Physical Communication 44 (2021) 101236

Contents lists available at ScienceDirect

Physical Communication

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

Full length article

Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones

Jalel Euchi [a,b,*], Abdeljawed Sadok [a]

[a] Department of Management Information Systems and Production Management, College of Business and Economics, Qassim University, P.O. Box: 6640, Buraidah 51452, Saudi Arabia
[b] Sfax University, ISGIS, OLID Laboratory, LR19ES21, 3021, Sfax, Tunisia

ARTICLE INFO

Article history:
Received 15 April 2020
Received in revised form 19 October 2020
Accepted 3 November 2020
Available online 7 November 2020

Keywords:
Drone delivery
Vehicle routing problem
Mathematical model
Genetic-sweep algorithm

ABSTRACT

Energy consumption has become a crucial problem in the design of vehicle routing problems, hence the need to use another delivery method powered by batteries. Unmanned aerial vehicles have become fundamental tools in tasks for which man has limited skills that prevent a superlative optimization of time. The increasing use of drones by commercial companies such as Amazon, Google, and DHL has given birth to a new variant of vehicle routing problem (VRP) called VRP with drones (VRPD) which has a positive influence on the environment. Where vehicles and drones are used to deliver packages or goods to customers. In VRPD, vehicles and drones make dependent or independent deliveries. In the case of a dependent delivery, at a given point (customer or depot) the drone takes off from a vehicle to serve a customer and then return to travel with the same vehicle, as long as the capacity and endurance constraints for a drone are satisfied. In the other case, each type of vehicle travels independently to others. A MILP model is presented to describe the problem, and then we confirm the formulation via a CPLEX software with small instances. We propose a hybrid genetic algorithm to solve the VRPD. Experiments are carried out on the instances taken from the literature in different settings. The results show the performance of the proposed algorithm to solve this variant.

1. Introduction

A new management system (stock, delivery, orders) has been created by companies given technological progress in robotics. Thanks to new technologies, new delivery strategies are being developed. Drones also called unmanned aerial vehicles (UAVs) have become a perfect work tool. Nowadays, drones take a place in several sectors, as in the audiovisual sector, energy, and agriculture, security, and surveillance.

Following the success of their use, drones are occupying more and more space in companies, especially for services, in particular, package delivery. For 10 years, Atechsys has developed drone systems for delivery and more specifically for "last-mile" logistics. In logistics, in particular, in the transport sector, the term "last-mile" designates the final stage of the delivery process. This is a crucial step in logistics; it has a significant impact on costs and customer satisfaction (e.g., [1]).

Several companies around the world now take the use of drones for delivery seriously. Retail giants like Amazon, Google, and DHL have invested in research projects on the use of drones to optimize their distribution chain. The advantages of drone delivery, as well as saving time, motivated researchers to invest in industrial projects involving the use of drones in the transport field (e.g. [2–4]). Two major problems have attracted the attention of researchers, namely, the traveling salesman problem with the drone (TSPD) and the vehicle routing problem with drones (VRPD).

A traveling salesman wishes to visit a defined number of customers or cities, starting from a point called depot and ending his tour at this same point by visiting all customers or cities once and only once. The traveler's objective is to minimize the total distance traveled (e.g. [5]). This problem is known as the traveling salesman problem (TSP) and is NP-complete. The integration of drones in delivery routes has given rise to a new problem called the traveling salesman problem with drones (TSP-D). The question that arises for this problem, how to optimize delivery times in a supply chain made up of vehicles and drones?

VRPD is a variant of the classic VRP in combinatorial optimization in which several vehicles serve a set of customers from a depot while satisfying customer demand without violating the vehicle capacity constraint the objective is to minimize the total distance traveled (e.g. [6–8]). VRPD consists of optimizing delivery times in a logistics chain made up of vehicles and drones that work without synchronization (vehicles and drones deliver

* Corresponding author at: Sfax University, ISGIS, OLID Laboratory, LR19ES21, 3021, Sfax, Tunisia.
E-mail address: Jalel.Euchi@fsegs.rnu.tn (J. Euchi).

https://doi.org/10.1016/j.phycom.2020.101236
1874-4907/

independently) or with synchronization. (Vehicles and drones make dependent deliveries).

For this work, we are interested in solving the VRPD, and for this purpose; we present a mathematical model and a hybrid genetic algorithm. Genetic algorithms are widely used in optimization tasks and in particular, in the programming of vehicle rounds, as it is a complex problem, which cannot always be, solved with conventional optimization techniques (e.g. [9,10]).

Genetic algorithms are currently one of the most promising lines in the field of artificial intelligence, which is part of the so-called evolutionary techniques, and were originally proposed by John Henry Holland in the 1970s. This technique has as its basic structure from an initial population, carrying out a selection, crossing, or recombination process, a mutation step, and a replacement step in the initial population obtaining as a global result the evolution of the population.

For a computational approach, genetic algorithms simulate a natural selection process to solve problems in the field of mathematical optimization. For this reason, we have chosen to solve the VRPD using a specific and modified hybrid genetic algorithm. Our hybridization offers structuring of the population into several segments and the definition of various fitness functions. These changes in the way in which the search space is run with a disturbance with a local search use performance indicator to maintain a balance between convergence "and the diversity of the solutions obtained.

The main objective of this paper is to propose solutions applicable to a distribution problem with drones. This tool will be very useful for an industrial partner. Among other things, while optimizing the problem, the intelligent use of drones has produced savings in time and distance. The major contributions of this paper are twofold: (i) we formulate the VRPD as a MILP where our study to be among the reduced list of works in this field, also the presentation of the problem concerning literature with a new classification represents a novelty on this subject. (ii) We have developed a new hybrid approach to improve the combination of sweep and genetic algorithms. The results confirm the choice of using the sweep algorithm method to enhance the solution given by the genetic algorithm.

The rest of the paper is organized as follows. The second section presents a review of the literature on work that addresses the problems of VRP with drones. A detailed description of the VRPD with a formulation of the problem as a mixed-integer program is described in Section 3. A solution methodology for the VRPD is outlined in Section 4. An experimental study with an analysis of the results is carried out in Section 5. The paper ends with a conclusion as well as on research prospects in the same context in Section 6.

2. Related works

The considerable development of robotization has given companies more choice. New technology drones are integrated into transport issues by companies, given its speed, these low costs where one or more drones can assist the delivery vehicle. The limits of this technology are small capacity and the total journey time is limited, these limits have given rise to the concept of last-mile delivery.

From 2000, drones began to be used more frequently in non-military activities. They have become more accessible, helping many types of businesses to use them in various applications, such as line inspection, aerial surveying, cartography, geography, film engineering services, and delivery The number of research projects has treated the traveling salesman problem with drones (TSP-D) and vehicle routing problem with drones (VRPD) in a different way has increased for the last years. Many methods – exact and approximate – have been used to determine the best solutions for this problem (e.g. [1,11]).

In the literature, we see the existence of different variants of the problem, different solving approaches, and several related fields that share the same problem. The literature review is divided into three parts: the first concerns the different models discussed in the literature, namely the different sub-problems linked to the management of VRPs with drones, starting with the case of a TSP with a single drone or with m drones. The second part looks more in the case of a VRP with n vehicles and m drones and VRP with the same number of vehicles and drones. Then, in the last part, we are interested in the different solution approaches, that are used to solve these types of problems.

2.1. Traveling salesman problem with drones

Murray and Chu [12] among the first who used drones for a delivery problem, they treated two types of delivery. The first problem concerns delivery by a vehicle and a drone. The second problem concerns the delivery of several drones and a vehicle. Two mathematical models of mixed linear programming have been presented for these two problems. The goal was to minimize the total time traveled. They solved the first problem in two stages, first, a TSP heuristic is used to build the routes for the vehicle, then these routes are improved by integrating the drone in the solution found to reduce the total delivery time, this step is repeated several times until no improvement can be made.

Bouman et al. [13] Based on dynamic programming to find the solution of TSPD. This solution is found from three stages, first, the shortest paths for the vehicle from each start node and end node is determined by dynamic programming then the vehicle paths are combined with drone nodes to find minimum costs by delivering a set of nodes, finally, the optimal sequence which covers all the nodes and starts and ends at the depot is obtained considering the two previous steps. Luo et al. [14] have developed a new integer-programming model for a vehicle and drone routing problem. They assume that the drone can serve any customer, and it is attached to the vehicle so that the drone leaves the vehicle to serve customers and returns to the vehicle to charge its battery for other tours. This problem was solved by two heuristics. The first heuristic H1 determines the route of the TSP, which minimizes the total journey time, this tour is divided into sub-tours after the addition of parking spots for the vehicle. Depending on these parking points, the second heuristic H2 has the role of allocating the tours to the drone. Chang and Lee [15] developed a non-linear programming model for TSP-D. The problem was solved using the technique of k-means. Groups of customers are determined as well as their centers. These form the points of the vehicle tour. At each point, the vehicle sends the drone to serve the group customers. The authors have proven the effectiveness of their method compared to two others. Murray and Raj [16] make an extension of the FSTSP problem by considering several drones on the vehicle. A MILP formulation is proposed to manage several drones. A heuristic solution approach that exploits a subset of sub-problems is given. Tests have shown that the proposed approach effectively solves large problems. Fig. 1 describes TSP with a drone within 14 customers.

2.2. Vehicle routing problem with drones

Wang et al. [17] are the first to study the problem of vehicle routing with drones (VRPD). The goal of this problem is to minimize the time required to serve all customers. The problem is summed up as follows; a fleet of vehicles carries drones to serve customers. The drones leave for delivery, and then the vehicles can pick them up at the depot or one of the customer's locations.

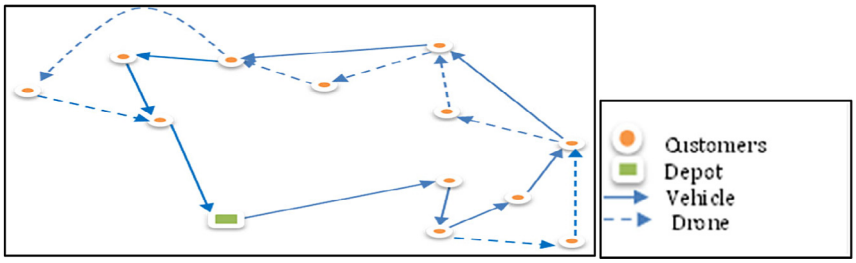

Fig. 1. TSP with drone.

The authors have developed mathematical relationships for different scenarios of combining drones with vehicles. Finally, they showed the impact of the number of drones per vehicle and the speed of drones on the results.

Schermer et al. [18,19] have compared two heuristics they have proposed. The first heuristic is based on the idea of a route-first cluster-second called two-phase heuristic (TPH). For this heuristic, drones, and vehicles delivering independently. Vehicle routes are determined first and then drones are effectively inserted into these routes. The second heuristic is called single-phase heuristics (SPH). For this heuristic, drones and vehicles deliver synchronously. In this case, the routes of vehicles and drones are built together. To determine the performance, numerical experiments on TSP instances were carried out. They found from the comparison of two proposed heuristics that the best solutions are found by the TPH heuristic. Ulmer and Thomas [20] study a new type of delivery called same-day delivery (SDD). They proposed a problem of vehicle routing with drones where delivery is done the same day using heterogeneous fleets of drones and vehicles. To solve this problem, the authors used a political approximation function. They noticed that the delivery costs decrease by the combination of drones and vehicles. Wang and Sheu [21] have proposed a mixed integer-programming model for the vehicle routing problem with drones where the drone can leave a vehicle to serve customers and return to another vehicle to load and do another round. They developed a branch and price algorithms to determine the solution to this problem.

Sacramento et al. [22] considering vehicle capacity and the limitation of total journey time studied a problem similar to that of Murray and Chu [12]. They have formulated a mathematical model for this problem where the objective is to minimize the total traveled time of delivery. Given the complexity of the problem and the difficulty of optimally solving large instances, the authors proposed an Adaptive Large Neighborhood Search metaheuristic to resolve these instances. Finally, they proved the need to integrate drones in vehicle delivery routes. Scherme et al. [23] formulated the VRPD as a mixed-integer linear program (MILP). To improve the performance of MILP solvers in solving VRPD, they determined several valid inequalities like Completion Time Bounds, Symmetry in the VRPD, and Limited Flight Distance without Recharge. To deal with large instances, they proposed a metaheuristic approach that effectively exploits the structure of the VRPD problem considering an existing route of vehicles and they seek an optimal allocation of drones. Finally, the advantages of the valid inequalities offered and the performance of the metaheuristic are studied. Fig. 2 describes VRP with a drone for 19 customers.

2.3. Solution techniques

The basic problems, which concern our topic, are mainly: the problem of the Traveling Salesman Problem (TSP) and the VRP. In all works, the basic principle is the modeling of the problem as a graph followed by the application of an algorithm or heuristic for calculating the shortest path. On this graph, the problem is modeled as a VRP type problem in the case of a fleet of drones or the TSP type in the case of a single drone.

Although for most combinatorial optimization problems, the resolution process is very difficult, many effective methods have been developed which are capable of solving VRP with drones in a reasonable time for small and medium instances. There are several very different approaches to solving these types of optimization problems, and currently, they are frequently combined in hybrid solutions that seek to exploit the good of each by minimizing the flaws. A breakdown according to different criteria and classes is presented in the following table with research that has been recently published (see Table 1).

3. Problem statement and MILP formulation

A fleet of identical vehicles V characterizes the VRPD, such that each vehicle is equipped with a drone, intended to deliver packages to customers $N = \{0, 1, 2, \ldots, n\}$. Vehicles and drones begin their mission from depot 0 and return to it after finishing the mission. VRP involves determining a set of routes that minimizes the total time of transportation.

The objective of VRPD is to find a tour of vehicles and drones allowing serving all customers so that at the end of the mission, all vehicles and drones are at the depot and that the total delivery time is minimized. As part of this work, we accept:

- A drone can start delivery and finish it either at a customer or at the depot.
- The drone moves at a higher speed than that of vehicles.
- A drone has the unit capacity and limited autonomy that allows them to serve only customers who are within the range of drones.
- A drone can serve a maximum of one customer on each flight.
- The drone loading time for new delivery is negligible.
- The drone must return to its own vehicle after delivery.
- During a drone delivery, the vehicle travels.
- If the drone serves the customer, the vehicle cannot visit him.

Also, the environment in which drones are used is considered safe. The interesting aspect of VRPD is that it uses the respective and complementary advantages of each type of delivery means.

3.1. Indices

i, j, k, m: Represent customers and depot index
d: Represent drone index
v: Represent vehicle index
$M = T_{\max}$: A maximum operational time.

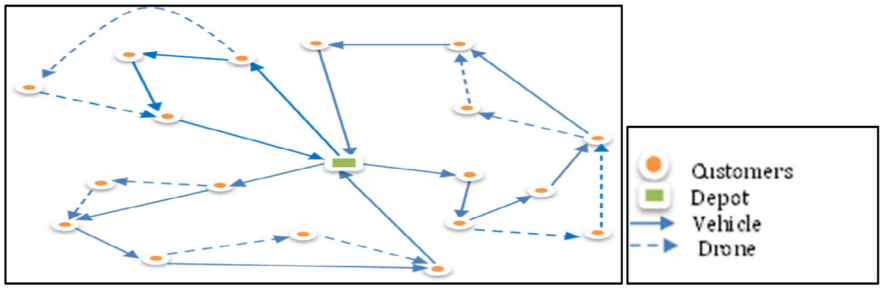

Fig. 2. VRP with drone.

Table 1
Research papers on drone delivery systems.

Reference	Type	Vehicles number	Drones number	With/without Synchro-nization	Formula-tion	Techniques
Murray and Chu [12]	FSTSP	1	1	With	MILP	Heuristic
	PDSTSP	1	N	without	MILP	Heuristic
Wang et al. [17]	VRPD	M	N	With		
Agatz et al. [24]	TSP-D	1	1	With	IP	Route first — Cluster second
Ha et al. [25]	Min cost TSP-D	1	1	With	MILP	GRASP
Dorling et al. (2017)	DDPs	0	N	without	MILP	Simulated annealing
Marlin et al. (2018)	SDDPHF	M	N	without		Adaptive dynamic programming
Schermer et al. [26]	VRPDERO	M	N	without	MILP	VNS/Tabu search algorithm
Bouman et al. [13]	TSP-D	1	1	with		Dynamic Programming
Ha et al. [27]	TSP-D	1	1	with		Genetic algorithm
Ulmer and Thomas [20]	VRP	M	N	with		Policy Function Approximation
Chang and Lee [15]	FSTSP	1	N	with	LP	
Schermer et al. [18,19]	VRPD	M	N	with		Route-first Cluster-second
Poikonen et al. [28]	VRPD	M	N	with	–	–
Sacramento et al. [22]	VRPD	M	N	with	MIP	Adaptive Large Neighborhood Search metaheuristic
Schermer et al. [29]	VRPD	M	N	with	MILP	heuristic
Wang and Sheu [21]	VRPD	M	N	with	MIP	branch and price algorithms
Moeini and Salewski [30]	TDA-RP	**1**	**N**	**with**		**Genetic Algorithm**

Where:
FSTSP: The Flying Sidekick Traveling Salesman Problem.
PDSTSP: Parallel Drone Scheduling Traveling Salesman Problem.
VRPD: Vehicle Routing Problem with Drone.
TSP-D: Traveling Salesman Problem with Drone.
DDPs: Drone Delivery Problems.
SDDPHF: Same-Day Delivery Problem with a Heterogeneous Fleet.
VRPDERO: Vehicle Routing Problem with Drones and En Route Operations.
TDA-RP: Truck-Drone-ATV Routing Problem.
MILP: Mixed Integer Linear Programming.
MIP: Mixed Integer Programming.
IP: Integer Programming.
LP: Linear Programming.

3.2. Sets

C: set of nodes $C = \{0, 1, \ldots, N\}$
M: a set of vehicles $M = \{1, \ldots, V\}$,
U: set of drones $U = \{0, 1, \ldots, D\}$,

3.3. Parameters

t_{ij}: Time traveled by a vehicle v from node i to node j
t'_{ij}: Time traveled by a drone d from node i to node j
T_i^v: Total time traveled at node i by vehicle v
q_i: The demand for a customer i
Q^v: The vehicle's load capacity
E: The drones' flight endurance
t_i^v: arrival time of a vehicle v at node (customer) i
t_i^d: arrival time of a drone d at node (customer) i

3.4. Decision variables

$$x_{ij}^v = \begin{cases} 1 & \text{if a vehicle } v \text{ moves from customer } i \text{ to customer } j \\ 0 & \text{otherwise} \end{cases}$$

$$z_{ijk}^{dv} = \begin{cases} 1 & \text{if the drone d associated with vehicle } v \text{ leaves from } i \\ & \text{serves } j \text{ and meet the vehicle in } k \\ 0 & \text{otherwise} \end{cases}$$

3.5. Mathematical model

In this section, we give a MILP formulation for VRP with drones. Our mathematical model is given as follows:

$$Min\ w = \sum_{i=0}^{N}\sum_{\substack{j=0\\ i\neq j}}^{N}\sum_{v=1}^{V} t_{ij}x_{ij}^{v} + \sum_{i=0}^{N}\sum_{j=1}^{N}\sum_{\substack{k=0\\ i\neq j\neq k}}^{N}\sum_{d=0}^{D}\sum_{v=1}^{V}(t'_{ij}+t'_{jk})z_{ijk}^{dv} \quad (1)$$

The objective function is to minimize the total traveled time of vehicles and drones. We form the constraints into different equations. The constraints (2) ensure that every customer should be visited either by a vehicle or a drone once and only once.

$$\sum_{i=0}^{N}\sum_{v=1}^{V} x_{ij}^{v} + \sum_{d=1}^{D}\sum_{v=1}^{V}\sum_{i=0}^{N}\sum_{k=0}^{N} z_{ijk}^{dv} = 1 \quad \forall j \in N\backslash\{0\} \quad (2)$$

Constraint (3) guarantees that all the vehicles must depart from the depot at most once. Correspondingly, constraint in (4), confirms that all the vehicles must return to the depot at most once.

$$\sum_{j=1}^{N} x_{0j}^{v} \leq 1 \quad \forall v \in M \quad (3)$$

$$\sum_{i=1}^{N} x_{i0}^{v} \leq 1 \quad \forall v \in M \quad (4)$$

The constraints (5) ensures that a vehicle v will arrive and leave the customer j

$$\sum_{\substack{i=0\\ i\neq j}}^{N} x_{ij}^{v} = \sum_{\substack{k=0\\ k\neq j}}^{N} x_{jk}^{v} \quad \forall v \in V,\ j \in N \quad (5)$$

Constraints (6) and (7) restricts a drone to leave a customer i and arrive at the customer k at most once.

$$\sum_{v=1}^{V}\sum_{j=1}^{N}\sum_{k=0}^{N} z_{ijk}^{dv} \leq 1 \quad \forall d \in D,\ i \in N \quad (6)$$

$$\sum_{v=1}^{V}\sum_{i=0}^{N}\sum_{\substack{j=1\\ i\neq j}}^{N} z_{ijk}^{dv} \leq 1 \quad \forall d \in D,\ k \in N \quad (7)$$

Constraint (8) ensure that the vehicle capacity must be respected

$$\sum_{i=0}^{N}\sum_{\substack{j=0\\ i\neq j}}^{N} q_i\, x_{ij}^{v} + \sum_{j=1}^{N}\sum_{\substack{i=0\\ i\neq j}}^{N}\sum_{k=0}^{N} q_j\, z_{ijk}^{vd} \leq Q^{v} \quad \forall v \in V \quad (8)$$

Constraints (9) require that each drone does not exceed the total endurance of battery E.

$$\sum_{j=1}^{N}\sum_{i=0}^{N}\sum_{\substack{k=0\\ i\neq j\neq k}}^{N}(t'_{ij}+t'_{jk})Z_{ijk}^{vd} \leq E \quad \forall v \in V,\ d \in D \quad (9)$$

$$2z_{ijk}^{vd} \leq \sum_{\substack{m=0\\ m\neq i}}^{N} x_{mi}^{v} + \sum_{\substack{n=0\\ n\neq i}}^{N} x_{nk}^{v} \quad \forall i,k \in N,\ j \in N\backslash\{0\},\ v \in V,\ d \in D \quad (10)$$

$$z_{Ojk}^{vd} \leq \sum_{\substack{m=0\\ m\neq k}}^{N} x_{mk}^{v} \quad \forall j,k \in N\backslash\{0\},\ v \in V,\ d \in D \quad (11)$$

Constraints (10) and (11) guarantee that if a drone starts from a customer i that can be the depot, and goes to customer k, the vehicle must visit these two customers.

The elimination of sub tours is imposed in constraints (12) where it prevents sub-cycles for each vehicle.

$$T_j^{v} - T_i^{v} + t_{ij} \geq t_{ij}(1 - x_{ij}^{v}) \quad \forall i,j \in N,\ v \in V,\ i \neq j \quad (12)$$

The set of constraints ((13)–(16)) ensure that a vehicle and drone will have a synchronous time in both take-off and landing customers. The first two ((13), (14)) synchronize the arrival at customer i, while the constraints ((15), (16)) synchronize the arrival at customer k. Here, the big M equal to $T_{\max}$, since the biggest difference (for non-bounding cases) in any schedule for the VRPD is less than the maximum operational time.

$$t_i^{d} \leq t_i^{v} + M\Big(1 - \sum_{j=1}^{N}\sum_{\substack{k=0\\ j\neq k}}^{N}\sum_{d=1}^{D}\sum_{v=1}^{V} z_{ijk}^{dv}\Big) \quad \forall i \in N \quad (13)$$

$$t_i^{d} \geq t_i^{v} - M\Big(1 - \sum_{j=1}^{N}\sum_{\substack{k=0\\ j\neq k}}^{N}\sum_{d=1}^{D}\sum_{v=1}^{V} z_{ijk}^{dv}\Big) \quad \forall i \in N \quad (14)$$

$$t_k^{d} \leq t_k^{v} + M\Big(1 - \sum_{j=1}^{N}\sum_{\substack{i=0\\ j\neq i}}^{N}\sum_{d=1}^{D}\sum_{v=1}^{V} z_{ijk}^{dv}\Big) \quad \forall k \in N \quad (15)$$

$$t_k^{d} \geq t_k^{v} - M\Big(1 - \sum_{j=1}^{N}\sum_{\substack{i=0\\ j\neq i}}^{N}\sum_{d=1}^{D}\sum_{v=1}^{V} z_{ijk}^{dv}\Big) \quad \forall k \in N \quad (16)$$

Constraints (13) and (14) state that the departure time of drone and vehicle must be the same. Similarly, constraints (15) and (16) ensure that the arrival time of both vehicle and drone will be the same when they merge at the same node.

$$x_{ij}^{v},\ z_{ijk}^{vd} \in \{0, 1\}$$

4. Solution methodology for the VRPD

The VRP with drones corresponds to the NP-HARD type, therefore, necessary to develop and implement an appropriate methodology for its solution; in this case, it is decided to use the genetic algorithm with sweep local research as the main tool metaheuristic to establish the routes of vehicles and drones.

4.1. Hybrid genetic algorithm for the VRP with drones

This section proposes a modified hybrid genetic algorithm combined with a set of heuristic techniques that make it possible to solve the VRP with drones (VRPD) by using the variant, which works with a heterogeneous fleet and several drones. The different stages of our hybrid genetic algorithm are described as follows:

Step 1. Vehicle Routing Problem with Drones representation

Step 2. Generation of the initial population (200 individuals) with a heuristic constructive method.

Phase 1: customers are divided between vehicles and the drone fleet

Phase 2: manages the routing optimization, which consists of solving 2 NP-hard problems:

- an m-TSP (Multiple Traveling Salesman Problem) for vehicles;
- a PMS (Parallel Machine Scheduling problem) for drones (when M ≥ 1).

Phase 3: Route generation and division

Step 3. Selection

Step 4. Recombination

Step 5. Mutation

Step 6. Sweep local search algorithm

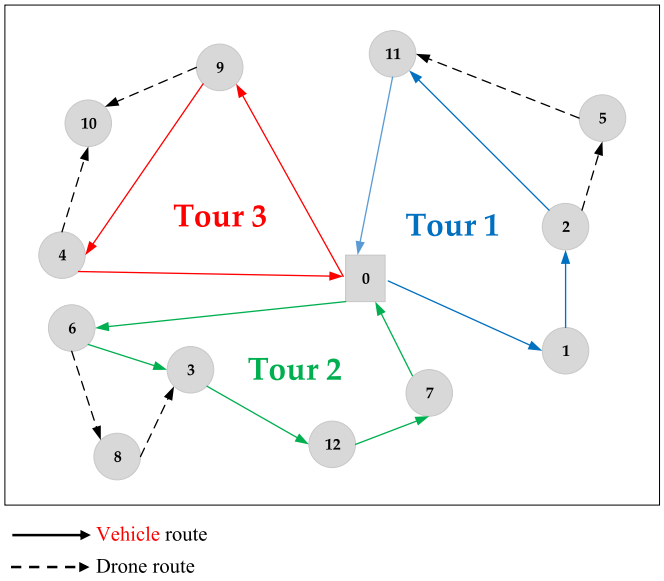

Fig. 3. Graphic of a possible solution for a VRPD.

4.1.1. VRPD representation

The representation of genetic solutions and operators in the GA is a principle stage in our algorithm. The first genetic algorithms all used a binary representation where each solution is coded in the form of a string of bits of length n. One of the proposals made by this paper focuses on the use of efficient coding, which aims to reduce the computational effort and balance the use of memory. Fig. 3 shows an example of a possible solution for a VRPD. The node (0) signifies the depots however the rest nodes are the customers. The routes fitting to the delivery of the vehicles are characterized by solid lines, whereas the routes belonging to the drone's delivery are depicted as dotted lines.

4.1.2. Initial population: Heuristic constructive

We initialize the GA by generating solutions for the first generation until we reach a maximum population size of P. Mainly, a sub-algorithm has been developed which divides the initial population into two different parts, which can be modified at the start of the algorithm to allow flexibility in the method, so that it is possible to test the different variations between the size of the population and the size of the fractions that make it up.

The next step was the implementation of a constructive algorithm capable of calculating the minimum route between customers and another which made it possible to add a random factor to the population. On this basis, a nearest neighbor heuristic procedure and a savings algorithm were developed, where the management of randomization was modified to grant diversity to the initial population. For the population filling step of the genetic algorithm, it was proposed to take each of the heuristics and use them as follows:

Nearest neighbor heuristic

The nearest neighbor heuristic is one of the techniques with the best balance between good results and computation time because it aims to connect the nodes closest to the node in which it is positioned at this time and then repeat the same procedure with the node connected, until the end of the visit of all the available nodes. The construction sequence of this method has been ordered as follows:

i. Assign an identifier to customers relative to their position in the table of positions of the instance, taking as their starting point the first customer served by a drone or a vehicle.
ii. Find the nearest neighbors to the starting point.
iii. Modify the starting point of the next customer in the list and perform step 2 until generating enough individuals to fill the corresponding fraction in the initial population.

Modified savings heuristic

As a preventive method to prevent the genetic algorithm from being quickly trapped in the local optimum, a savings algorithm such as that proposed in the literature has been implemented, which has been modified based on the initial solution algorithm. proposed with a variant in the random section of the method, to add great diversity to the population. The specialty of this heuristic lies in the ability to deceive the savings method, having a series of random factors capable of modifying the distance between customers so that the procedure takes distances from the point analyzed to other non-visited customers and multiply them by a series of random variables that allow the algorithm to see certain points closer or further from what they are. This selection technique is linked to customer capacity so that customers who do not damage the maximum capacity limit of the vehicle or the endurance of the drone are added.

Once the process of generating individuals is completed, the population matrix is browsed and each of the sequences is divided into routes according to the maximum capacity of the vehicle and the endurance of the drone (Capacity). The following Figs. 4 and 5 describe the steps of this method with a drone tour and a vehicle tour, R_c represents the solution of the m-TSP problem (order in which customers must be visited), $q^v = 10, q^d = 2$ represent the capacities of vehicle and drone respectively. The vector J^v, J^d stores the customer number where the route separation process begins, whether for the vehicle or the drone. In the example given below, we start at customer 1 for the vehicle and customer 8 for the drone. The result of the route separation process is stored in the same R_c vector which contains the ordered customers.

4.1.3. Selection

Unlike other optimization techniques, genetic algorithms do not require any particular hypothesis on the regularity of the objective function. In particular, the genetic algorithm does not use its successive derivatives, which makes its field of application very broad. No continuity assumption is required either. However, in practice, genetic algorithms are sensitive to the regularity of the functions they optimize. The few assumptions required allow us to deal with very complex problems. The function to be optimized can thus be the result of a simulation.

Selection helps to statistically identify the best individuals in a population and eliminate the bad ones. We find in this subsection our simple procedure to select the best individual basis on the objective function.

The criteria used in the selection process are the fitness function criteria. Each route in the initial population is calculated by the distance, the value of fitness, the fitness probability, and the cumulative probability of fitness. The steps for calculating the physical condition are:

a. Find the time traveled of each route w_i
b. Find the total traveled time of the entire route

$$w = \sum_{i=0}^{n}\sum_{\substack{j=o\\ i\neq j}}^{n}\sum_{v=1}^{v} t_{ij}x_{ij}^{v} + \sum_{i=0}^{n}\sum_{j=1}^{n}\sum_{\substack{k=0\\ i\neq j\neq k}}^{n}\sum_{d=1}^{m}\sum_{v=1}^{v}(t'_{ij}+t'_{jk})z_{ijk}^{dv}$$

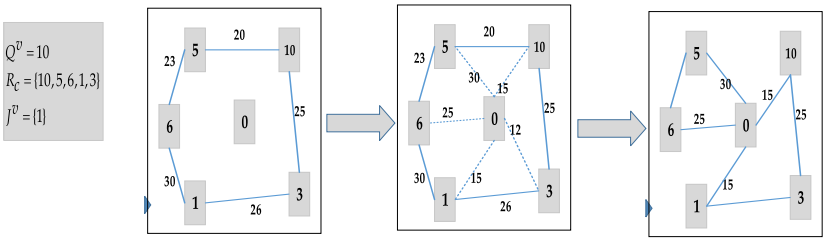

Fig. 4. Division of the big tour into itineraries for vehicles.

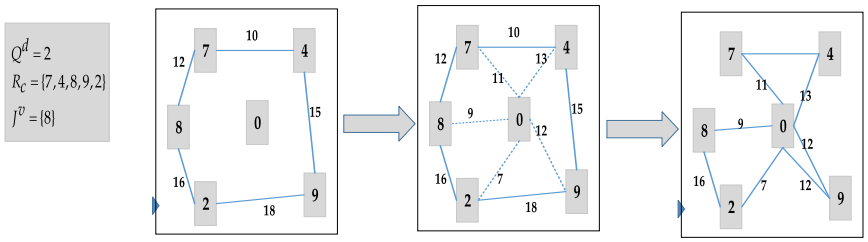

Fig. 5. Division of the big tour into itineraries for drones.

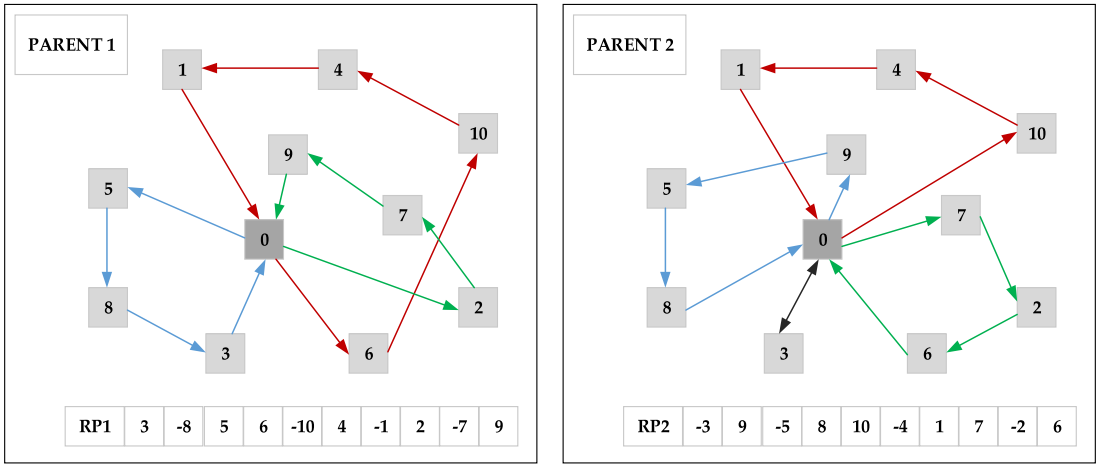

Fig. 6. Individuals recombination.

c. Find the value of each physical route

$$(f_i = \frac{\sum_{i=0}^{n}\sum_{\substack{j=o\\ i\neq j}}^{n}\sum_{v=1}^{v} t_{ij}x_{ij}^{v} + \sum_{i=0}^{n}\sum_{j=1}^{n}\sum_{\substack{k=0\\ i\neq j\neq k}}^{n}\sum_{d=1}^{m}\sum_{v=1}^{v}(t'_{ij}+t'_{jk})z_{ijk}^{dv}}{w_i})$$

d. Looking for a Total Fitness $\sum_{i=1}^{N} f_i$

e. Look for fitness probabilities for each route$p_i = \frac{f_i}{\sum_{i=1}^{N} f_i}$

f. Locate the cumulative probabilities of each route $q_i = \sum_{k=1}^{i} p_k$

Besides, the selection of a route that produces the next population is taking N random numbers r and comparing these with random numbers with the cumulative probability of fitness for each route.

4.1.4. Recombination: Crossover operator

As part of the work, several recombinations have been analyzed for instance single point recombination, two-point recombination, and multiple point recombination. Specialized recombination has also been analyzed when routes or parts of routes are exchanged between individuals. To reserve the good quality qualities, existent in the parents, a recombination step based on a process of combining the routes of each of the parents is used so that the resulting child contains segments of paths existing in each of them. To do this, the following procedure is performed:

(a) In parent 1, randomly select one of their routes and mark this route so that it will no longer be selected in the future.
(b) Copy the selected route to the descendant, making sure that the customers of the selected route are not already in the descendant. If they already exist, delete this customer

from the route to be added. If all the customers are on the downside.

(c) At the end of the route marker, change the negative sign to positive.

(d) In parent 2, randomly select one of their routes and mark this route so that it will no longer be selected in the future.

(e) Copy the selected route to the descendant, making sure that the customers of the selected route are not already in the descendant. If they already exist, delete this customer from the route to be added. If all the customers are on the downside.

(f) At the end of the route marker, change the negative sign to positive.

(g) Go back to step a.

Fig. 6 Describes the recombination of individuals.

4.1.5. Mutation

The descendants obtained in the previous step, which no longer contain routes, are subject to a variable number of simple mutations, which are commonly known as swaps or shift movement. These movements allow both the movement of intra- and inter-channel customer routes, as performed on the resulting m-TSP recombination. Following the mutation of the offspring, begins a process manager in the division of the sequence into routes taking into account the capacity of vehicles and drones. In Fig. 7 we show the first mutation represented by a swap movement. Fig. 8 describes the shift mutation movement for the VRPD.

4.1.6. A sweep local search heuristic

The sweep algorithm is considered to be a heuristic method that seeks to solve problems in two distinct stages: the first aims to group demand points according to a proximity criterion; while in the second step, each group is resolved independently. The sweep method is a fairly simple procedure for solving problems with vehicle routing.

This heuristic breaks down VRPD into a part of the representation, clustering, regrouping, and planning phase. In our sweep algorithm we consist of two stages: clustering and route generation; in the first part we divide the customers into different clustering, then a route generation phase using constructive heuristics is applied aimed to link all customers in every cluster starting from and ending to the same depot. The methodology consists of a series of steps, described below:

Representation: Graphical representation with the location of delivery points (customers) and the depot. The following figure Fig. 9 shows the representation of all customers in the 10.10.1 instance

Clustering: We draw a straight line from the depot and rotate it until it crosses one of the delivery points. The procedure must, therefore, continue successively. At each stop, it is necessary to check if the vehicle can meet local demand, if the increase in demand exceeds the capacity of the vehicle or the drone or exceeds the endurance of the drone, the point should be ignored and a new route must be started at this point using the same procedures. When all the points are contained in a route, this step is completed (see Fig. 10).

The depot is the origin of the benchmark, and the first group created consists of the 5 customers in blue ({7-8-9-10-1}). Two other groups were then formed. The transition from one group to another is due to a violation either in the endurance of the drone or the capabilities of the vehicle and drone.

Planning: The last step is to organize the sequence of stops for each route by applying all the algorithm that solves the problem of vehicle routing with drones (see Fig. 11).

The pseudo-code of hybrid GAxSweepLS is described in algorithm 1

Algorithm 1: Hybrid GAxSweepLS

1: P=200 (population size)

2: S_1^0: Nearest Neighbors initial solution

3: S_2^0: Modified Savings initial solution

4: S : Best solution; F: Fitness value

5: Find x; $f(x) = Min\, f(x_{Neighbor}, x_{Savings})$

6: **Begin**

7: i=0;

8: P=population (i);

9: $S_1^0 = S_1^0 \cup$(Nearest_Neighbor)

10: $S_2^0 = S_2^0 \cup$(Modified_Savings)

11: **While (itermax do not met) do**

12: R : Select (pop(i))

13: Calculate $\sum_{i=1}^{N} f_i$, $p_i = f_i / \sum_{i=1}^{N} f_i$

14: C : Crossover operator (S)

15: pop=C

16: M: Mutation operator (pop(i))

17: **For** $(i \le (pop)\ to\ n \le 1)$ **do**

18: S_3 = Sweep algorithm (pop(i))

19: Representation

20: Clustering

21: Planning

22: **If** $(F(S_3) > F(S))$ then

23: $S = S_3$

24: **End If**

25: **End For**

26: i=i+1

27: **End While**

28: End

5. Computational experiments

As a final stage, computational experiments are carried out to analyze the performance, behavior, and contribution of the hybrid GAxSweepLS with the sweep algorithm as an improvement stage, proposed as a general solution technique for the VRP with a drone. The results are compared in the existing instances in the specialized literature which are known as the best answer found by other authors using different algorithms proposed for the same problem.

5.1. Benchmarks instances

For the purpose to test the effectiveness of our hybrid GA to solve the VRPD, several instances taken from the literature and simulations are performed in several cases with a variable

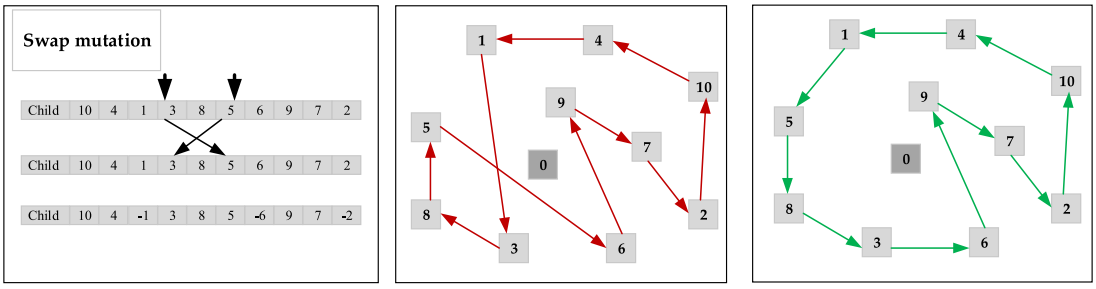

Fig. 7. Swap movement mutation.

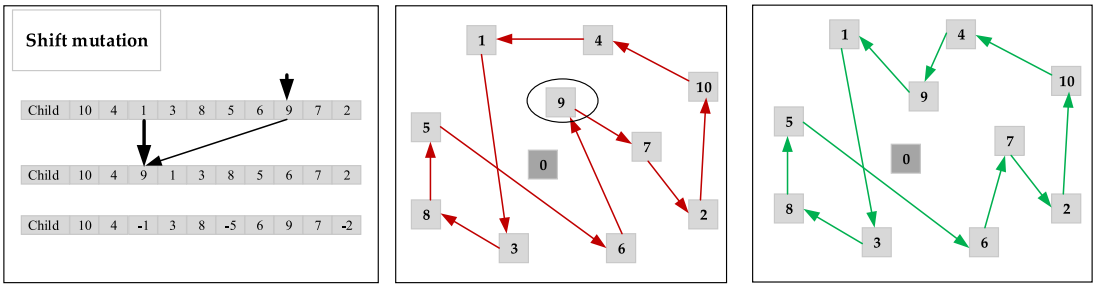

Fig. 8. Shift movement mutation.

number of points. To fairly compare the computing times, we apply a ratio depending on the power of the machines. We refer to the work of Dongarra [31] which measures the performance of many microprocessors.

Two types of instances are used to test the performance of the GAxSweepLS. The first set corresponds to the instances of Sacramento et al. [22] where the data corresponding to 112 instances corresponding to different scenarios for the VRPD.

The second set represents our benchmarks instances simulation. The data correspond to 38 instances corresponding to different scenarios for the VRPD. Each instance is named INS_JA_n_k, where n is the number of customers, k represents the number of drones. The first line of each instance file indicates the number of customers in a specific instance. The second line indicates the number of drones. The third line indicates the homogeneous capacity vehicle. The fourth line describes the characteristics of each customer, where it is written in the following way:

XCOORD, YCOORD, DEMAND, READY TIME, DUE TIME, and SERVICE TIME.

Finally, we provide information about the characteristics of each drone, where it is written as follows:

DRONE.NO, ENDURANCE, CAPACITY, DRONE FIXED COST, and DRONE VARIABLE COST. The benchmarks instances are available at https://sites.google.com/view/euchi-jalel/instances.

5.2. Experiment design

The HGA was implemented in the Visual Studio C ++ application, 64 bits (win64), and is run on Intel® Core™ i5-2450M @ 250 GHz and 4 GB of RAM. The HGA parameter setting based on Table 2 uses a constant population size of 200 populations. Elite probability 0.1; 0.15; 0.2; 0.25, mutant probability 0.1; 0.2; 0.3 and the probability of an elite line 0.5; 0.6; 0.7; 0.8. The configuration parameter settings produce 48 parameters, each parameter calculated up to 50 samples of fitness values with the number of iterations is 700 repetitions and the average fitness generated in each selected parameter is a parameter with the average fitness value the lowest of all parameters. Table 2 describes the parameter settings of the HGA.

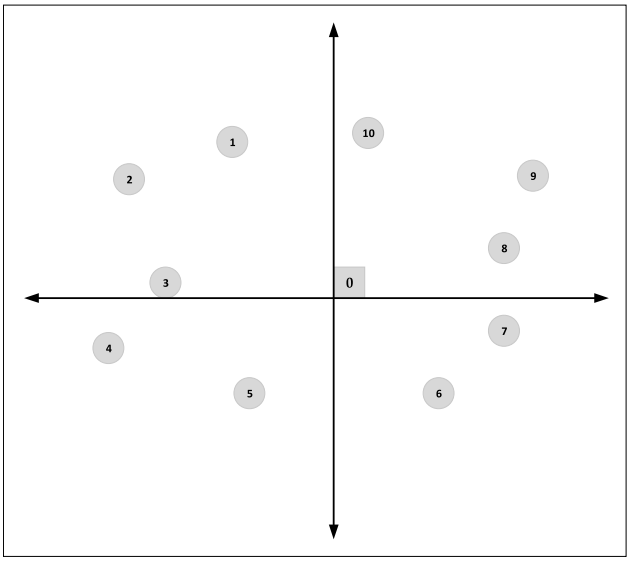

Fig. 9. Step 1 in the sweep algorithm.

Table 2
Parameters setting of the Hybrid sweep genetic algorithm.

Parameter	Value
Population size: P	200
Mutation rate: r	0.8
Drone served (chromosome random)	0.5
Generation number	1000, 1500
Stopping criteria: itermax	50

5.3. Discussions

To better understand the use of hybrid genetic sweep algorithms in finding solutions to vehicle routing problems with a drone, a benchmarks instances of Sacramento et al. [22] and simulations are carried out in several cases with a varying number of points. The proposed algorithm is compared with the results of

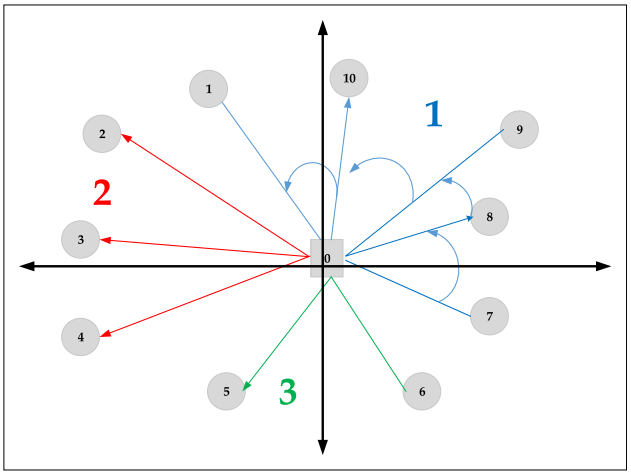

Fig. 10. Step 2 in the sweep algorithm.

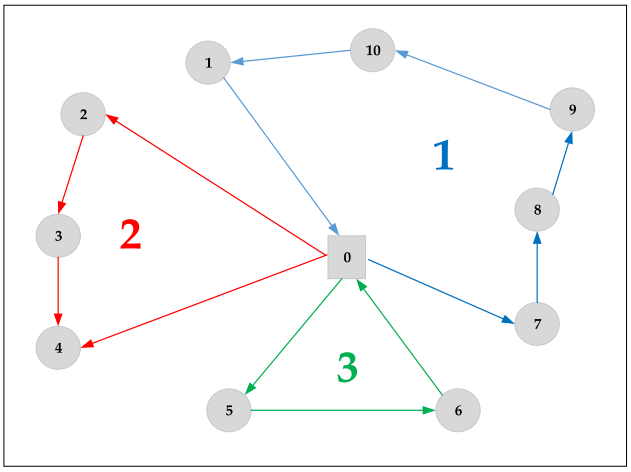

Fig. 11. Illustration of the second phase of the sweep algorithm.

Sacramento et al. [22]. To evaluate and compare the performance of the algorithms, a subset of 36 instances defined as a small benchmark, a subset of 44 problems used as a medium instance, and a subset of 32 problems defined as a large instance was used to evaluate the performance of H-Sweep-GA algorithm.

Tables 3–5 present a comparison between the proposed hybrid genetic algorithm (GAxSweepLS) and the Adaptive large neighborhood search (ALNS) algorithm of Sacramento et al. [22]. Both algorithms were tested with the same instances proposed and are compared with the best answers published in the specialized literature (Best Known Solution BKS) to measure their performance and effectiveness in the search for the optimal response. In the first three columns, the name of the instance used followed by the number of customers (Nb.Cust), the BKS of each problem followed by the results is shown in order obtained from the tests performed on the ALNS and GAxSweepLS (with 100 and 1500 generations) algorithms, accompanied by the GAP between the BKS and the best solution achieved by our solution methodology.

In Table 3 we remark that our algorithm produces the best result as the result of the MILP. which has a time limit of 22.178 s. We show that when we have augmented the number of generations, HGA gives the optimal solution with a minimum difference of time-limit equal to 2.904 (22.178–19.274) seconds.

In Table 3 we remark that the GAxSweepLS outperform the ALNS algorithm in total traveled time and give the optimal solution in all small benchmark instances up to 12 customers.

Table 4 summarizes the performance measures of the two alternative strategies above, namely GAxSweepLS with 1000 generations and the GAxSweepLS with 1500 generations for medium-sized instances. Some observations are made. First, the algorithm with 1500 generations gives 37 good solutions of 44 given instances while comparing it with the best solution found. We also notice that the GAxSweepLS algorithm makes considerable improvements when the generation size is increased. For medium-sized instances, our algorithm was able to solve the problem with an almost short execution time with a time limit of 70.392 s.

The GAxSweepLS algorithm obtained the best results, now in 37 of the 44 instances, with a 1, 35832 gaps of the best-known solution in the literature. Table 4 shows a summary of the performance of the GAxSweepLS algorithm. GAxSweepLS also performed well in instances with a small and medium number of customers. The result closest to the best-known solution was obtained by GAxSweepLS, while we use several generations equal to 1000 both in the best and average objective.

Table 5 displays a computational result for large instances. In this table, we show that the GAxSweepLS gives a good and best solution compared with the best-known solution founded by Sacramento et al. [22]. From 32 large instances, a 27 new best solution is founded.

A summary of the results obtained is shown in Table 5, the average of the objective function is given with 1000 and 1500 generation for the GAxSweepLS algorithm. It is noted in Table 5, that as well as mentioned above, the best results are obtained with a gap respected to the best-known solution found by the ALNS algorithm of Sacramento et al. [22]. It is noted that GAxSweepLS metaheuristics show results that are very competitive, taking into account the generation of initial feasible solutions. It is also observed that the GAxSweepLS algorithm yields better results most of the time and presents a better average in terms of the difference in the solutions concerning the optimal one. This algorithm was designed with a sweep local search in mind, this was done because the initial feasible solutions obtained from the algorithms proposed in this work (nearest neighbor and Modified savings heuristic yielded competitive results so that these are explored in wide way search regions. So, what is sought with this improvement algorithm is to locally exploit the search for different solutions trying to improve to a certain extent the different initial feasible solutions.

In Fig. 12 we have seen the best-known solution for the VRPD for an instance with 150 customers (scenario 150.40.4).

Table 6 display the computational experiments of the simulated instances given in this paper. We remark that the GAxSweepLS give a good solution in a promising time-limit equal to 65.785 s. We observe that when we increase the number of generations the GAxSweepLS algorithm takes more time but it gives the best solution comparing to the solution given within 1000 generation. In Fig. 13 we have seen the best-known solution for the VRPD for a simulated instance with 200 customers (scenario INS_JA_200_15).

6. Conclusions

This paper addresses a new problem of generating a tour with a drone which is, in fact, a new variant of the problem of vehicle routing.

The VRPD was formulated according to the need to route a heterogeneous fleet of drones and vehicles to deliver a set of customers from large areas. The mathematical model presented in the paper was tested by the IBM CPLEX solver which obtained 36 optimal solutions for instances of up to 12 with different scenarios. Their values were used as a parameter for comparison with the ALNS method of Sacramento et al. [22].

Table 3
Comparison results for small instances up de 12 customers.

Instances	Nb.Cust	BKS	ALNS ([22])			HGA with 1000 iterations			HGA with 1500 iterations			Mean deviation
			Objective	Average objective	CPU (s)	objective	Average objective	CPU (s)	OBJECTIVE	Average objective	CPU (s)	
06.05.1	6	1.09821	1.09821	1.09821	0.014	1.11345	1.11345	0.147	1.09821	1.09821	0.072	0
06.05.2	6	0.84215	0.84215	0.84215	0.001	1.21415	1.21415	0.135	0.84215	0.84215	0.148	0
06.05.3	6	1.21137	1.21137	1.21137	0.001	1.54326	1.54326	0.147	1.21137	1.21137	0.151	0
06.05.4	6	0.94599	0.94599	0.94599	0.003	0.98949	0.98949	0.100	0.94599	0.94599	0.158	0
06.10.1	6	2.40611	2.40611	2.40611	0.004	2.69023	2.69023	0.054	2.40611	2.40611	0.163	0
06.10.2	6	1.67027	1.67027	1.67027	0.002	1.67027	1.67027	0.046	1.67027	1.67027	0.170	0
06.10.3	6	1.32552	1.32552	1.32552	0.003	1.32552	1.32552	0.104	1.32552	1.32552	0.101	0
06.10.4	6	1.44307	1.44307	1.44307	0.001	1.44307	1.44307	0.004	1.44307	1.44307	0.018	0
06.20.1	6	2.67759	2.67759	2.67759	0.11	2.67759	2.67759	0.134	2.67759	2.67759	0.192	0
06.20.2	6	4.31959	4.31959	4.31959	0.054	4.31959	4.31959	0.064	4.31959	4.31959	0.072	0
06.20.3	6	3.82475	3.82475	3.82475	0.002	3.82475	3.82475	0.016	3.82475	3.82475	0.114	0
06.20.4	6	3.67872	3.67872	3.67872	0.001	3.67872	3.67872	0.049	3.67872	3.67872	0.151	0
10.5.1	10	1.65563	1.65563	1.65563	0.002	1.65563	1.65563	0.003	1.65563	1.65563	0.158	0
10.5.2	10	1.45185	1.45185	1.45185	0.339	1.45185	1.45185	0.467	1.45185	1.45185	0.163	0
10.5.3	10	1.47357	1.47357	1.47357	0.193	1.47357	1.47357	0.167	1.47357	1.47357	0.170	0
10.5.4	10	1.28489	1.28489	1.28489	0.002	1.28489	1.28489	0.007	1.28489	1.28489	0.011	0
10.10.1	10	232647	232647	232647	0.026	232647	232647	0.067	232647	232647	0.087	0
10.10.2	10	3.15856	3.15856	3.15856	0.075	3.15856	3.15856	0.167	3.15856	3.15856	0.192	0
10.10.3	10	2.55274	2.55274	2.55274	0.427	2.55274	2.55274	0.607	2.55274	2.55274	0.075	0
10.10.4	10	2.53931	2.53931	2.53931	0.008	2.53931	2.53931	0.007	2.53931	2.53931	0.014	0
10.20.1	10	4.45240	4.45240	4.45240	3.946	4.45240	4.45240	2.177	4.45240	4.45240	2.580	0
10.20.2	10	6.16776	6.16776	6.16776	0.011	6.16776	6.16776	0.029	6.16776	6.16776	0.166	0
10.20.3	10	4.54630	4.54630	4.54630	1.197	4.54630	4.54630	0.988	4.54630	4.54630	0.172	0
10.20.4	10	6.15355	6.15355	6.15355	49.170	6.15355	6.15355	19.274	6.15355	6.15355	22.178	0
12.5.1	12	1.37381	1.37381	1.37381	31.444	1.57163	1.57163	12.056	1.37381	1.37381	15.187	0
12.5.2	12	1.05899	1.05899	1.05899	1.110	1.05899	1.05899	1.298	1.05899	1.05899	1.9300	0
12.5.3	12	1.44765	1.44765	1.44765	0.028	1.57166	1.57166	0.028	1.44765	1.44765	0.199	0
12.5.4	12	1.58100	1.58100	1.58100	0.100	1.89216	1.89216	0.08	1.58100	1.58100	0.055	0
12.10.1	12	2.68103	2.68103	2.68103	81.447	2.89612	2.89612	45.807	2.68103	2.68103	50.106	0
12.10.2	12	2.68420	2.68420	2.68420	0.059	2.71910	2.71910	0.023	2.68420	2.68420	0.126	0
12.10.3	12	2.88048	2.88048	2.88048	0.030	3.14414	3.14414	0.026	2.88048	2.88048	0.132	0
12.10.4	12	2.31418	2.31418	2.31418	0.011	2.441456	2.441456	0.010	2.31418	2.31418	0.137	0
12.20.1	12	5.77759	5.77759	5.77759	0.272	6.026940	6.026940	0.193	5.77759	5.77759	0.145	0
12.20.2	12	8.27254	8.27254	8.27254	0.004	8.69104	8.69104	0.002	8.27254	8.27254	0.164	0
12.20.3	12	4.16693	4.16693	4.16693	0.054	4.93107	4.93107	0.04	4.16693	4.16693	0.174	0
12.20.4	12	6.08859	6.08859	6.08859	0.210	6.10783	6.10783	0.207	6.08859	6.08859	0.281	0

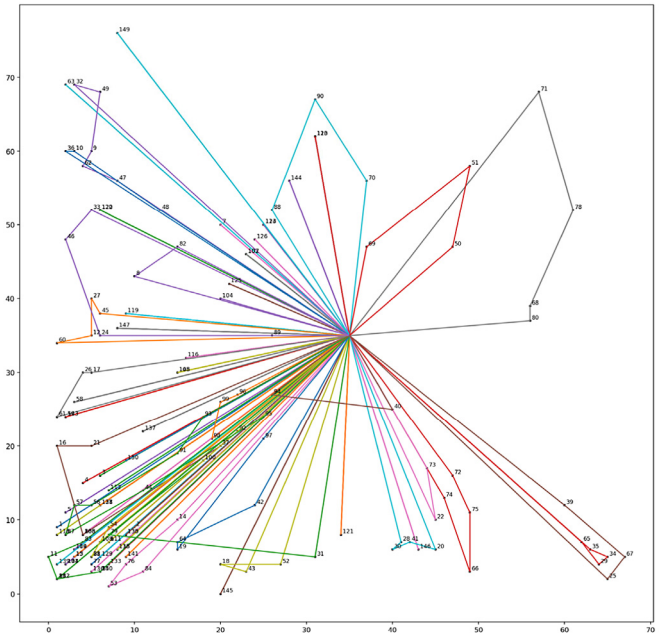

Fig. 12. The best-known solution for the VRPD (150.40.4 instance).

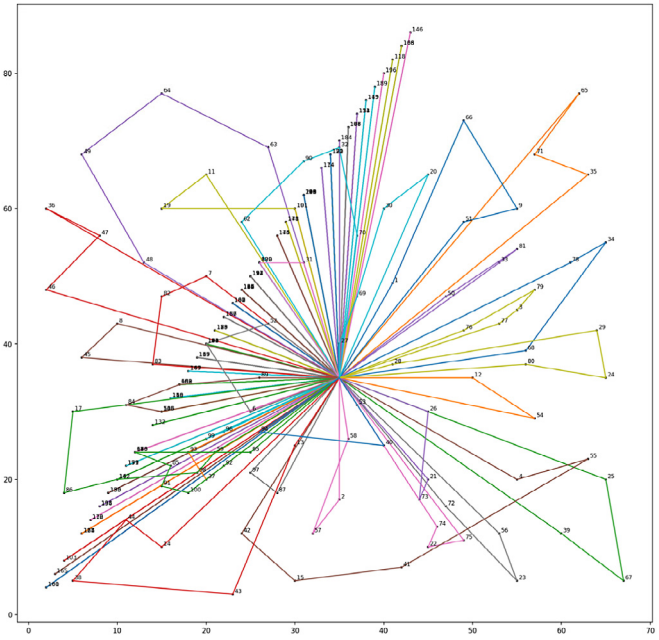

Fig. 13. The best-known solution for the VRPD (INS_JA_200_15 instance).

Table 4
Comparison results for medium instances up de 100 customers.

Instances	Nb.Cust	BKS	ALNS ([22])		HGA with 1000 generations			HGA with 1500 generations			Mean deviation
			Objective	Average objective	objective	Average objective	CPU (s)	objective	Average objective	CPU (s)	
20.5.1	20	1.35849	1.79347	1.79347	1.44149	1.41747	0.226	1.35849	1.578949	0.073	0
20.5.2	20	1.35789	1.95401	1.95401	1.45162	1.93565	0.475	1.35789	1.578949	0.937	0
20.5.3	20	1.48658	1.48658	1.48658	1.86301	1.60058	0.673	1.578949	1.58930	1.148	0,0461845
20.5.4	20	1.30794	1.37893	1.37893	1.31825	1.45813	0.799	1.30794	1.58193	1.156	0
20.10.1	20	2.98140	3.25253	3.25253	3.23428	3.45287	1.362	2.98140	3.28156	1.161	0
20.10.2	20	3.06000	3.08938	3.08938	3.06094	3.08846	2.052	3.06000	3.06000	2.168	0
20.10.3	20	3.29814	3.70226	3.72576	3.37973	3.70924	2.224	3.29814	3.29814	1.172	0
20.10.4	20	3.09811	3.30890	3.31367	3.33109	3.56987	2.607	3.09811	3.09811	1.177	0
20.20.1	20	6.64279	7.34453	7.35115	6.94012	7.06685	0.297	6.64279	7.06000	1.182	0
20.20.2	20	7.60000	7.54889	7.54889	10.1199	10.67181	0.326	7.60000	7.87016	1.185	0
20.20.3	20	7.46100	7.46100	7.47458	7.41095	7.49567	0.166	7.64279	7.60000	1.189	0,090895
20.20.4	20	6.90427	7.01331	7.01331	6.926275	6.90244	0.216	6.90427	6.90427	1.193	0
50.10.1	50	5.73421	5.86134	5.86134	5.93011	6.02673	0.240	5.73421	5.89230	1.201	0
50.10.2	50	5.44155	5.58493	5.62101	5.44155	5.47432	0.267	5.44155	5.47432	1.207	0
50.10.3	50	5.18016	5.42240	5.42546	5.18016	5.27549	0.293	5.18016	5.27549	1.212	0
50.10.4	50	5.07308	5.20834	5.35262	5.07308	5.19813	0.405	5.07308	5.19813	1.217	0
50.20.1	50	10.08218	10.45526	10.45635	10.08218	10.26358	0.859	10.08218	10.26358	1.221	0
50.20.2	50	10.02894	10.05611	10.05611	10.02894	10.04605	1.036	10.02894	10.04605	1.226	0
50.20.3	50	10.50871	10.54249	10.65703	10.50871	10.55667	1.070	10.50871	10.55667	2.230	0
50.20.4	50	9.35910	10.66415	11.00082	9.35910	9.70421	1.322	9.35910	9.70421	2.234	0
50.30.1	50	15.72352	15.81788	15.81788	15.72352	15.73795	2.135	15.72352	15.73795	3.240	0
50.30.2	50	14.9088	15.01482	15.46361	14.9088	14.99262	2.356	14.9088	14.99262	3.244	0
50.30.3	50	16.1284	16.76899	16.77134	16.1284	16.46805	2.761	16.1284	16.46805	3.250	0
50.30.4	50	18.28746	18.28746	18.28746	18.380197	18.95677	1.303	18.38019	18.95677	3.255	0,046365
50.40.1	50	20.30190	20.37508	21.17709	20.30190	20.30190	1.425	20.30190	20.30190	3.267	0
50.40.2	50	20.56229	20.62624	20.62624	20.56229	20.56229	1.522	20.56229	20.56229	3.276	0
50.40.3	50	22.632833	22.64523	22.70534	22.632833	22.99322	1.819	22.632833	22.99322	3.289	0
50.40.4	50	22.33708	22.33708	22.78912	22.34808	22.94614	1.953	22.34808	22.94614	3.298	0,0055
100.10.1	100	6.36332	6.85741	6.89015	6.59306	6.60714	2.001	6.36332	6.38590	3.306	0
100.10.2	100	7.37956	7.58505	7.67814	7.48619	7.58412	2.072	7.37956	2.451175	3.311	0
100.10.3	100	8.11945	7.18353	7.30551	8.11945	8.22664	2.222	8.11945	8.22664	3.320	0
100.10.4	100	6.89257	7.45675	7.54594	7.41056	7.54891	2.287	6.89257	7.04396	3.326	0
100.20.1	100	12.540971	13.60671	13.79462	13.02675	13.20438	2.674	12.540971	13.244306	3.332	0
100.20.2	100	14.01182	14.13399	14.53749	14.10273	14.61597	4.196	14.01182	14.597898	6.339	0
100.20.3	100	12.48793	13.70990	13.76722	13.361313	13.76162	4.701	12.48793	12.532536	6.345	0
100.20.4	100	12.13500	13.84944	14.19761	13.74248	13.88785	5.364	12.13500	12.21572	8.351	0
100.30.1	100	22.58818	22.58818	23.63641	23.49271	23.49293	5.770	23.568611	23.900604	6.356	0,4902155
100.30.2	100	22.03956	22.31432	22.38464	22.15947	22.17714	6.136	22.03956	22.13497	7.375	0
100.30.3	100	23.29882	23.71948	23.90941	23.426597	23.614716	6.379	23.29882	23.50325	70.392	0
100.30.4	100	22.14946	22.37011	22.65848	22.298585	22.563379	7.200	22.14946	0.505212	8.409	0
100.40.1	100	29.13966	29.13966	30.18073	29.89490	29.92377	7.614	29.13966	29.54630	9.427	0
100.40.2	100	30.36095	30.98999	31.20916	30.42939	31.03512	8.139	30.36095	30.41290	10.446	0
100.40.3	100	29.01123	29.02475	29.66526	29.01123	29.06346	9.016	29.01123	29.06346	10.473	0
100.40.4	100	28.59036	28.97348	29.20493	28.59036	28.63601	9.709	28.59036	28.63601	10.490	0

An adaptation of the genetic algorithm with a sweep technique as a local search (GAxSweepLS) was proposed as a solution methodology. To this end, several neighborhoods for inserting and swapping requests were developed to utilize the structure of the problem. The new algorithm proved to be more efficient in terms of solution quality and execution time by comparing it with the methods proposed in the literature.

We conclude that our approach brings very good results. The improvement in the objective reaches a good level. In the VRPD instances, our GAxSweepLS obtained honorable results and made improvements to the objective which could reach 84% in some cases. We also achieved an improvement of 69%, but on a solution generated by a GAxSweepLS algorithm with 1500 generation. All the solutions calculated are of very good quality: they respect the general constraints of VRP and the endurance and the capacity of the drones. The introduction of the modified saving heuristic makes it possible to respond effectively to the improvement of the initial solution.

Several future developments can now be envisaged. Since this is the first work on this subject, it would be interesting to design other metaheuristics exploiting, even more, the delivery in the presence of drones. It would also be interesting to study other variants of our problem. Thus, a problem where the delivery consists only of a fleet of aerial vehicles.

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.

Acknowledgments

We would like to thank the referees for carefully reading our manuscript and for giving such constructive, insightful comments and suggestions, which substantially helped to improve the quality of our paper.

References

[1] C. Qi, L. Hu, Optimization of vehicle routing problem for emergency cold chain logistics based on minimum loss, Phys. Commun. 40 (2020) 101085.
[2] W.C. Chiang, Yuyu. Li, J. Shang, T.L. Urban, Impact of drone delivery on sustainability and cost: Realizing the UAV potential through vehicle routing optimization, Appl. Energy 242 (2019) 1164–1175.

Table 5
Comparison results for large instances up de 200 customers.

Instances	Nb.Cust	BKS	A	LNS ([22])	HGA with 1000 iterations			HGA with 1500 iterations			Mean deviation
			Objective	Average objective	objective	Average objective	CPU (s)	objective	Average objective	CPU (s)	
150.10.1	150	8.79027	8.79027	8.93509	8.5865	8.64414	0.708	8.5865	9.63644	0.495	−0,20377
150.10.2	150	8.25905	8.25905	8.41602	8.14615	8.14615	0.812	8.14615	10.90093	0.994	−0,1129
150.10.3	150	8.49602	8.49602	9.02065	9.70515	9.70515	1.300	9.70515	9.84302	1.522	1,20913
150.10.4	150	8.83734	8.83734	9.03983	10.83017	10.83017	1.783	10.83017	10.97248	2.012	1,99283
150.20.1	150	17.31938	17.31938	17.59636	17.01989	17.01989	1.872	17.00989	17.61747	2.323	−0,30949
150.20.2	150	16.63405	16.63405	17.45066	17.87372	17.87372	2.143	16.18372	16.63331	2.612	−0,45033
150.20.3	150	17.40579	17.40579	18.34468	16.98301	16.98301	2.387	16.98301	16.98301	3.025	−0,42278
150.20.4	150	16.87516	16.87516	17.47742	16.81024	16.81024	2.465	16.81024	16.81024	3.339	−0,06492
150.30.1	150	25.98537	25.98537	26.54882	27.85243	27.85243	2.618	25.85243	25.85243	3.489	−0,13294
150.30.2	150	26.20552	26.20552	26.74112	26.13700	26.13700	2.908	26.13700	26.13700	3.569	−0,06852
150.30.3	150	25.31642	25.31642	26.11368	26.01669	26.01669	3.015	25.01006	25.01006	3.707	−0,30636
150.30.4	150	26.10274	26.10274	27.29231	29.89924	29.89924	3.285	25.98124	25.98124	3.863	−0,1215
150.40.1	150	34.01210	34.01210	35.45338	29.46831	29.46831	4.221	29.46831	29.86510	6.477	−4,54379
150.40.2	150	36.56164	36.56164	38.29645	35.23426	35.23426	5.966	35.23426	45.98851	6.997	−1,32738
150.40.3	150	36.65738	36.65738	38.29545	36.87694	36.87694	7.001	36.87694	36.87694	8.353	0,21956
150.40.4	150	35.01556	35.01556	36.06616	35.95548	36.80954	9.524	35.58541	35.75861	10.290	0,56985
200.10.1	200	10.09452	10.09452	10.40499	10.95619	10.95619	12.610	10.09561	10.15742	10.446	0,00109
200.10.2	200	10.42260	10.42260	10.61488	10.28546	10.28546	12.768	10.28546	10.53034	10.597	−0,13714
200.10.3	200	9.79897	9.79897	9.92350	9.81657	9.81657	14.803	9.10497	9.10497	13.983	−0,694
200.10.4	200	10.35528	10.35528	10.63997	10.12512	10.12512	14.948	10.12512	10.12512	15.617	−0,23016
200.20.1	200	21.21505	21.21505	21.46013	21.30734	21.30734	15.097	21.21853	21.21853	15.755	0,00348
200.20.2	200	21.45845	21.45845	22.04610	21.40198	21.78046	15.255	21.019348	21.194771	17.884	−0,43910
200.20.3	200	20.85218	20.85218	21.06040	19.03127	19.03127	18.278	19.031227	19.031227	19.398	−1,82095
200.20.4	200	19.23495	19.23495	20.18035	18.88397	19.05317	18.408	18.88397	19.18397	19.547	−0,35098
200.30.1	200	30.36023	30.36023	31.78264	30.99502	31.72968	18.563	30.07298	30.68295	19.699	−0,28725
200.30.2	200	32.81279	32.81279	33.21640	32.09152	32.09152	19.899	32.09152	32.09152	19.835	−0,72127
200.30.3	200	32.25350	32.25350	32.73727	32.08346	32.08346	21.085	32.08346	32.09672	21.699	−0,17004
200.30.4	200	32.09314	32.09314	32.76376	32.37428	32.37428	22.464	32.03748	32.03748	21.853	−0,05566
200.40.1	200	41.49802	41.49802	42.30479	41.29542	41.93362	22.614	41.29542	41.57956	22.018	−0,2026
200.40.2	200	43.25021	43.25021	44.22107	43.07165	43.66571	24.043	43.07165	43.12561	25.568	−0,17856
200.40.3	200	43.33753	43.33753	44.26132	43.18571	43.18571	25.412	43.18571	43.18571	26.182	−0,15182
200.40.4	200	42.05785	42.05785	43.33703	42.03127	42.03127	27.127	42.03127	42.03127	26.428	−0,02658

Table 6
Results for simulation benchmarks.

Instances	Nb.Cust	CPLEX	BKS	HGA with 1000 iterations			HGA with 1500 iterations		
				objective	Average objective	CPU time	objective	Average objective	CPU time
INS_JA_5_1	5	77,113	77,113	183,404	190,314	1,464	77,113	197,756	3,452
INS_JA_5_2	5	177,797	177,797	211,365	276,650	1,017	177,797	196,656	4,338
INS_JA_10_1	10	14,831	14,831	15,823	15,831	0,030	14,831	27,226	0,003
INS_JA_10_2	10	72,397	72,397	114,545	140,574	0,439	72,397	78,997	0,512
INS_JA_10_3	10	73,399	73,399	107,029	111,698	0,404	73,399	99,123	0,560
INS_JA_15_1	15	206,066	206,066	311,380	338,508	0,425	206,066	231,593	0,631
INS_JA_15_2	15	271,352	271,352	358,484	443,998	1,245	271,352	297,781	3,208
INS_JA_15_3	15	292,411	292,411	304,764	384,544	2,492	292,411	345,744	4,318
INS_JA_15_4	15	1851,17	1851,17	1917,212	2063,551	5,040	1851,17	1982,577	5,776
INS_JA_15_5	15	177,488	177,488	243,530	281,958	4,056	177,488	196,348	5,328
INS_JA_25_1	25	–	64,451	84,557	142,059	1,018	64,451	130,444	2,193
INS_JA_25_2	25	–	99,868	121,944	127,722	8,279	99,868	99,707	9,123
INS_JA_25_3	25	–	163,608	171,659	197,717	5,063	163,608	351,261	7,846
INS_JA_25_4	25	–	236,132	328,012	340,611	10,483	236,132	280,756	15,634
INS_JA_50_2	50	–	75,862	113,811	142,768	13,487	75,862	79,456	21,542
INS_JA_50_3	50	–	146,598	166,754	176,460	20,838	146,598	146,478	21,975
INS_JA_50_4	50	–	162,747	180,793	182,886	22,856	162,747	169,212	22,940
INS_JA_50_5	50	–	207,794	236,714	268,421	25,252	207,794	236,601	25,374
INS_JA_75_2	75	–	212,521	252,138	254,267	19,899	212,521	249,911	23,875
INS_JA_75_3	75	–	324,517	337,947	351,552	26,012	324,517	344,352	30,010
INS_JA_75_4	75	–	317,805	323,516	323,677	24,395	317,805	317,516	28,440
INS_JA_75_5	75	–	344,862	358,291	499,655	27,943	344,862	351,261	31,018
INS_JA_75_6	75	–	479,260	479,260	479,500	30,325	479,260	478,829	30,580
INS_JA_75_7	75	–	365,006	492,590	499,655	36,064	365,006	371,388	27,522
INS_JA_100_6	100	–	280,983	358,459	372,175	32,228	280,983	294,148	37,207
INS_JA_100_7	100	–	302,129	302,129	400,442	32,353	302,129	392,173	39,232
INS_JA_100_8	100	–	333,802	376,093	369,563	30,466	333,802	335,402	34,613
INS_JA_100_9	100	–	472,328	487,661	492,820	35,755	472,328	478,613	38,394
INS_JA_100_10	100	–	466,515	479,945	485,901	38,447	466,515	472,804	40,088
INS_JA_200_7	200	–	602,833	602,833	604,135	39,689	602,833	603,291	41,261
INS_JA_200_8	200	–	633,404	633,404	637,243	40,370	633,404	634,935	44,697
INS_JA_200_9	200	–	693,872	702,583	708,656	41,367	693,872	700,957	43,345
INS_JA_200_10	200	–	701,587	702,583	717,895	44,879	701,587	700,957	49,07
INS_JA_200_12	200	–	756,732	761,154	767,251	41,366	756,732	765,762	45,12
INS_JA_200_15	200	–	891,772	897,587	911,721	62,809	891,772	896,780	65,88

[3] J. Euchi, Do drones have a realistic place in a pandemic fight for delivering medical supplies in healthcare systems problems?, 2020.
[4] S.O. Al-Jazzar, Y. Jaradat, AOA-Based drone localization using wireless sensor-doublets, Phys. Commun. 42 (2020) 101160.
[5] J. Euchi, Hybrid estimation of distribution algorithm for a multiple trips fixed fleet vehicle routing problems with time windows, Int. J. Oper. Res. 21 (4) (2014) 433–450.
[6] R. Moussi, J. Euchi, A. Yassine, N.F. Ndiaye, A hybrid ant colony and simulated annealing algorithm to solve the container stacking problem at seaport terminal, Int. J. Oper. Res. 24 (4) (2015) 399–422.
[7] J. Euchi, The vehicle routing problem with private fleet and multiple common carriers: Solution with hybrid metaheuristic algorithm, Veh. Commun. 9 (2017a) 97–108.
[8] J. Euchi, S. Zidi, L. Laouamer, A hybrid approach to solve the vehicle routing problem with time windows and synchronized visits in-home health care, Arab. J. Sci. Eng. (2020) 1–16.
[9] J. Euchi, Genetic scatter search algorithm to solve the one-commodity pickup and delivery vehicle routing problem, J. Model. Manag. 12 (1) (2017b) 2–18.
[10] Z. Zhou, D. Luo, J. Shao, Y. Xu, Y. You, Immune genetic algorithm based multi-UAV cooperative target search with event-triggered mechanism, Phys. Commun. 41 (2020) 101103.
[11] M.N. Boukoberine, Z. Zhou, M. Benbouzid, A critical review on unmanned aerial vehicles power supply and energy management: Solutions, strategies, and prospects, Appl. Energy 255 (2019) 113823.
[12] C. Murray, A. Chu, The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery, Transport. Res. Part C: Emerg. Technol 54 (2015) 86–109.
[13] P. Bouman, N. Agatz, M. Schmidt, Dynamic programming approaches for the traveling salesman problem with drone, Networks 72 (4) (2018) 528–542.
[14] Z. Luo, Z. Liu, J. Shi, A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle, Sensors, MDPI AG 17 (5) (2017) 1144.
[15] Y.S. Chang, H.J. Lee, Optimal delivery routing with wider drone-delivery areas along a shorter truck-route, Expert Syst. Appl. 104 (2018) 307–317.
[16] . Murray, R. Raj, The Multiple Flying Sidekicks Traveling Salesman Problem: Parcel Delivery with Multiple Drones, Technical Report SSRN 3338436, University of Buffalo, NY, USA, 2019.
[17] X. Wang, S. Poikonen, B. Golden, The vehicle routing problem with drones: several worst-case results, Optim. Lett. 11 (4) (2016) 679–697, Springer Nature.
[18] D. Schermer, M. Moeini, O. Wendet, Algorithms for Solving the Vehicle Routing Problem with Drones, in: Intelligent Information and Database Systems, 35, Springer International Publishing, 2018, pp. 2–361.
[19] D. Schermer, M. Moeini, O. Wendt, Algorithms for Solving the Vehicle Routing Problem with Drones, in: Lecture Notes in Artificial Intelligence, 10751, Springer, 2018, pp. 352–361.
[20] M.W. Ulmer, B.W. Thomas, Same-day delivery with heterogeneous fleets of drones and vehicles, Networks 72 (4) (2018) 475–505.
[21] Z. Wang, J.-B. Sheu, Vehicle routing problem with drones, Transp. Res. B 122 (issue C) (2019) 350–364.
[22] D. Sacramento, D. Pisinger, S. Ropke, An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones, Transp. Res. C 102 (2019) 289–315.
[23] D. Scherme, M. Moeini, O. Wendt, A matheuristic for the vehicle routing problem with drones and its variants, Transp. Res. C 106 (2019) 166–204.
[24] N. Agatz, P. Bouman, M. Schmidt, Optimization approaches for the traveling salesman problem with drone, Transp. Sci. 52 (4) (2018) 965–981.
[25] Q.M. Ha, Y. Deville, Q.D. Pham, M.H. Hà, On the min-cost traveling salesman problem with drone, Transp. Res. C 86 (2018) 597–621.
[26] D. Schermer, M. Moeini, O. Wendt, A hybrid VNS/Tabu search algorithm for solving the vehicle routing problem with drones and en route operations, Comput. & Oper. Res. 109 (2019) 134-58.
[27] Q.M. Ha, Y. Deville, Q.D. Pham, et al., A hybrid genetic algorithm for the traveling salesman problem with drone, J. Heuristics (2019) 1–29.
[28] S. Poikonen, X. Wang, B. Golden, The vehicle routing problem with drones: extended models and connections, Networks 70 (1) (2017) 34–43.
[29] D. Schermer, M. Moeini, O. Wendt, A matheuristic for the vehicle routing problem with drones and its variants, Transp. Res. C 106 (2019) 166–204.
[30] M. Moeini, H. Salewski, A genetic algorithm for solving the truck-drone-ATV routing problem, in: H. Le Thi, H. Le, T. Pham Dinh (Eds.), Optimization of Complex Systems: Theory, Models, Algorithms and Applications. WCGO 2019, in: Advances in Intelligent Systems and Computing, 991, 2020, pp. 1023–1032.
[31] J.J. Dongarra, Performance of Various Computers using Linear Equation Software, Technical report, University of Manchester, Manchester, UK, 2009.

Jalel Euchi is a faculty member at the Department of Management Information Systems, CBE, Qassim University, KSA. He received his Ph.D. in Computer Science, in the field of Optimization and Transportation Problems from the Sfax University in Tunisia and LeHavre University in France, in 2011. He is an Associate Researcher in the OLID Laboratory, Sfax University, Tunisia, also in Laboratoire de Mathématiques Appliquées du Havre (LMAH), LeHavre University, France. He has published extensively in reputed journals like the International Journal of Operational Research, Swarm and Evolutionary Computation, Management Decision, Journal of Operational Research Society, and Vehicular Communications. He is a referee of many journals published by Springer and Elsevier publishers. He is a life member of the Operational Research Society of Tunisia. His primary research interests include complex vehicle routing problems, heuristics, and meta-heuristics algorithms to solve the NP-Hard problems, computational operations research, decision making, and logistics research and supply chains. He is an Executive editor in the Open Transportation Journal.