# The Vehicle Routing Problem with Drones and Flexibility Demands


Tran Thi Hue*†, Bao Nguyen*, Quoc Nguyen*, Pham Phu Manh*,
Nguyen Khanh Phuong*, Huynh Thi Thanh Binh*, Dang Quang Thang*
*Hanoi University of Science and Technology, Hanoi, Vietnam
† Banking Academy of Vietnam
huett@hvnh.edu.vn, nguyenngocbaocmt@gmail.com, quoc.ntn210726@sis.hust.edu.vn, phumanh1998@gmail.com,
phuongnk@soict.hust.edu.vn, binhht@soict.hust.edu.vn, thang.dq232044M@sis.hust.edu.vn



2024 IEEE Congress on Evolutionary Computation (CEC) | 979-8-3503-0836-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/CEC60901.2024.10611902



***Abstract*—In recent years, the Vehicle Routing Problem, with its variants handling complex realistic constraints, continues to be a key area of research, reflecting its enduring importance in modernized societies. Nevertheless, less attention has been paid to the variant with the flexibility of customer demands. This paper introduces the problem of addressing the integration of drones into the existing transportation system to maximize the profit of distributing freight during a limited working time by allowing flexibility in the delivered quantity. We propose a Mixed Integer Linear Programming (MILP) formulation and a genetic algorithm (GA) to solve the problem. To assess the proposed algorithm, we create the benchmark instances of the problem from published data sources. Experimental results on the instances show a significantly better performance of our proposal than MILP. Moreover, we evaluate our approach by comparing its performance with existing solutions in published closely related works.**



***Index Terms*—drone, routing, split delivery, flexibility demand, profit, genetic algorithm**


## I. Introduction

The distribution system allowing customer demands to vary flexibly within a predefined range is raised in the context of vendor-managed inventory (VMI) systems. This is because the vendor in such systems needs to monitor the inventory levels of customers and decide when and how much to serve each customer so that the volume of freight delivered to each customer must not cause the stock-out at the customer nor exceed the storage capacity of its warehouse. A popular example of a VMI system is the supplier-retailer partnership between the Procter & Gamble (P&G) manufacturer and Wal-Mart retailers [1]. To our knowledge, in the academic literature on vehicle routing problems, research works on the idea of allowing customer demands in the range are relatively scarce. However, the experiment results in those studies [2]–[5], demonstrated that flexibility in customer demands could reduce the average distribution and inventory costs. For further investigation, this paper will extend those existing works.

Specifically, we introduce the Vehicle routing problem with drones and flexibility demands (VRP-DFD) which addresses the integration of drones into the existing transportation system to maximize the profit of distributing freight during a limited working time. As far as the authors are aware, the VRP-DFD is the first variant of the VRP in which fleets of trucks and drones collaborate to make deliveries, while flexibility in the delivered quantity and split deliveries are allowed. Further, a wide range of constraints, including heterogeneous fleet with different payload capacities and ways of performing the delivery trip, maximum working and driving time limits of vehicles, is also taken into account in the VRP-DFD, which makes it a hard problem. We provide a MILP model and generate benchmark instances for the VRP-DFD. Due to the complexity to optimally solve the problem, a GA is proposed. The performance of the GA is assessed through extensive computational experiments and comparison with published closely related problem. The results show that our proposed GA outperforms the CPLEX solver and provides better results to the literature. In the following, section II introduces problem description and the related works. The mathematical formulation is then presented in section III. Section IV describes the proposed GA. Computational results are reported and analyzed in section V, while the conclusions of this study are considered in section VI.

## II. Problem Description and Literature review

### A. Problem description

In the VRP-DFD, there is a depot, a limited set of trucks $\mathcal{K}$ and drones $\mathcal{D}$ operating independently to deliver freight to a set of customers $\mathcal{C}$. Each customer $i \in \mathcal{C}$ has a demand in a range $[l_i, u_i]$, and the profit value per each freight unit delivered to them is $w_i$. The profit $w_i$ is estimated based on numbers factors such as inventory cost at the warehouse and expected revenue. In our problem, each customer $i \in \mathcal{C}$ could be serviced more than once by different vehicles, and thus the total delivered quantity to this customer must be greater than or equal to the lower limit $l_i$ and must not exceed the upper limit $u_i$. Each trip in the VRP-DFD is a route run by a vehicle that loads freight at the depot, delivers all freight to one or several customers in $\mathcal{C}$, and then goes back empty at the depot.

The VRP-DFD could be seen as the problem of determining trips for trucks and drones. Each truck only performs one trip, while each drone does either one or a sequence of trips, and the traveling time of each drone trip must not be longer than the maximum flight time $L_d$ due to limited battery endurance. The objective is to maximize the total profit obtained by delivering freight to customers while the following constraints and assumptions are also considered: All trucks have the same capacity $M_t$ and remain in constant speed while on a route; All drones have the same capacity $M_d$, and they are always assumed to be fully recharged when leaving the depot and remain in constant flight while on a sortie; Service time of vehicles at customers is negligible; All vehicles must start their journeys from time 0 and finish work at the latest time $L_w$.

### B. Literature review

Due to the characteristics of the VRP-DFD problem, this section reviews the relevant literature on: (1) parallel truck-and-drone scheduling problem and (2) vehicle routing problem with flexibility demand.

*1) Parallel truck-and-drone scheduling problem:* The related works on the use of drones in routing and scheduling problems could be classified into three categories: (1) using only drones for deliveries, (2) drones and trucks completing their mission in parallel mode, (3) drones and trucks rely on each other to complete their tasks in synchronization. The integration of drones into a truck delivery system in VRP-DFD shares the parallel setting with the second category.

The idea of building an integrated model using drones and ground vehicles in parallel mode to achieve faster deliveries was first introduced by Murray and Chu [6] in 2015. In their work, a single truck and a homogeneous fleet of drones operate independently to minimize the completion time of servicing customers. The drones could perform back-and-forth trips between the depot and customers but serve only one customer per trip. Our proposed VRP-DFD further extends the work of Murray and Chu [6] in several respects. First, the VRP-DFD considers split delivery where each customer could be serviced more than once and by different types of vehicles, while each customer is serviced exactly once in [6]. Second, customer demands in [6] is fixed by a value of one package. In contrast, the VRP-DFD allows the demands of each customer within an interval, thus providing flexibility in the delivered quantity. It eventually requires a solution approach that not only determines routes for vehicles but also assigns freight volumes per delivery to customers. The third respect concerns the fact that the VRP-DFD allows using a limited homogeneous fleet of trucks, and each drone trip could service more than one customer, while only one truck is used in [6], and each drone could only serve one customer per trip.

Table I summarizes the main features of the parallel truck-and-drone scheduling problem in the scientific literature, from which the comparison between the main features of our VRP-DFD problem with those of other problems is displayed. To our knowledge, split delivery, flexible demands, and the objective function for profit maximization are three aspects that have not been considered all together in the literature on vehicle routing problem with drones.

*2) Vehicle routing problem with flexibility demand:* The idea of flexibility in customer demands was first introduced by [2] in the context of ship routing with inventory management constraints. The problem stemmed from the need for an international company that produces and consumes ammonia in their factories around the world. The ammonia produced in a factory is transported by ships to other factories that consume the product so that they always have a sufficient supply of ammonia. However, all of the consuming factories have stock capacity limits. Thus, during the planning period, each factory will need to be supplied with an amount of ammonia within a predefined range. In this context, their studied problem considers a set of pickup and delivery points, which are harbors; each harbor has a predetermined delivery quantity interval; the problem aims to decide which harbors each ship should visit and how much to pick up/drop off at each harbor so that the cost of ship travel is minimized. Their model thus could be considered as a pickup and delivery problem using only ships as transportation means. Our problem, however, uses both trucks and drones in which each type has its own different way to perform the task, and only considers the delivery demands to customers. Moreover, customer demands are allowed to be split in our VRP-DFD. The research works in [3], [15] then extended the problem in [2] by considering the objective function that maximizes the profit gained by operating the fleet and servicing the cargoes minus the cost of using fleet.

Campbell in [4] considered the problem where delivery quantity for each customer $i$ is allowed to vary from its original size $d_i$ by an amount $\alpha d_i$ where $0 \leq \alpha < 1$. By adding this limited flexibility to the problem, their experiment showed that significant savings in the total distance traveled for the vendors and customers could be achieved. Vincent et al. [5] proposed an extension variant of the location-routing problem by allowing the amount of demand delivered to each customer could be in a range. Therefore, among a set of available depots, the problem needs to determine which ones should be opened in order to minimize the sum of the costs associated with the fixed cost of opening the depots, using the vehicles, and the travel cost of the vehicles minus the total extra revenue.

Shetty et al. [16] studied a model using a fleet of drones as combat aerial vehicles to destroy a set of targets. The drone needs to bring an amount of ammunition in the range $[l_i, u_i]$ to each target $i$ where the lower bound $l_i$ corresponds to the minimal ammunition required to destroy the target $i$, while the upper bound $u_i$ corresponds to the maximal ammunition to restrict the collateral damage to the neighboring civilian population. Each target $i$ has an importance value $w_i$ that indicates the criticality of that target. Each drone performs only one trip and brings ammunition to several targets during its trip. The problem aims to maximize the total weighted service to the targets from the drones which is calculated as $\sum w_i m_i$ where $m_i$ is the total amount of ammunition delivered to target $i$. This problem thus could be viewed as a simplified version of our VRP-DFD problem where just one type of vehicles, i.e., drone, is used and each drone performs only one trip.

## III. Mathematical Modelling

Let $\mathcal{C} = \{1, 2, .., c\}$ represents the set of all customers. Although a single physical depot location exists, notationally, we assign it to two unique node numbers, such that vehicles depart from the depot at node 0 and return to the depot at node

TABLE I

SUMMARY OF THE RELATED WORKS ON THE PARALLEL TRUCK- AND-DRONE SCHEDULING PROBLEM

| Reference | Drone capacity | Truck capacity | Working time limit | Multi-visit per drone trip | Multi-trip drone | Split delivery | Demands of customer | Objective function | Solution method |
|---|---|---|---|---|---|---|---|---|---|
| Ulmer & Thomas [7] | no | no | yes | no | yes | no | fixed | #customers visited | heuristic |
| Chen et al. [8] | no | no | yes | no | yes | no | fixed | #customers visited | deep Q-learning |
| Nguyen et al. [9] | yes | yes | yes | no | yes | no | fixed | operational cost | ruin and recreate heuristic |
| Ham [10] | no | no | no | no | yes | no | fixed | makespan | constraint programming |
| Wang et al. [11] | yes | no | no | yes | yes | no | fixed | makespan | three-step heuristic |
| Saleu et al. [12] | no | no | no | no | yes | no | fixed | makespan | branch-and-cut, local search |
| Manh et al. [13] | no | no | yes | yes | yes | no | fixed | makespan | tabu search |
| Nguyen et al. [14] | yes | no | no | no | yes | no | fixed | makespan | ruin and recreate heuristic |
| Our proposal | yes | yes | yes | yes | yes | yes | in range | profit | genetic algorithm |

$c+1$. We use the notation summarized in Table II to formulate the VRP-DFD problem as a MILP model mathematically. The VRP-DFD can be formulated as follows:

$$\text{Maximize} \sum_{i\in\mathcal{C}} w_i m_i \tag{1}$$

subject to

$$x_{ij}^k \in \{0,1\} \quad \forall i \in \mathcal{C}\cup\{0\}, \quad \forall j \in \mathcal{C}\cup\{c+1\}, \quad k \in \mathcal{K} \tag{2}$$

$$y_{ij}^{dr} \in \{0,1\} \quad \forall i \in \mathcal{C}\cup\{0\}, \quad \forall j \in \mathcal{C}\cup\{c+1\}, \quad d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{3}$$

$$T_d^r \geq 0 \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{4}$$

$$\sum_{j\in\mathcal{C}} x_{0j}^k = \sum_{i\in\mathcal{C}} x_{i,c+1}^k \quad \forall k \in \mathcal{K} \tag{5}$$

$$\sum_{i\in\mathcal{C}\cup\{0\}} \sum_{j\in\mathcal{C}\cup\{c+1\}} x_{ij}^k = 0 \ \text{ if } \sum_{j\in\mathcal{C}\cup\{c+1\}} x_{0j}^k = 0 \quad \forall k \in \mathcal{K} \tag{6}$$

$$\sum_{j\in\mathcal{C}} x_{0j}^k \leq 1 \quad \forall k \in \mathcal{K} \tag{7}$$

$$\sum_{j\in\mathcal{C}} y_{0j}^{dr} = \sum_{i\in\mathcal{C}} y_{i,c+1}^{dr} \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{8}$$

$$\sum_{i\in\mathcal{C}\cup\{0\}} \sum_{j\in\mathcal{C}\cup\{c+1\}} y_{ij}^{dr} = 0 \ \text{ if } \sum_{j\in\mathcal{C}} y_{0j}^{dr} = 0 \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{9}$$

$$\sum_{j\in\mathcal{C}} y_{0j}^{dr} \leq 1 \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{10}$$

$$l_0^k = 0 \quad \forall k \in \mathcal{K} \tag{11}$$

$$l_j^k = l_i^k + 1 \ \text{ if } x_{ij}^k = 1 \quad \forall k \in \mathcal{K}, i \in \mathcal{C}\cup\{0\}, j \in \mathcal{C}\cup\{c+1\} \tag{12}$$

$$l_0^{dr} = 0 \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{13}$$

$$l_j^{dr} = l_i^{dr} + 1 \ \text{ if } y_{ij}^{dr} = 1 \ \forall d \in \mathcal{D}, \quad r \in \mathcal{R}, \quad i \in \mathcal{C}\cup\{0\}, j \in \mathcal{C}\cup\{c+1\} \tag{14}$$

$$\sum_{i\in\mathcal{C}\cup\{0\}} x_{ij}^k = \sum_{i\in\mathcal{C}\cup\{c+1\}} x_{ji}^k \quad \forall j \in \mathcal{C}, \quad k \in \mathcal{K} \tag{15}$$

$$\sum_{i\in\mathcal{C}\cup\{0\}} y_{ij}^{dr} = \sum_{i\in\mathcal{C}\cup\{c+1\}} y_{ji}^{dr} \quad \forall j \in \mathcal{C}, \quad d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{16}$$

$$\sum_{j\in\mathcal{C}} y_{0j}^{dr} \geq \sum_{j\in\mathcal{C}} y_{0j}^{d,r+1} \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{17}$$

$$\sum_{i\in\mathcal{C}\cup\{0\}} \sum_{j\in\mathcal{C}\cup\{c+1\}} x_{ij}^k t_{ij} \leq L_w \quad \forall k \in \mathcal{K} \tag{18}$$

$$T_d^r = \sum_{i\in\mathcal{C}\cup\{0\}} \sum_{j\in\mathcal{C}\cup\{c+1\}} y_{ij}^{dr} t'_{ij} \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{19}$$

$$\sum_{r\in\mathcal{R}} T_d^r \leq L_w \quad \forall d \in \mathcal{D} \tag{20}$$

$$T_d^r \leq L_d \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{21}$$

$$m_i^k = 0 \ \text{ if } \sum_{j\in\mathcal{C}\cup\{c+1\}} x_{ij}^k = 0, \quad \forall i \in \mathcal{C} \tag{22}$$

$$m_i^{dr} = 0 \ \text{ if } \sum_{j\in\mathcal{C}\cup\{c+1\}} y_{ij}^{dr} = 0, \quad \forall i \in \mathcal{C} \tag{23}$$

$$m_i = \sum_{k\in\mathcal{K}} m_i^k + \sum_{d\in\mathcal{D}} \sum_{r\in\mathcal{R}} m_i^{dr}, \quad \forall i \in \mathcal{C} \tag{24}$$

$$m_i \geq l_i, \quad \forall i \in \mathcal{C} \tag{25}$$

$$m_i \leq u_i, \quad \forall i \in \mathcal{C} \tag{26}$$

$$\sum_{i\in\mathcal{C}} m_i^{dr} \leq M_d, \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{27}$$

$$\sum_{i\in\mathcal{C}} m_i^k \leq M_t, \quad \forall k \in \mathcal{K} \tag{28}$$

The objective function (1) maximizes the total profit obtained by the system. Constraints (2) - (4) specify the domains and allowable ranges of the variables. Constraints (5) - (10) guarantee that every trip must start and end at the depot. Constraints (11) - (14) ensure that each truck and drone trip has no cycles created by visited customers. Constraints (15) - (16) respectively state that if a truck or a drone arrives at a customer then it must leave this customer. Constraint (17) verifies the correct sequence of assignment of trips to each drone. Constraints (18) - (20) guarantee that the task completion time of each truck and drone must not exceed the working time $L_w$ of the system. Constraint (21) states that drones do not exceed limited flying time in a sortie. Constraints (22) - (26) enforce the total volume of freight delivered to each customer $i$ must be within the range $[l_i, u_i]$. Constraints (27) - (28) ensure the maximum payload that each drone and truck could carry.

TABLE II
NOTATION USED IN MATHEMATICAL MODEL

| | |
|---|---|
| **Parameters** | |
| $t_{ij}, t'_{ij}$ | Travel time from node $i$ to node $j$ by truck, drone |
| **Variables** | |
| $x_{ij}^k$ | Binary variable; 1 if $k^{th}$ truck uses arc $(i,j)$ to move from node $i$ to node $j$; otherwise, 0; |
| $y_{ij}^{dr}$ | Binary variable; 1 if $r^{th}$ trip of $d^{th}$ drone uses arc $(i,j)$ to move from node $i$ to node $j$; otherwise, 0; |
| $T_d^r$ | The task completion time of the $r^{th}$ trip of $d^{th}$ drone |
| $l_i^k$ | The order of node $i$ in the trip of the $k^{th}$ truck |
| $l_i^{dr}$ | The order of node $i$ in the $r^{th}$ trip of the $d^{th}$ drone |
| $m_i^k$ | The volume of freight delivered to customer $i$ by the $k^{th}$ truck |
| $m_i^{dr}$ | The volume of freight delivered to customer $i$ by the $r^{th}$ trip of $k^{th}$ drone |
| $m_i$ | Total volume of freight delivered to customer $i$ |

## IV. THE PROPOSED GENETIC-BASED ALGORITHM

The VRP-DFD is *NP-hard* as it includes the *NP-hard* Vehicle Routing Problem. We thus propose a population-based hybrid meta-heuristic, called **F**low **E**nhanced **G**enetic **A**lgorithm (FEGA) to solve the problem. The general structure of the FEGA is illustrated as follows: The FEGA uses one population only with a fixed size $N$, which may contain both feasible and infeasible individuals. First, the initial population $\mathcal{P}$ is created using a greedy heuristic seeking to fully utilize vehicles and maximize the total profit (section IV-C). At each generation, a new population $\mathcal{P}'$ is generated from the current one $\mathcal{P}$ through the selection, crossover to create offspring, then apply mutation, education, and replacement phases of the algorithm (sections IV-D to IV-F, respectively) to offspring. Before starting a new generation, the Control procedure is called to change the proportion to be used for each crossover and mutation operator based on their performance in the previous generation (section IV-D). The algorithm is stopped when the total number of executed generations reaches $G_{max}$.

### *A. Individual representation*

**Encoding**: An individual for FEGA corresponds to either a feasible or an infeasible solution to the VRP-DFD. The relevant characteristics of each individual are encoded into a chromosome consisting of two arrays of the same size. The *Route* array is the concatenation of all routes, each route is an ordered sequence of customers. The *Volume* array represents the amount of freight delivered to customers. Starting from the left, the $i$th element of *Volume* array encodes the freight quantity delivered to the corresponding customer at $i$th position of the *Route* array. Figure 1 illustrates a chromosome of a solution with 6 customers. In this illustration, customer 3 at the first position is delivered freight with an amount of 500.

**Decoding**: Given a chromosome, a split procedure is used to divide its corresponding *Route* and *Volume* arrays into routes of $|\mathcal{K}|$ trucks and $|\mathcal{D}|$ drones according to constraints of the problem. The procedure generates the order of routes by giving priority firstly to a route for each truck and secondly sequences of routes for $|\mathcal{D}|$ drones. The $i$th sequence decodes the $i$th trips of these $|\mathcal{D}|$ drones rather than trips of the $i$th drone. The goal is to use as many available vehicles as possible so that the limited working time of the system could be satisfied. We define an $i$th element of a chromosome as a combination of the customer at $i$th position in *Route* array and the freight volume at $i$th position in *Volume* array. Starting from the leftmost element of a given chromosome, the split procedure transforms the corresponding solution by considering each element, denoted by (customer $c$, freight volume $v$), through the following process:

- It will cease construction on the current route and initiate the development of a new route if adding the considered element to the current route would violate either the system working time $L_w$ or the flight time limit of drone $L_d$. If these two constraints are satisfied, but appending the considered element would violate the vehicle capacity, the split procedure will split this element into two distinct ones. The former (customer $c$, freight volume $v_1$) will be appended at the end of the current vehicle, where $v_1$ equals to the difference between the limited capacity of the vehicle and its current total capacity. It thus helps to fill up the vehicle. The latter (customer $c$, freight volume $v_2$) is considered to initiate a new route, where $v_2$ is obtained by subtracting $v_1$ from $v$.
- If the customer $c$ within the considered element already appears at a previous position in the current route, the freight volume delivered to that customer in the route is added by a value of $v$. However, if this addition results in a violation of the capacity constraint for the current route, we must split the element as mentioned above.

Normally, the decoding ceases when going through all elements of the given chromosome. This process might also terminate when it cannot assign any route to drones or all customers reach their upper bound of demands. In that case, the redundant part at the end of the chromosome is omitted. Figure 1 illustrates how the split procedure transforms a

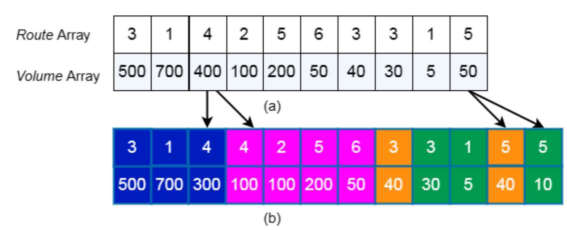


Fig. 1. Illustration of encoding and decoding. **a** Chromosome. **b** Solution.

solution from a given chromosome when two trucks and two drones are available. In Figure 1b, segments with the same color represent trips of one vehicle. Scanning the $Route$ and $Volume$ arrays from the left, it commences to generate the

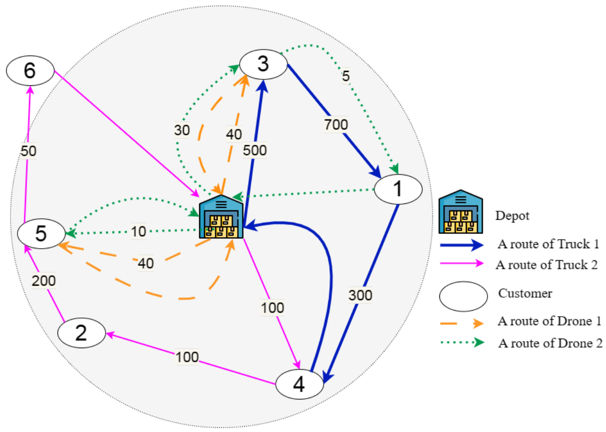


Fig. 2. Solution decoded from Figure 1

route for the first truck by adding the first two elements to the route as all constraints of the problem are satisfied. However if continuing to include the 3rd element to this route, it would exceed the truck capacity as the remaining capacity can only carry a volume of 300. Accordingly, the 3rd element (customer 4, volume 400) is bifurcated into two distinct elements: the element (customer 4, volume 300) is appended to the route of the first truck, while the element (customer 4, volume 100) serves as the inception of a new route. This procedure is repeated until reaching the end of arrays. Figure 2 is the depiction of the solution obtained.

### B. Search space and individual evaluation

Allowing meta-heuristics to consider infeasible solutions often helps the search reaches higher-quality solutions more efficiently, we thus allow infeasible solutions during the FEGA by relaxing constraint on lower limit of customer demands.

Given a solution $s$, let $p(s)$ denote the total profit obtained by performing its routes, and let $d(s)$ denote the total violation of the lower bound of customer demands. The values of $d(s)$ is $\sum_{i \in \mathcal{C}} max\{(m_i - l_i), 0\}$ where $l_i$ and $m_i$ are lower bound of demands at customer $i$ and total volume of freight delivered to customer $i$, respectively. Solutions are then evaluated according to the weighted fitness function $f(s) = p(s) + \beta d(s)$, where $\beta$ is a penalty parameter adjusted dynamically during the search based on the quality of individuals in previous population. We set $\beta = maxP - minP$, where $maxP$ and $minP$ are the maximum and minimum values of the profit among all individuals in the previous population, respectively.

### C. Initial population

To fully utilize all available vehicles of the system, a greedy heuristic is proposed to generate each individual for the initial population. The heuristic consists of two phases. In the first phase, the heuristic first creates $|\mathcal{K}|$ routes, one for each truck; then it iteratively constructs a set of $ith$ routes for $|\mathcal{D}|$ drones where $i = 1, 2, ...$ The goal of the first phase is to create routes that have as many customers meeting their lower bound of demands as possible, given vehicle constraints (i.e., capacity, limited working time, and flight time) are not violated. As a result, in the individual obtained at the end of the first phase, there might still exist some customers whose delivered freight equals either 0 or less than their own lower bound of demands. Consequently, in the second phase, this individual will be sent to the education procedure (detailed in Section IV-E) to further add those customers and then to reassign the freight delivered to customers within the given routes so that the system's total profit is maximized.

### D. Crossover and mutation operators

Whenever offspring are required during the FEGA, two distinct individuals in the population $P$ are selected randomly and passed to the crossover operator. We use one-point and two-point crossover operators, but for each pair of parents, only one operator is selected to create two offspring. Each offspring yielded by the selected crossover operator has a probability $p_m$ of being mutated. We introduce two mutation operators. The former is the segment-swap operator, the latter aims to alter the genes of the chromosome more than the former does. It thus reverses two selected random segments before exchanging them. Similarly, only one mutation operator is selected and applied for each offspring.

At each generation of the FEGA, a Control procedure is applied to calculate the probability of selecting each crossover and mutation operator among all operators. The selection of an operator is based on its performance in the previous generation. We control the probability by first using the crossover-selection parameter $\theta_1$, assigning to the first and second crossover type respectively the probability $\theta_1$ and (1-$\theta_1$) of being selected. Similarly, the mutation-selection parameter $\theta_2$ is used for mutation. Initially, the search allows the selection of each crossover (mutation) operator with equal probability, i.e., values of $\theta_1$ and $\theta_2$ are set to 0.5. At the end of each generation, $\theta_1$ and $\theta_2$ are adjusted by equations: $\theta_1 = (1-\alpha) \times \theta_1 + \alpha \times \frac{A_1}{A}$ (29) where $A_1(A)$ is the number of times the offspring generated by the one-point crossover operator (both two crossover operators) is better than one of its corresponding parents, and $\theta_2 = (1-\alpha) \times \theta_2 + \alpha \times \frac{B_1}{B}$ (30) where $B_1(B)$ is the number of times the individual generated by the first mutation operator (both two mutation operators) is better than one of its parents. The parameter $\alpha$ controls how quickly the probability adjustment $\theta_1$ and $\theta_2$ react to changes in the effectiveness of the operators.

### E. Education

Crossover and mutation operators yield offspring, which might be feasible or infeasible, and are then sent to the education procedure. The goal of education is to restore the feasibility of the individual as much as possible (if needed) and to enhance its quality. Two issues are addressed in the education procedure:

- The first issue concerns that crossover and mutation operators often yield offspring that violates the lower bound of customer demands and a repair phase is required. Therefore, firstly, if there exists any customer whose delivered freight has not yet reached its lower bound of demands, the education will find a place to insert each of them into vehicles, at a position that increases the travel time of the vehicle by the least and satisfies the constraints for the vehicle. Once all customers are satisfied with the lower bound of demands, the insertion process is then repeated so that the upper bound of demands can be reached. These two steps in general help to maximize the utilization of vehicles' working time and to change the order of customers within routes.

- The second issue is that given the fixed sequence of customers within routes of the offspring, the volumes of delivered freight at customers might not be optimal. We thus build and solve the minimum cost flow problem related to the current offspring to reassign those volumes so that the profit for the system could be optimized, i.e., improve the quality of offspring. Given an offspring, the minimum cost flow problem works on the auxiliary graph $G = \{V, A\}$ with vertex set $V = s \cup t \cup R \cup \mathcal{C}$, where $s$ and $t$ are respectively the source and sink, $R$ is the route set of offspring, $\mathcal{C}$ is the set of customers, and the arc set $A = \{(s, r) : r \in R\} \cup \{(r, i) : r \in R, i \text{ is customer served in route } r\} \cup \{(i, t) : i \in \mathcal{C}\}$. A tuple of three elements (lower limit capacity $l_a$, upper limit capacity $u_a$, cost $c_a$) is associated to each arc $a \in A$ as following:
  - $\{(s, r) : r \in R\}$ includes arcs connecting the source $s$ to each route $r \in R$ where $l_a = 0$, $u_a = M_d$ if $r$ corresponds to the route of drone and $u_a = M_t$ if $r$ is the route of truck, aiming to satisfy the limited of vehicle capacity on all routes, $c_a = 0$.
  - $\{(r, i) : r \in R, i \text{ is customer serviced in route } r\}$ includes arcs connecting the route $r$ to each customer $i$ serviced within it where $l_a = 0$, $u_a = +\infty$, $c_a = 0$, restricting each route to deliver freight only to its customers.
  - $\{(i, t) : i \in \mathcal{C}\}$ includes arcs connecting each customer $i \in \mathcal{C}$ to the sink $t$ where $l_a = l_i$, $u_a = u_i$, $c_a = w_i$, ensuring the lower and upper bound of demands for customer $i$ and measuring the impact of each freight unit going through the arc on the objective value.

  The objective of the minimum cost flow problem is minimize $\sum_{i \in C}(-w_i)m_i$ whose absolute value equals to the value of the objective function (1) in the VRP-DFD problem, while the constraints on the lower limits, upper limits of customers' demands, and vehicles' capacity are satisfied (equations (25) - (28), respectively).

### F. Generation replacement

The goal of the replacement phase is to produce a new generation that conserves the best characteristics of the individuals encountered so far and displays various genetic materials. An elitist approach is thus used, keeping $p_k$% best individuals from the prior generation and all offspring generated.

## V. Computational Results

The goal of the numerical experimentation is threefold: (1) to study the impact of a number variants of major operators on the performance of the proposed FEGA to identify the best design; (2) to evaluate the performance of the FEGA through comparisons with CPLEX results and published results for related problem defined in [16]; and (3) to analyze the impact on solution quality of integrating drones.

Our proposed FEGA is coded in C++ and implemented on an Intel Core i9-12900k 3.20 GHz processor. We generated VRP-DFD test instances by adding lower and upper limits for customer demands, profit values per each freight unit of customers to the Vehicle routing problem with drones instances of Sacramento et al. [17]. In total, 112 instances were used where the number of customers ranges from 6 to 200, with the depot located at [0, 0] in all cases. The customers in each instance were generated using a uniform distribution $\mathcal{U}(-d/2, d/2)$ in a grid size of $d \times d$ where $d$ ranges from 5 to 40 km. We fix the capacity of truck and drone to 1500 kg and 40 kg, respectively. The lower limit of each customer demand was generated randomly in the interval of [0, 400], it consequently there might exist some customers not need to be serviced if their lower limits of demands equal to 0. For each customer $i$, its upper limit demands $u_i$ was then set by adding a random value in the range of $[l_i, 3 * l_i]$ to its lower limit demands $l_i$. The instance set and results can be accessed via https://github.com/phuongnkHUST/VRPFlexDemand.

### A. Analysis of design decisions

Designing a genetic algorithm always requires a number of decisions on the structure, operators and parameter values. In this section, we report the experiments studied the impact of three main design decisions on the behavior of the proposed FEGA: the variants of crossover and mutation operators (section V-A1), the education procedure (section V-A2), the calibration of the search parameters (section V-A3).

*1) Variants of crossover and mutation operators:* The first experiment analyzes the impact of the crossover and mutation operators and the effectiveness of adapting parameters $\alpha$ on the solution quality. Two strategies were investigated:

- Strategy 1: the value of $\alpha$ is set to 0, i.e., the values of $\theta_1$ and $\theta_2$ are fixed during the search.
- Strategy 2: the value of $\alpha$ is different from 0, thus the values of $\theta_1$ and $\theta_2$ are varied during the search based on the equations (29) and (30) respectively.

The experiment was run for 200 generations with population size $N$ = 100. We studied multiple versions of Strategy 1 and Strategy 2. Overall, Strategy 2 yields better solutions than Strategy 1, and the scheme with $\alpha = 0.3$ provides the best solutions. This is explained because each operator might be suitable for certain stages of the solution-finding process. Increasing the frequency of an operator at the right time, when it is most effective, can significantly enhance the algorithm. The value of $\alpha = 0.3$ was thus used in all subsequent experiments.

*2) Capacity of education procedure to repair offspring:* In this experiment, we show that education procedure has ability in repairing offspring generated by crossover and mutation operators, and thus enhances the solution quality for FEGA.

Through 200 generations of FEGA, the education procedure could restore the feasibility of offspring at average of 35.08%. Consequently, the FEGA using education yields better solutions than the FEGA without education, with an improvement gap of 22.2%, and requires 39 seconds for running time which is 2.5 times longer than the case without education.

*3) Calibration of search parameters:* The main parameters of the proposed FEGA requiring calibration are the population size $N$, the number of generations $G_{max}$, the percentage of keeping best individuals from previous population $p_k$, and the probability of applying mutation $p_m$. These parameter values were chosen by considering a number of criteria: improvement of the best solution through generations, solution quality, and the computation time for FEGA. The intervals examined and the final parameter settings are shown in Table III.

TABLE III
CALIBRATION OF MAIN FEGA PARAMETERS

| Parameters | Interval explored | Final value |
|---|---|---|
| Population size ($nPop$) | [50, 200] | 100 |
| Maximum number of generations ($G_{max}$) | [100, 300] | 200 |
| Percentage (%) of keeping best individuals ($p_k$) | [5, 30] | 10 |
| Probability of applying mutation ($p_m$) | [0.03, 0.1] | 0.05 |

### B. The performance of CPLEX and the proposed FEGA

We have tested the MILP model proposed in section III by running CPLEX for 5 hours. The obtained results were compared to that provided by FEGA over 10 runs, as indicated in Table IV. The first column indicates the number of customers in each instance set. The next three columns provide, respectively, number of instances for which the solution obtained belongs in one of the three status types (Optimal, Non-optimal, Unknown), the objective value of solution, computation time in seconds for CPLEX. The next three columns report, respectively, the average of best objective values, average computation time and average standard deviation obtained by our proposed FEGA over 10 runs for each instance set. The GAP column indicates the gap to the average best objective value of solutions obtained by FEGA from the objective value of solutions obtained by CPLEX.

In Table IV, we could observe that FEGA exhibits slightly poorer performance (0.01%) than CPLEX on 6-customer instance set, but gets performance better than or equal to CPLEX for larger size instances. Noteworthy, within 5 hours, the CPLEX could only solve to optimality for all instances of 6 customers, but not for larger instances with 10 to 50 customers. It even fails to find feasible solutions for 8 over 16 instances of 50 customers. The FEGA could achieve optimal solutions to all instances of 10, 12, 20, 50 customers for which CPLEX provides optimal solutions. Moreover, FEGA yields better solutions, with an average improvement gap of 2.8% and 7.92% compared to CPLEX result on 20 and 50-customer instances, respectively. In this context, the population based meta-heuristic like FEGA illustrated clearly its superior performance compared to the CPLEX in terms of both the solution quality and running time when instances of more than or equal to 10 customers are contemplated.

TABLE IV
PERFORMANCE COMPARISON BETWEEN CPLEX AND FEGA

| Instance set | CPLEX | | | FEGA | | | |
|---|---|---|---|---|---|---|---|
| | Optimal/ Feasible/ Unknown | Obj. Value | Time (s) | Obj. Best | Time (s) | Std | GAP (%) |
| 6-customer | 12/0/0 | 21,360.42 | 0.81 | 21,358.33 | 4.61 | 7.03 | -0.01 |
| 10-customer | 11/1/0 | 24,118.75 | 3,137.52 | 24,118.75 | 5.52 | 9.68 | 0.00 |
| 12-customer | 11/1/0 | 24,952.08 | 3,342.93 | 24,952.08 | 6.60 | 5.34 | 0.00 |
| 20-customer | 5/7/0 | 26,762.50 | 10,534.66 | 27,512.50 | 6.90 | 16.42 | 2.80 |
| 50-customer | 1/5/8 | 51,337.50 | 16,988.58 | 55,404.17 | 20.00 | 91.77 | 7.92 |
| 100-customer | - | - | - | 113,650.00 | 44.03 | 45.49 | - |
| 150-customer | - | - | - | 177,464.06 | 91.86 | 72.69 | - |
| 200-customer | - | - | - | 231,615.62 | 120.33 | 75.92 | - |

### C. Comparing with results in the literature

As mentioned in section II, the drone routing problem based on target priorities introduced by Shetty et al. [16] resembles our problem setting most closely among existing works in the literature. This section thus compares the performance of our FEGA with results of their tabu search (TS) on that problem. The test instances include number of targets (i.e., customers) ranging from 10 to 50, which are serviced by using five drone's types with different payload capacity and flight time range attributes of $\{(500, 20), (400, 30), (300, 40), (200, 50), (100, 60)\}$. Thus, the drone with high payload capacity can travel less and vice versa. We use the parameter settings for FEGA as indicated in the 'Final value' column of Table III.

Table V sums up the comparison, based on an average over 10 runs of TS and FEGA. The second to sixth columns provide TS results in two versions of TS implementation, with the original setting in [16] and with computation time extended to 120 seconds. The Feasible column shows the number of instances that TS could find feasible solutions over the total number of instances. The Obj. Best columns report the mean of best objective values over 10 runs for each algorithm only on instances where it can find feasible solutions. One could see that the average computation time of TS implemented in the original setting is only about 12.25 seconds, i.e., nearly 3.5 times smaller than that of FEGA, but there exist instances of all sizes that TS fails to find feasible solutions. We thus extend the running time of TS to 120 seconds. However, the TS can only find a feasible solution for one more instance of 50 customers, while FEGA could provide feasible solutions for all 50 customer instances. The GAP_feasible column shows the gaps for the average best values obtained by FEGA with respect to that of TS on instances in both algorithms that could find feasible solutions. Overall, FEGA outperforms TS, with an average improvement gap of 20.34% on instances in which TS can find feasible solutions.

### D. The benefit of integrating drones

To evaluate the value of integration drones with conventional ground vehicles, i.e., trucks, we evaluated two types of instances: 1) Large upper limit of customer demands, all customers having upper limit of demands greater than or equal to the payload capacity of drone; 2) Small upper limit of customer demands, in which at least 40% customers having upper limit of demands smaller than one-fourth of drone's capacity. Three scenarios are studied:

- $\mathcal{K}$, where the system uses $|\mathcal{K}|$ trucks, 0 drone.
- $\mathcal{K}+1$, where the system uses $|\mathcal{K}|+1$ trucks, 0 drone.

TABLE V

PERFORMANCE COMPARISON BETWEEN TS IN [16] AND FEGA

| Instance set | TS - Original setting version | | | TS -Time extended version | | FEGA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Feasible | Obj. Best | Time (s) | Feasible | Obj. Best | Feasible | Obj. Best | Std | Time (s) | GAP_feasible (%) |
| 10-customer | 8/12 | 5,087.50 | 0.41 | 8/12 | 5,087.50 | 12/12 | 6,293.75 | 3.75 | 29.00 | 18.06 |
| 12-customer | 2/12 | 7,500.00 | 1.11 | 2/12 | 7,500.00 | 10/12 | 6,216.67 | 21.03 | 30.29 | 4.13 |
| 20-customer | 1/12 | 8,300.00 | 3.02 | 1/12 | 8,300.00 | 12/12 | 10,656.25 | 10.57 | 38.36 | 3.01 |
| 50-customer | 0/12 | - | 44.45 | 1/12 | 13,575.00 | 12/12 | 29,292.19 | 13.62 | 74.37 | 56.17 |

- $\mathcal{K}$ + drone, where $|\mathcal{K}|$ trucks and one drone are used.

TABLE VI

COMPARISON OF WITH AND WITHOUT DRONE

| Instance set | Large upper limit of demands | | | Small upper limit of demands | | |
|---|---|---|---|---|---|---|
| | $\mathcal{K}$ | $\mathcal{K}+1$ | $\mathcal{K}+$ drone | $\mathcal{K}$ | $\mathcal{K}+1$ | $\mathcal{K}+$ drone |
| | Obj. Best | GAP (%) | GAP (%) | Obj. Best | GAP (%) | GAP (%) |
| 10-customer | 910.00 | 22.25 | 13.55 | 285.00 | 30.52 | 44.56 |
| 20-customer | 2,630.00 | 30.92 | 16.49 | 366.00 | 19.27 | 28.63 |
| 50-customer | 3,370.20 | 44.28 | 34.73 | 1,094.00 | 6.97 | 25.04 |
| Average | 2,303.33 | 32.48 | 21.59 | 585.67 | 18.92 | 32.74 |

Table VI compares the average best results over 10 runs of these scenarios on each instance type. The Obj. Best column reports the average best objective value of the $\mathcal{K}$ scenario. The GAP(%) columns display the gaps for the average best objective values obtained by $\mathcal{K}$+1 and $\mathcal{K}$+drone scenarios relative to that of $\mathcal{K}$ scenario. Not surprisingly, the information in Table VI indicates that the larger the upper limit of demands, the less the gain in performance for integrating drones into the system. Drone integration could demonstrate its advantage compared to trucks only when there exists more customers with customer demands less than the payload capacity of drone in the system. In this case, with drone integration, the $\mathcal{K}$+drone scenario gives the best solutions, with an average increase in the system's profit of 32.74% and 13.82% relative to that of $\mathcal{K}$ and $\mathcal{K}$+1 scenarios, respectively.

## VI. CONCLUSIONS

We introduce the VRP-DFD problem of addressing the integration of drones into the existing transportation system to maximize the profit of distributing freight during a limited working time by allowing flexibility in the delivered quantity. A MILP and a hybrid genetic algorithm FEGA are proposed to solve the problem. Computational experiments demonstrate the validity of the developed MILP and the performance of the FEGA on the first benchmark instances of the problem. Future work includes considering multiple depots, heterogeneous fleets of trucks and drones. Moreover, the case that drones have constant power consumption could be extended by investigating more realistic models in which the power consumption depends on the vehicle speed and payload.

## ACKNOWLEDGMENT

This work is funded by the Ministry of Education and Training of Vietnam for Nguyen Khanh Phuong under project code B2024-BKA-19. It is also sponsored by the Office of Naval Research (ONR) Global (ONRG) and the U.S. Army Combat Capabilities Development Command Indo-Pacific (DEVCOM Indo-Pacific), under grant number N62909-23-1-2018 for Huynh Thi Thanh Binh, Tran Thi Hue, Nguyen Tran Nhat Quoc. In addition, Dang Quang Thang is funded by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2022.ThS.BK.10.

## REFERENCES

[1] M. A. Mahmood, L. A. Gemoets, and A. O. Solis, “Supply chain management,” in *Encyclopedia of Information Systems*, H. Bidgoli, Ed. New York: Elsevier, 2003, pp. 315–327.

[2] M. Christiansen and B. Nygreen, “A method for solving ship routing problems with inventory constraints,” *Annals of Operations Research*, vol. 81, pp. 357—378, 1998.

[3] J. E. Korsvik and K. Fagerholt, “A tabu search heuristic for ship routing and scheduling with flexible cargo quantities,” *Journal of Heuristics*, vol. 16, pp. 117–137, 2010.

[4] A. M. Campbell, “The vehicle routing problem with demand range,” *Annals of Operations Research*, vol. 144, no. 1, 2006.

[5] F. Y. Vincent, P. Jodiawan, Y.-H. Ho, and S.-W. Lin, “Location-routing problem with demand range,” *IEEE Access*, vol. 7, pp. 142–149, 2019.

[6] C. C. Murray and A. G. Chu, “The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery,” *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.

[7] M. Ulmer and B. Thomas, “Same-day delivery with heterogeneous fleets of drones and vehicles,” *Networks*, vol. 72, no. 4, pp. 475–505, 2018.

[8] X. Chen, M. W. Ulmer, and B. W. Thomas, “Deep q-learning for same-day delivery with vehicles and drones,” *European Journal of Operational Research*, vol. 298, no. 3, pp. 939–952, 2022.

[9] M. A. Nguyen, G. T.-H. Dang, M. H. Ha, and M.-T. Pham, “The min-cost parallel drone scheduling vehicle routing problem,” *European Journal of Operational Research*, vol. 299, no. 3, pp. 910–930, 2022.

[10] A. M. Ham, “Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming,” *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 1–14, 2018.

[11] D. Wang, P. Hu, J. Du, P. Zhou, T. Deng, and M. Hu, “Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones,” *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 483–10 495, 2019.

[12] R. G. M. Saleu, L. Deroussi, D. Feillet, N. Grangeon, and A. Quilliot, “The parallel drone scheduling problem with multiple drones and vehicles,” *European Journal of Operational Research*, vol. 300, no. 2, pp. 571–589, 2022.

[13] P. P. Manh, T. T. Hue, H. T. T. Binh, and N. K. Phuong, “The min-timespan parallel technician-and-drone scheduling in door-to-door sampling service system,” in *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10.

[14] M. A. Nguyen and M. H. Ha, “The parallel drone scheduling traveling salesman problem with collective drones,” *Transportation Science*, 2023.

[15] G. Brønmo, M. Christiansen, and B. Nygreen, “Ship routing and scheduling with flexible cargo sizes,” *Journal of the Operational Research Society*, vol. 58, pp. 1167–1177, 2007.

[16] V. K. Shetty, M. Sudit, and R. Nagi, “Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles,” *Computers & Operations Research*, vol. 35, no. 6, pp. 1813–1828, 2008.

[17] D. Sacramento, D. Pisinger, and S. Ropke, “An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones,” *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 289–315, 2019.