06	4.87	9.35
20-10-30	687.3	617	51.79	52.71	89.26	68.89	4.84	7.75
20-10-60	710.38	641.41	48.23	46.83	89.83	68.93	4.95	6.94
20-20-30	884.97	795.65	56.92	67.46	161.49	122.51	7.91	11.49
20-20-60	896.85	803.17	48.42	53.52	160.79	120.59	7.81	12.36
20-40-30	1419.02	1362.3	126.26	126.43	362.27	275.94	26.2	29.34
20-40-60	1424.32	1278.24	95.72	109.25	354.84	268.16	22.81	30.34
50-10-30	2111.09	1923.98	198.9	201.48	232.1	174.99	12.05	12.88
50-10-60	2123.28	1938.26	178.66	194.47	230.23	177.85	12.93	15.08
50-20-30	2857.11	2546.53	234.05	234.62	470.45	360.57	29.65	35.19
50-20-60	2826.63	2527.02	222.73	216.16	471.4	358.1	26.93	30.05
50-40-30	3926.3	3537.89	320.76	322.28	929.99	710.12	39.78	42.86
50-40-60	4005.16	3553.57	261.41	297.46	940.69	723.63	41.12	54.97
100-10-30	4269.97	3855.23	255.65	303.57	492.88	382.83	19.18	21.15
100-10-60	4221.19	3822.44	166.2	181.46	490.43	380.1	17.6	22.66
100-20-30	5613.37	5076.19	242.41	241.77	982.27	756.89	39.96	44.49
100-20-60	5638.11	5107.68	209.27	232.35	990.26	768.84	45.81	45.14
100-40-30	9230.45	8405.48	429.84	426.45	1958.73	1510.03	67.3	89.37
100-40-60	9447.11	8515.61	488.67	513.15	1940.86	1504.62	82.43	89.56
150-10-30	5810.57	5229.49	379.87	351.95	740.56	580.1	12.3	22.47

lower part indicates the customers that will be served using the drone. In this case, the drone will serve customers 11 and 6 in route 1, similar to customer 10 in route 2, and customers 5 and 3 in route 3. Further, the customers served by the drone previously and afterward are considered the launching and returning points of the drone. For example, for customer 11 in route 1, customer 2 is the launching point and customer 9 is the returning point; for customer 6, the launching point is customer 9 and the returning point is customer 13. The chromosome in Fig. 3 can be presented as a graph in Fig. 4 if we change the chromosome to be a more route-like representation, where the dashed line represents the drone's travel, and grey-colored nodes represent the nodes served by the drone.

3.5. Performance evaluation

A numerical experiment will be conducted to determine the feasibility of the proposed NSGA-II for the VRPD with two objectives in this study. Some data will be employed and used to check whether the solution from the proposed model is good or not. Several types of cases are employed, from small-size to large-size customers. All results will be presented in the later part. Further, quality indicators will be employed—that is, Hypervolume and Spacing—to evaluate the quality of the solutions set in the proposed multi-objective algorithm. The hypervolume indicator is a unary performance measure designed to compare the Pareto fronts given by multi-objective combinatorial optimization algorithms (Drugan & Thierens, 2010). The larger the hypervolume, the larger the volume between a reference point and the Pareto front, and the better the algorithm performs. It is worth noting that the reference point specification has not been studied in detail from the viewpoint of fair performance comparison of multi-objective optimization algorithms. In this study, the reference point will be calculated using Eq. (38) (Ishibuchi et al., 2018):

$$r = 1 + \frac{1}{H} \tag{38}$$

Table 7
The differences in makespan and carbon emission between with and without drone.

Instance	Makespan difference	Carbon emission difference
10-05-30	9.71 %	23.12 %
10-05-60	10.14 %	23.96 %
10-10-30	10.73 %	23.53 %
10-10-60	10.76 %	23.51 %
10-20-30	9.59 %	23.45 %
10-20-60	9.35 %	19.79 %
20-10-30	10.23 %	22.82 %
20-10-60	9.71 %	23.27 %
20-20-30	10.09 %	24.14 %
20-20-60	10.45 %	25 %
20-40-30	3.99 %	23.83 %
20-40-60	10.26 %	24.43 %
50-10-30	8.86 %	24.61 %
50-10-60	8.71 %	22.75 %
50-20-30	10.87 %	24.03 %
50-20-60	10.6 %	27.97 %
50-40-30	9.89 %	23.64 %
50-40-60	11.28 %	23.08 %
100-10-30	9.71 %	22.33 %
100-10-60	9.45 %	22.5 %
100-20-30	9.57 %	22.95 %
100-20-60	9.41 %	22.36 %
100-40-30	8.94 %	22.91 %
100-40-60	9.86 %	22.48 %
150-10-30	10.1 %	21.61 %
Average	9.69 %	23.36 %
Standard deviation	0.0135	0.0143

where r is the reference point, and the H is a user-defined integer parameter; the value of H can be defined using Eq. (39):

$$C_{m-1}^{H+m-1} \leq \mu < C_{m-1}^{H+m} \tag{39}$$

The μ in Eq. (39) is the population size, and the m is the number of objectives. This reference point can be used to calculate the hypervolume where the objective must be normalized to become values between 0 and 1. Hypervolume can be used to check the quality of the solution sets regarding convergence, spread, and cardinality. On the other hand, spacing can be used to check the uniformity of the solution sets. Spacing measures the variation of the distance between solutions in a set (Schott, 1995). It can be calculated using Eq. (40):

$$Spacing(A) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\overline{d} - d_1(a_i, A/a_i)\right)^2} \tag{40}$$

In the solution $A = \{a_1, a_2, \ldots, a_N\}$, the $\overline{d}$ is the mean of all $d_1(a_1, A/a_1)$, $d_1(a_2, A/a_2)$, …, and $d_1(a_N, A/a_N)$ means the L^1 norm distance (Manhattan distance) of a_i to the set A/a_i, namely

$$d_1(a_i, A/a_i) = \min_{a \in A/a_i} \sum_{j=1}^{m} |a_{ij} - a_j| \tag{41}$$

where m is the number of objectives and a_{ij} is the j th objective of solution a_i. Spacing must be minimized, as a lower spacing value is better. A lower spacing value means that the solution is more distributed equally. Therefore, this study will use hypervolume and spacing as quality indicators to evaluate whether the solution sets produced by the algorithm are good enough or not.

4. Experimental results

This section attempts to evaluate the performance of the NSGA-II using randomly generated instances. This study implemented the NSGA-II algorithm in Python version 3.8 and ran on a desktop PC with Intel i7-7700 3.6 GHz processor and 32 GB RAM.

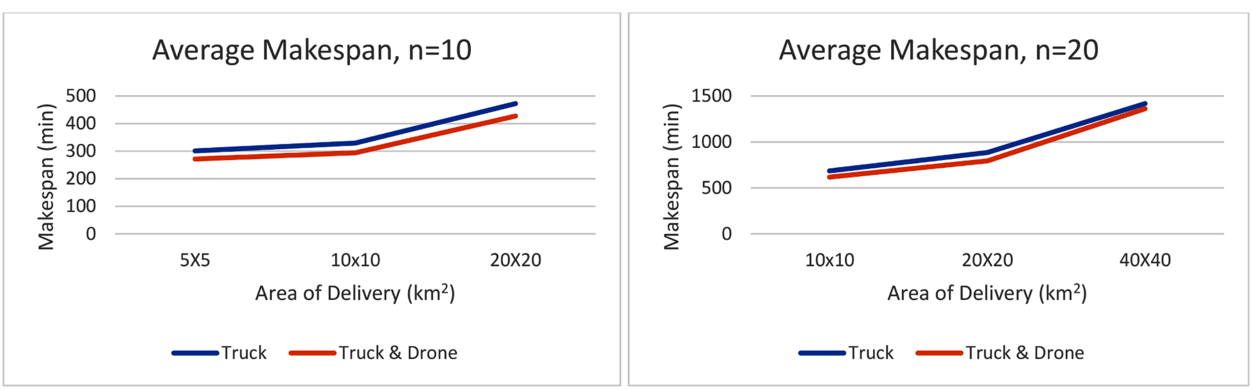

Fig. 6. Average makespan for 10 and 20 customers.

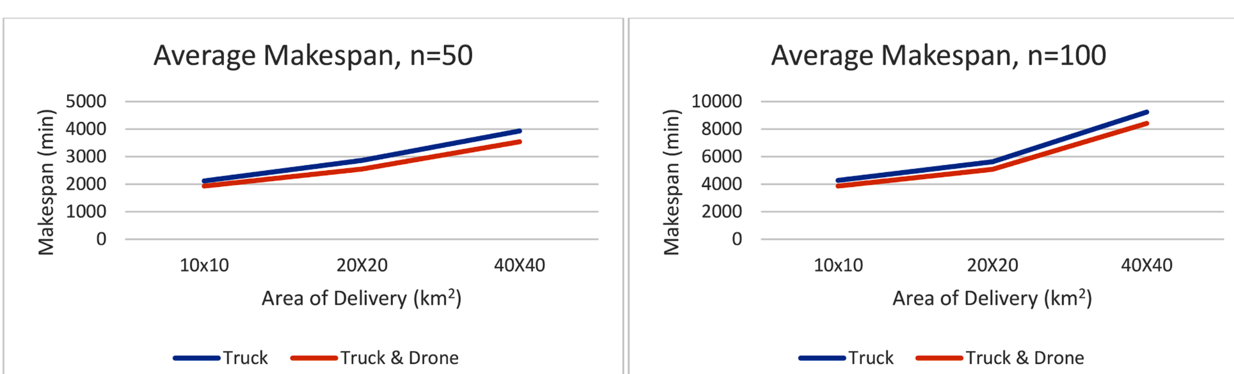

Fig. 7. Average makespan for 50 and 100 customers.

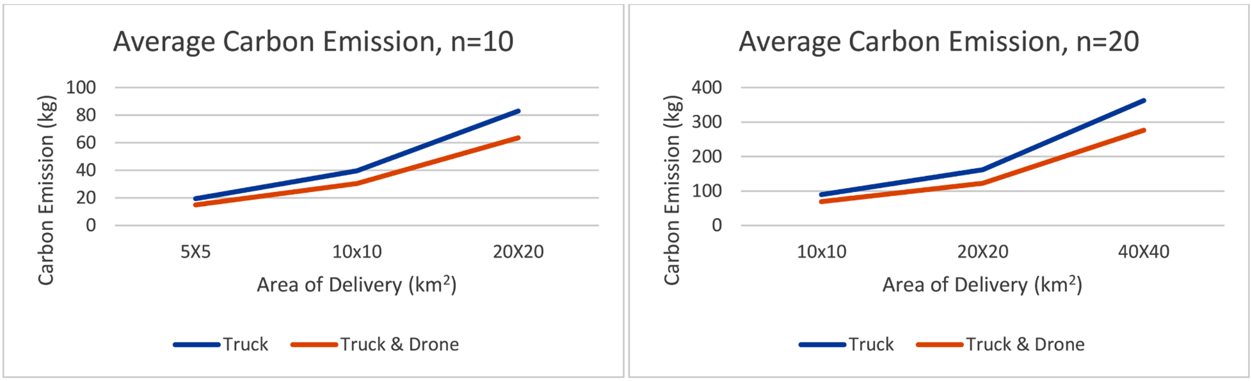

Fig. 8. Average carbon emission for 10 and 20 customers.

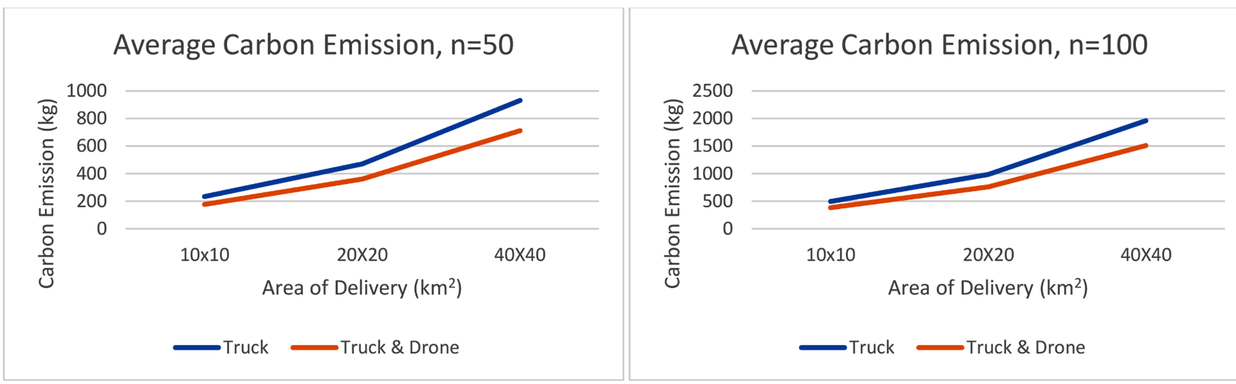

Fig. 9. Average carbon emission for 50 and 100 customers.

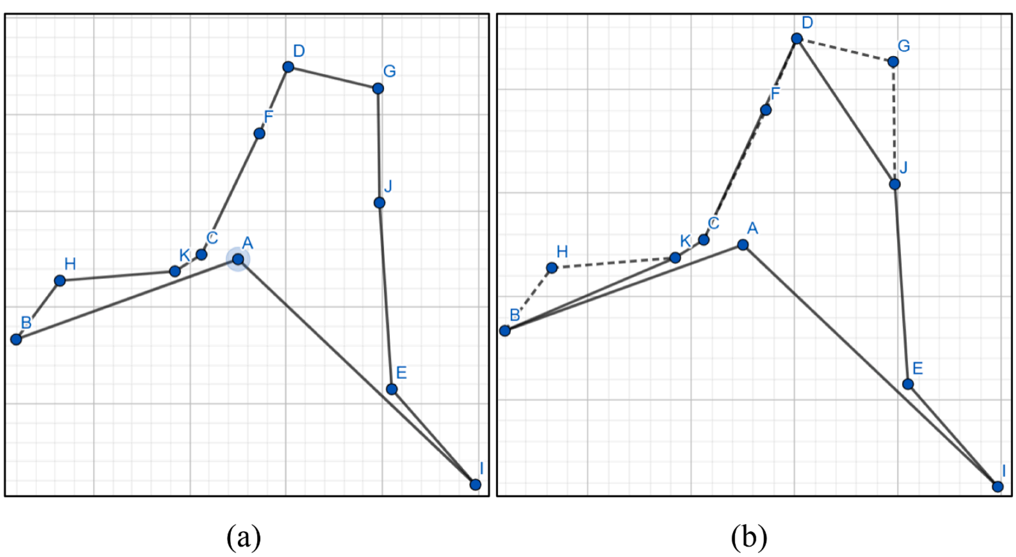

Fig. 10. Routing result.

4.1. *Test instances*

For experimental purposes, customers' locations are randomly generated to test the proposed algorithm. The instances are constructed with 10, 20, 50, and 100 customers, randomly generated in four-area dimensions including 5 × 5, 10 × 10, 20 × 20, and 40 × 40 in miles area. The speed of the trucks is set at 35 mph, and the speed of the drones is set at 50 mph (Sacramento et al., 2019). The endurance of the drones is set to be 30 and 60 min, respectively. In addition, an instance from Sacramento et al. (2019) is adopted to examine the feasibility of the algorithm with 150 customers in a 10-mile area and a 30-minute drone endurance. The setup time of launching and receiving the drone is set to be 1 min. Table 1 summarizes the parameters for the generated instances.

The distance matrix of trucks is calculated using the Manhattan distance. This study simulates the delivery process using trucks and drones in the city. Since the delivery is assumed to be done in the city, buildings in the city will be considered obstacles to the drones' flight;

hence, the distance matrix is also calculated using the Manhattan distance. The location of the depot is assumed to be at the center of the area.

4.2. Experiment and results

The instances are constructed using 10, 20, 50, and 100 customers, where 10-customer instances are generated in the areas of 5 × 5, 10 × 10, and 20 × 20, while the 20-, 50-, and 100- customer instances are generated in the areas of 10 × 10, 20 × 20, and 40 × 40. The numbers of vehicles for the 10-, 20-, 50-, and 100-customer instances are 1, 2, 6, and 12, respectively, and the number of drones used is 1 for each truck. All instances will be run 30 times using the NSGA-II algorithm. Then, the non-dominated sorting process is conducted on the solution sets to choose only the first front of the solution sets to reduce the size of the solutions.

Parameters in the NSGA-II algorithm—the crossover and mutation rate—must be tested to determine whether or not the combination of the parameter values can produce good results. Some values for the crossover and mutation rate are used to find solutions for an instance with 10 customers. Then, the hypervolume and spacing values are calculated to compare the solutions' quality and determine which combination of the parameter values can produce the best solution.

Table 2 lists all hypervolume values, spacing values, and running time for different combinations of the parameter values tested. The result shows that the highest hypervolume value can be produced using a crossover rate of 0.8, either in a mutation rate of 0.7 or 0.6. However, the spacing is lower than that in the mutation rate of 0.7, and the running time is lower than that in the mutation rate of 0.6. Based on this result, the parameter values used for mutation rate is 0.7 and crossover rate is 0.8. It is marked with yellow color on the table.

As previously mentioned, the soft time windows in this model will be incorporated to measure the service level of the delivery route. The service level boundary is decided before running the program, as in Eq. (29). It must first be tested to determine the maximum service level the model can attain.

Table 3 lists the average result of the makespan and the carbon emission of the delivery route using only a truck and using both a truck and a drone. This test employs a 10-customer instance, in which the service level tested is from 0.8 and will be increased to find the maximum service level. Further, Table 3 shows that the model can still achieve a good result even when using service level 0.8. However, after increasing the service level to 0.9, some of the solutions' service levels cannot reach 0.9. If the solution service level cannot reach the determined service level, the solution will be multiplied by a penalty. Therefore, the service level will be set to 0.8 for all instances.

An example of the solutions from the 10-customer instance obtained for deliveries with and without drones is presented in Table 4. For the delivery without a drone, the result indicates that all the final solutions are exactly the same, which means that the employed solution is the most optimal. On the other hand, the solutions for the delivery with a drone are not always the same, although the difference is not significant. On average, the delivery with a drone also costs smaller transportation time and carbon emissions.

As mentioned in Section 3, there are two different chromosomes—the truck and the drone chromosomes. The truck chromosome represents the truck route, which includes a first front from the solution sets. The truck time and the truck carbon emission are the total makespans of that truck route and the total emission in kilograms, respectively. The drone chromosome represents the route that uses trucks and drones for delivery. Most of them only have one solution, as shown in Table 4. However, there are two that have two drone chromosomes, which implies that the truck route from chromosome 5 has two types of routes in a combined truck and drone delivery. The truck and drone time is the total makespan of that route, and the truck and drone carbon emissions are the total carbon emissions.

Hypervolume can be used to examine the convergence of the algorithm. The hypervolume value of the solutions for each iteration must be calculated; the value from each iteration will show the convergence. The convergence is calculated from a 10-customer instance with the same parameters as all the solutions. Fig. 5 shows the convergence from the 10-customer data. The result shows that the convergence of the algorithm converges from a hypervolume value of 0.51 in iteration 1 to 0.9 in iteration 20. The X-axis in this figure is the iteration, and the Y-axis is the hypervolume values.

Because of the reference point specification using the formula of Ishibuchi et al. (2018), all the solution sets are normalized to have a value between 0 and 1. Then, the hypervolume and the spacing of each sample from all test instances can be calculated. The hypervolume value ranges from 0 and 1. Similarly, since the spacing is also calculated using the normalized value, the result of the spacing also ranges from 0 and 1. The average results and the standard deviation of hypervolume and spacing from all samples for all test instances are listed in Table 5.

The hypervolume value, in percentage, refers to how big the area is covered by the Pareto front solutions and the reference point. The average hypervolume value shown in Table 5 is above 80 %. There are also three instances with average hypervolume values over 90 %—that is, the 20–20-30, 50–40-30, and 100–20-60 instances. These hypervolume values show that the quality of the proposed algorithm is quite good, it can produce solutions with high hypervolume values around 80 % and above, and the average from all instances is 85.54 %.

The average spacing value of all instances is also between 0 and 1 since it uses a normalized value to make the calculation. The value is around 0.13 to 0.26; however, there is one with a value of 0.27 from the 20–20-60 instance. There is also an instance with a spacing value of 0.12 from the 20–10-30 instance. The algorithm can produce solutions that are quite equally distributed since the spacing values are around 0.13 to 0.26, and the average from all instances is 0.208.

Based on these two quality indicators, the hypervolume and spacing values, it can be concluded that the algorithm can produce good solutions with hypervolume values above 80 % and spacing values between 0.13 and 0.26. The difference in the makespan of the delivery routes and the total carbon emissions emitted will be compared between using only trucks for delivery and using both trucks and drones for delivery. Table 6 shows the average and the carbon emission of the makespan and the carbon emission of all instances using trucks and using both trucks and drones for delivery. Table 7 shows the difference between the makespans and the total carbon emissions emitted in all instances.

The differences in the makespan and the carbon emission are calculated using Eq. (42) as follows:

$$\%diff = \left(1 - \frac{z_a}{z_b}\right) \times 100\% \tag{42}$$

where z_a is the solution for using trucks and drones for delivery, and z_b for using only trucks for delivery; both for the makespan and the carbon emission are the same. The result shown in Table 7 is obtained by calculating all the differences and then averaging all results. From this result, we can see that the usage of drones in delivery can decrease the total makespan by around 9 % and the total carbon emission by around 23 % for all instances.

Fig. 6 and Fig. 7 show the difference in the average makespan between using trucks only and using both trucks and drones in delivery. In these figures, the drone endurance is set to 30 min; in all instances and areas, the difference between using only trucks and using trucks and drones can be seen clearly.

The reduction in the makespan is not very high for all instances; it is only around 9 %, with an average of 9.69 %. Although the reduction in the makespan is not only around 9 %, in the case of the last-mile delivery, the reduction of the delivery time of 9 % can produce quite a big impact. From the experiment, in the case of the 100-customer delivery in a 40-mile area, using only trucks can lead to a 9000-minute makespan.

However, using drones can decrease 810 min and speed up the delivery. From an economic point of view, the reduction of 810 min can lead to significant cost savings. The 100-customer instance uses 12 trucks for delivery; if the total makespan can be reduced by 810 min, the truck needed can then be reduced. With an average working hour per day of 8 h (480 min) and the makespan reduction of 810 min, we can reduce the truck and the driver needed by one.

Fig. 8 and Fig. 9 show the difference in the average carbon emission between using trucks only and using both trucks and drones in delivery. The X-axis in these figures refers to the area of delivery, and the Y-axis is the range of the values for carbon emission. In these figures, the endurance is also set to 30 min for all instances. From these graphs, the difference in using drones in delivery can be seen clearly. Noteworthily, as the area of delivery increases, the average makespan and the average carbon emission also increase.

The average reduction in carbon emissions is 23.36 %. It is safe to say that using drones for delivery can reduce carbon emissions by over 20 %. From the environmental point of view, this reduction in carbon emission is quite high and shows that the usage of drones in delivery is good. From the experiment involving 100 customers in a 40-mile area, using only trucks can produce around 2000 kg of carbon emission. However, combining trucks and drones for delivery can reduce the carbon emission to around 1600 kg. This reduction is highly beneficial for the environment, especially now when online shopping is very popular and last-mile delivery is increasing. The usage of drones in delivery, instead of only using trucks, can reduce not only the time required to complete the delivery but also benefit the environment as it can reduce the carbon emission produced from that delivery route.

In addition, hypothesis testing was also conducted to test whether or not the statistical difference exists between using only trucks and using both trucks and drones for all instances. The *P*-values of all tests are all 0 s, which are smaller than the α value of 0.05. This finding means that the null hypothesis $\mu_{VRPTW} \leq \mu_{VRPTWD}$ is rejected and that the alternative hypothesis $\mu_{VRPTW} > \mu_{VRPTWD}$ is true, and the difference when using trucks and drones for delivery is significant. Since the difference is significant, it means that the usage of drones in delivery can be applied by companies focusing on last-mile delivery.

One example of the route created by the algorithm is presented in Fig. 10. This route is one of the solutions from the 10-customer instance in a 5-mile area with a 30-minute drone endurance battery. Fig. 10 shows the route result from one of the solutions; (a) is the route using a truck only for delivery, and (b) is the route using both a truck and a drone for delivery. The solid line in Fig. 10(b) refers to the truck movement, and the dashed line refers to the drone movement.

4.3. Time complexity

The vehicle routing problem is an *NP*-hard problem, which means that the required solution time increases exorbitantly with size. The proposed algorithm used in this study is NSGA-II. To observe the scalability of the proposed algorithm, this study adopts a simple analysis method using the big *O* method in the worst-case running time and provides the upper bound complexity of the algorithm. The overall complexity of the proposed algorithm is $O(2MN^2)$, where M is the number of objectives and N is the population size. In this study, the algorithm runs two NSGA-II algorithms; the first is used for the truck route, and the second is used for the drone route in one generation. Hence, the usual $O(MN^2)$ complexity from the NSGA-II algorithm is multiplied by 2.

5. Conclusions

Many companies seek the usage of drones in delivery. Meanwhile, the growth of online shopping makes the need for higher delivery speed—a significant tool for competitiveness. This study finds the benefit of collaborating trucks and drones for delivery to reduce the makespan and carbon emissions. This study minimizes the makespan and the total carbon emission of the delivery route. The result shows that the usage of drones for delivery can reduce the makespan by around 9 % and carbon emission by around 23 %. Hypervolume and spacing are used as quality indicators to determine the quality of the solutions produced by the proposed algorithm. It can be concluded that the algorithm is quite promising as it can produce solutions with hypervolume values above 80 % and spacing values between 0.13 and 0.26. The difference in using drones for delivery is quite significant. All significance test results indicate that the difference is significant and that the usage of drones can provide a better benefit in delivery.

Furthermore, this study extends and modifies the mathematical model, focusing on drone delivery, makespan, and carbon emission formulation. It also proposes the mathematical formulation of VRPD, which is an extension of the mixed integer programming (MIP) of the FSTSP presented by Murray and Chu (2015), the VRPD presented by Sacramento et al. (2019), and the VRPTWD by Kuo et al. (2022). In addition, this study uses the NSGA-II algorithm to solve the VRPD with two objectives of minimizing the makespan and carbon emission. By comparing the result of using trucks only and using both trucks and drones, the experimental results showed that the algorithm is effective for both small- and large-scale problems. The proposed algorithm can reduce both the makespan and carbon emissions of the delivery route.

Moreover, this study considers two objectives of minimizing the makespan and carbon emission of the VRPD. Although the result is quite good, there are still some directions for future research, including the application of different metaheuristics like the multi-objective particle swarm optimization (MOPSO) and strengthening the Pareto evolutionary algorithm (SPEA) or other multi-objective evolutionary algorithms. Further, using different types of vehicles and drones with different carbon emission levels may lead to different results. Therefore, allowing the drones to do more than one delivery—as long as the drone endurance complies—and investigating the movement of drones between trucks may be another possible research direction.

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.

Data availability

The data can be found in the Internet.

References

Andrea, R. D. (2014). Can drones deliver. *IEEE Transactions on Automation Science and Engineering, 72*(4), 647–648.

Bento, M. N. (2008). Unmanned aerial vehicles: An overview. *Inside GNSS*, 54–61.

Boysen, N., Briskorn, D., Fedtke, S., & Schwerdfeger, S. (2018). Drone delivery from trucks: Drone scheduling for given truck routes. *Networks: An International Journal, 72*(4), 506–527.

Bulhões, T., Minh, H., Martinelli, R., & Vidal, T. (2018). The vehicle routing problem with service level constraints. *European Journal of Operational Research, 265*(2), 544–558.

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.

Cordeau, J., Desaulniers, G., Desroriers, J., Solomon, M. M., & Soumis, F. (1999). The vehicle routing problem with time windows. *Les Cahiers du GERAD G-99-13*, 1–38.

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*, Article 105004.

Deb, K., Pratap, A., Argawal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Drugan, M. M., & Thierens, D. (2010). Path-guided mutation for stochastic pareto local search algorithms. *Parallel Problem Solving from Nature - PPSN XI, Part, I*, 485–495.

Elabib, I., Basir, O. A., & Calamai, P. (2002). An experimental study of a simple ant colony system for the vehicle routing problem with time windows. *International Workshop on Ant Algorithms ANTS, Lecture Notes in Computer Science, 2463*, 53–64.

GmbH, D. I. (2014). DHL Parcelcopter Launches Initial Operations for Research Purposes. Retrieved from https://www.dhl.com/en/press/releases/releases_2014/group/dhl_parcelcopter_launches_initial_operations_for_research_purposes.html.

Han, Y. Q., Li, J. Q., Liu, Z., Liu, C., & Tian, J. (2020). Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones. *International Journal of Advanced Robotic Systems, 17*(2), 1–14.

Huang, S. H., Huang, Y. H., Blazquez, C. A., & Chen, C. Y. (2022). Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm. *Advanced Engineering Informatics, 51*, Article 101536.

Ishibuchi, H., Imada, R., Setoguchi, Y., & Nojima, Y. (2018). How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary Computation, 26*(3), 411–440.

Jozefowiez, N., Semet, F., & Talbi, E.-G. (2008). Multi-objective vehicle routing problems. *European Journal of Operational Research, 189*(2), 293–309.

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

Kitjacharoenchai, P., Min, B. C., & Lee, S. (2020). Two echelon vehicle routing problem with drones in last mile delivery. *International Journal of Production Economics, 225*, 1–14.

Kuo, R. J., Lu, S. H., Lai, P. Y., & Windra, S. Y. (2022). Vehicle routing problem with drones considering time windows. *Expert Systems with Applications, 191*, Article 116264.

Kyriakakis, N. A., Stamadianos, T., Marinaki, M., & Marinakis, Y. (2022). The electric vehicle routing problem with drones: An energy minimization approach for aerial deliveries. *Cleaner Logistics and Supply Chain, 4*, Article 100041.

Macrina, G., Pugliese, L. D., Guerriero, F., & Laporte, G. (2020). Drone-aided routing: A literature review. *Transportation Research Part C, 120*, 1–25.

Molina, J. C., Eguia, I., Racero, J., & Guerrero, F. (2014). Multi-objective vehicle routing problem with cost and emission functions. *Social and Behavioral Science, 160*, 254–263.

Moshref-Javadi, M., & Winkenbach, M. (2021). Applications and Research avenues for drone-based models in logistics: A classification and review. *Expert Systems with Applications, 177*, Article 114854.

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

Nahum, O. E., Hadas, Y., Spiegel, U., & Cohen, R. (2014). The Real-Time Multi-Objective Vehicle Routing Problem – Case Study: Information Availability and the Quality of the Results. *93rd TRB Annual Meeting*, 1-21.

Poikonen, S., & Golden, B. (2020). Multi-visit drone routing problem. *Computers and Operations Research, 113*, 1–10.

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.

Pollet, B. G., Staffel, I., & Shang, J. L. (2012). Current status of hybrid, battery and fuel cell electric vehicles: From electrochemistry to market prospects. *Electrochimica Acta, 84*, 235–249.

Sacramento, D., Pisinger, D., & Ropke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C, 102*, 289–315.

Schaffer, J. (1985). Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of the First International Conference on Genetic Algorithm and Their Applications*, 93-100.

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

Schott, J. R. (1995). *Fault Tolerant Design using Single and Multicriteria Genetic Algorithm Optimization*. Massachusetts Institute of Technology.

Srinivas, N., & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation, 2*(3), 221–248.

J. Stewart Stewart, J. (2014). *Google tests drone deliveries in Project Wing trials*. Retrieved from BBC News: https://www.bbc.com/news/technology-28964260.

Tang, J., Pan, Z., Fung, R. Y., & Lau, H. (2009). Vehicle routing problem with fuzzy time windows. *Fuzzy Sets and Systems, 160*(5), 683–695.

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

Wohlsen, M. (2014). *WIRED*. Retrieved from https://www.wired.com/2014/06/the-next-big-thing-you-missed-delivery-drones-launched-from-trucks-are-the-future-of-shipping/.

Zhang, H., Zhang, Q., Ma, L., Zhang, Z., & Liu, Y. (2019). A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows. *Information Sciences, 490*, 166–190.

Zhang, J., Lam, W. H., & Chen, B. Y. (2013). A stochastic vehicle routing problem with travel time uncertainty: Trade-off between cost and customer service. *Network and Spatial Economics, 13*, 471–496002E.