Journal of Mathematical Modeling
Vol. -, No. -, 2026, pp. -. Research Article

Mathematical model and hybrid meta-heuristic solution approaches for hub location problem with hybrid drone-airplane delivery mode

Mehrnaz Mohebbi[†], **Hamid Reza Maleki**[†*], **Sadegh Niroomand**[§]

[†]*Department of Mathematics, Shiraz University of Technology, Shiraz, Iran*
[§]*Department of Industrial Engineering, Firouzabad Higher Education Center, Shiraz University of Technology, Shiraz, Iran*
Email(s): m.mohebbi@sutech.ac.ir, maleki@sutech.ac.ir, niroomand@sutech.ac.ir

Abstract. This study addresses the integrated hub location and drone delivery problem, an area with few prior investigations. We propose a bi-objective integer linear programming model to minimize total cost and total drone route time. A novel three-zone structure allows drone transfers between zones via cargo planes, enhancing realism and complexity. Drone capacities are categorized as light and heavy, improving allocation flexibility. Due to the model's complexity, several metaheuristic algorithms including genetic algorithm, differential evolution, simulated annealing, and their hybrid versions are developed and compared. Parameter tuning is performed using the Taguchi method. Computational experiments on various instances show that hybrid algorithms outperform classical methods, providing a practical and integrated framework for hub location and drone delivery planning.

Keywords: Hub location problem, drone delivery problem, meta-heuristic algorithm, hybrid meta-heuristic algorithm, Taguchi method.
AMS Subject Classification 2010: 90C05, 90C11, 90B80.

1 Introduction

A transport network operation becomes very expensive if all destination locations are reached by direct shipments transportation from the source location. So, it is a cost effective method to establish several hubs acting as transshipment node to handle the passing flow [45]. Many people gain considerable advantage from hubs in some networks, such as trading networks, social networks, transportation networks, etc. [50]. In reality, we may get advantages from hubs by using proper paths between them as a hub location problem. The primary purpose of the hub location problem is to determine location of the hubs

*Corresponding author
Received: 19 September 2025/ Revised: 17 November 2025/ Accepted: 03 December 2025
DOI: 10.22124/jmm.2025.31720.2857

and allocation of the non-hub nodes to the hubs in order to minimize network cost [35]. In general, Hub Location Problems (HLPs) are NP-hard problems, including network design decisions and allocation decisions [31]. Single allocation and multiple allocation policies are two basic classes of hub networks. In single allocation policy, all the receiving and sending traffic of each customer is flowed through a single hub. In multiple allocation policy, all the incoming and outgoing traffic of each customer is routed through more than one consecutive hubs [2]. As important applications, HLPs play substantial roles in telecommunication and transportation systems .

In recent years, drones have become more applicable in transportation operations in society and industry. Particular use of uncrewed aerial vehicles (UAVs) have been started in military application since 1916. As the technology continued to show signs of improvement in drones, these devices can now be applied for monitoring purposes, emergency response, delivery of medical tools, entertainment, weather forecasts, etc. [5]. Use of drones has many advantages compared to classical transportation devices. The drones can deliver the objects much faster and with lower operational costs. A drone works without a human pilot and may need less fuel [42, 58]. As the use of online shopping platforms increases sharply, customers become very demanding in terms of time of delivery. Because of proper time of delivery for drones, the use of drones in online shopping platforms may grow considerably in future years [34].

The drone delivery problem (DDP) is a variant of the vehicle routing problem (VRP) that basically consists of a fleet of drones only [34]. The DDP is generally formulated as NP-hard problem [40]. Over the past years, a growing number of companies such as Amazon, Google, UPS, and DHL have become interested in home delivery by drones. Use of drones for parcel delivery has been focused on and evaluated by many firms as a complementary approach to the traditional delivery methods [4].

Literature of the HLPs is full of interesting and application-based studies. As pioneers of this topic, Minas and Mitten [37] assumed one central terminal as the “hub” of a system with outlying terminals. These terminals were far from each other and direct transportations between them were forbidden. So, the flow between outlying terminals (spokes) was forced to be transported via some hubs. O’Kelly [39] proposed the first mathematical formulation for the single allocation p-hub median problem for studying airline passenger networks. The study of Campbell [8] was the first to present a mixed integer linear programming formulation for the single allocation p-hub median problem. Skorin-Kapov et al. [49] presented a new linear model for the single allocation p-hub median problem. They provided the first optimal solution for the single allocation p-hub median problem. A different mixed integer linear programming formulation with fewer constraints and variables was proposed by Ernst and Krishnamoorthy [15]. In another study, a novel formulation for the multiple allocation p-hub median problem was presented by Ernst and Krishnamoorthy [16]. For a better understanding of the subject, the studies of Contreras and O’Kelly [13] and Alumur et al. [3] can be referred. In addition, Kara [27] claimed that the single-allocation p-hub median problem (p-SHMP) is NP-hard when $p \geqslant 3$. In connection with this topic, it emerged that more realistic models with better approximation of real practice have evolved. Rodríguez-Martín et al. [44] presented a model of the routing and hub location problem. The HLPs are classified as NP-hard problem [36, 41]. They may take long time to solve small size instances by using general-purpose mixed integer programming (MIP) solvers [19].

Murray and Chu [38] proposed models for the routing and scheduling of drones for parcel delivery. They assumed that a drone collaborates with a delivery truck to distribute parcels. Minimization of the completion time was their objective. Ha et al. [21] minimized the operational costs for parcel delivery. They considered the costs of transportation and the delay that a vehicle has to wait for the others. Tavana et al. [54] focused on the truck scheduling problem, including inbound trucks and drones. Wang et al. [57] presented the vehicle routing problem with drones (VRP-D), including a group of trucks provided with drones for carrying packages to customers. Poikonen et al. [43] discussed VRP with dispatching drones from the trucks. They improved the analysis for the worst-case outcomes obtained by Wang et al. [57]. Wu et al. [59] studied a cooperative truck-drone problem, in which either a drone or a truck should provide each customer once with a service of parcel delivery. Salama and Srinivas [46] investigated a mixed truck-drone system, in which it was possible to pause the truck at a non-customer location and launch the drone.

Dorling et al. [14] dealt with a typical DDP in which drones may serve several customers per route. They presented a mixed integer linear programming (MILP) formulation and developed a simulated annealing heuristic. Yadav and Narasimhamurthy [60] developed a heuristic for the model that drones may serve one or several customers according to the capacity constraints. Coelho et al. [12] presented a multi-objective DDP. To compensate for the limited driving range, charging stations were introduced. Troudi et al. [55] proposed an approximation model including drones that may perform multiple missions per day. Minimizing the total number of drones was their objective. Liu [32] presented a dynamic drone delivery model to optimize an on-demand meal delivery process. Farajzadeh et al. [17] formulated the routing of multiple drones, considering their paths from different depots. Lately, for increasing the range of drone parcel delivery, application of public transportation networks has been considered [24, 25].

A continuous approximation approach was utilized by Chowdhury et al. [11] to develop a humanitarian logistics supply chain using both drones and truck deliveries. Golabi et al. [20] considered a relief distribution center location problem. In their model, drones could serve the demand points that were difficult to reach. Hub-covering facility location models for limited-range UAV deliveries were proposed by Chauhan et al. [9].

There are many real-world uses for the HLPs in drone-based delivery systems. Medical supplies like blood or vaccinations are delivered by drone to inaccessible or distant locations, hubs are positioned in key areas. Drones are used by online shops to transport merchandise, hubs are positioned close to urban areas with strong demand. Drones provide food and medical supplies to impacted areas as part of disaster relief efforts, near the impacted areas, temporary hubs are erected. Drones are used in agriculture to deliver supplies to farms or spray crops, near agricultural areas, hubs are positioned.

The hub location problem with drone delivery has attracted significant attention in recent years within the field of intelligent transportation and logistics. Various studies have addressed optimization methods for this problem using metaheuristic algorithms. In the area of hub location, several studies have focused on mathematical modeling and solving the problem using metaheuristics. For example, Khalilzadeh et al. [29] and Chobar et al. [10] investigated HLPs under uncertainty with robust mathematical models and metaheuristic approaches. Khaleghi and Eydi [28] employed hybrid solution methods to address a continuous-time multi-period HLP, considering time-dependent demand and sustainability aspects. Their work underscores the advantages of combining metaheuristic approaches to improve solution quality in dynamic and multi-objective logistics networks. Shadkam et al. [48] provided a comprehensive comparative study of popular metaheuristic algorithms, emphasizing the importance of proper parameter tuning for Genetic Algorithm (GA), Simulated Annealing (SA), and Differential Evolution (DE) to enhance performance in complex optimization problems such as hub location and drone delivery. In a subsequent study, Shadkam [47] introduced a hybrid method combining Data Envelopment Analysis (DEA) and Response Surface Methodology (RSM) for practical parameter setting in metaheuristic algorithms, which can notably improve convergence speed and solution accuracy.

Contrary to the most previous studies that have used uniform location models, this study considers a zoned structure for the allocation of hubs in different geographic areas. Utilizing a three-zone structure with distinct roles and integrating the transportation of drones by cargo planes between zones, add complexity and realism to the model. In this study, product delivery is performed by drones, but drone transfers from the third zone to the first and second zones are handled by cargo aircraft. This integration of two vehicle types with different characteristics and roles increases model complexity and simulates more realistic conditions. While many studies assume uniform capacities, here drone payloads are categorized into light and heavy. This enhances modeling accuracy and provides greater flexibility in resource allocation. While numerous studies have investigated the performance of hybrid algorithms like SA-GA or SA-DE, few have implemented both hybrids separately within the same study and directly compared their results. This scarcity is mainly due to implementation complexities and structural differences between the algorithms. Therefore, the present study leverages both SA-GA and SA-DE hybrid algorithms and compares their outcomes to propose the most effective optimization approach for the hub location and drone delivery problem, thereby filling a gap in the existing literature. The objective function is designed to minimize drone flight time while also reducing costs. This multi-objective approach makes the problem more realistic and practical, improving decision quality.

To the best of our knowledge, there is a few important studies on the design of network combining hub location and drone delivery problems. In this study, we consider this issue. Therefore in this study, in order to address an integration of the hub location and the drone delivery problems, we present an integer linear programming formulation. The considered problem is novel and has many contributions compared to those of the literature (see [9, 33]). These contributions can be summarized as follows:

- The study area of the problem is divided into three zones with some depots in each zone, where there are enough drones in the depots.

- Two types of drones with different capacities are considered. For jobs requiring accessibility, accuracy, and speed, drones are an effective tool. Drones do not need infrastructure like roads and bridges, and avoid traffic, obstructions on the ground, and the dangers of traffic accidents. Additionally, they have easy access to hard-to-reach or isolated locations. Drones do not release any pollutants and are suitable for the environment.

- The best location for each hub and assignments of demand nodes and depot nodes to these hubs are determined in the formulation.

- Two objectives such as overall cost and total drone route traversing time are minimized simultaneously, therefore, the problem is formulated as a bi-objective model.

As the proposed formulation has a high degree of complexity, the large-sized instances of the problem cannot be resolved with traditional solution approaches. In order to solve the problem, we introduce a complete encoding-decoding scheme, and apply some meta-heuristic solution approaches such as GA, DE, and SA. Further than these classical approaches, some hybrid versions of these algorithms are developed in this study. All parameters of the proposed algorithms are tuned by the Taguchi method in order to improve their performance. Some test problems with various sizes are considered to evaluate the proposed algorithms numerically. According to the obtained results, the proposed hybrid algorithms perform better than the classical versions of the algorithms.

The rest of this paper is organized in some sections. In Section 2, the proposed problem is formulated, and its solution complexity is discussed. The classical and hybrid meta-heuristic solution approaches are described in Section 3. A complete computational study including test problems, parameter tuning, and final experiments and results is presented by Section 4. Finally, some concluding remarks are presented by the last section of paper.

2 Problem description and formulations

2.1 Problem description and assumptions

As mentioned earlier, this study considers an integration of the hub location and the drone delivery problems. The area of the problem is divided into three zones. These zones include some depots, some hub points, and some customers. Transportation of goods demanded by the customers occurs within and between the zones. The goods are transferred within the zones using drones and between the zones by means of cargo planes that include the loaded drones. In each zone, there are several potential places to establish only one hub. In addition, a limited pre-determined number of depots exist in each zone. We assume that the customers only exist in the first and second zones. However, all of the zones include depots and hubs. If a customer requests a product that is available in the depots of his/her zone, this demand should be satisfied from the depots of that zone. Otherwise, this demand is answered from the depots of the third zone (this zone includes some depots and a hub). It is assumed that there are enough drones in the depots. In general, we assume that there are two types of drones with different capacities and operational costs.

Particularly in contemporary supply chain systems, the combination of cargo aircraft and drones provides a potent logistics option [7]. Lowers labor and fuel costs while lowering reliance on ground transportation and its carbon footprint, speeds up delivery and makes difficult-to-reach places accessible, readily adjusts to logistical networks increasing demand are the main benefits of combining drones and cargo aircraft. Applications of this combination in the real world include:
Online retailers such as Amazon, military logistics, postal services, Relief operations in impacted areas, and so on.

We have employed this combination in our work because of the benefits of employing drones in conjunction with cargo planes.

The customers can be divided into three categories that are explained below:

- The customers who receive their demands from the depots of the same zone. For such customer a drone is loaded in a depot of the zone and moves to the customer and come back to the depot. The short distance between the depot and the customer or flight constraints on the depot-to-hub and hub-to-customer routes are the reasons for direct deliveries from the depot to the customer. Therefore,no-fly restrictions have been explicitly incorporated.

- The customers who receive their demands from the hub of the same zone. In this case, a drone departs from a depot and, after reaching the hub of the same zone, it goes to the customer in the same zone and provides its service. Then, it returns to the hub and goes to the depot. This issue may happen because of some reasons such as long distance between a customer and a depot, lack of aerial route between a customer and a depot, etc.

- The customers who receive their demand from the third zone. This case happens because of the limited capacities of the depots of the first and second zones. In this case, two hubs are used. A drone is loaded in a depot from the origin zone and travels until the hub of that zone. A cargo plane moves the drone (and some other drones) to the hub of the destination zone. Then the drone travels to the customer point. An inverse route is similarly passed, and the drone goes back to the origin depot.

The detailed assumptions below are considered in order to formulate the above-described problem:

- Each drone may visit only one customer.
- Each drone receives its lack of electric charge from the hub located in its zone. The drone must stop at the hub and be charged in order to finish the delivery route because specific consumers are distant from the depot, and it cannot get there without recharging.
- Each customer is served by only one drone.
- All demands must be satisfied.
- Each non-hub node (depot or customer) must be allocated to only maximum one hub point.
- The first zone (Zone 1) includes some customers, some depots, and a hub.
- The second zone (Zone 2) includes some customers, some depots, and a hub.
- The third zone (Zone 3) includes some depots and a hub.
- If the demand of a customer is not available in the depots of the customer's zone, the demand will be satisfied from the third zone. A cargo plane must be used to serve customers who get services from depots in the third zone, and the hubs provide the infrastructure required to land cargo planes.
- Some customers demand a product with less than or equal to α kg weight and other customers request a package with greater than α kg weight.
- Two types of drones are available. Drone type 1 with carrying capacity of less than or equal to α kg and drone type 2 with carrying capacity of more than α kg are considered for serving the customers.
- There are enough number of type 1 and 2 drones available in each depot.
- The navigation time and cost between each pair of nodes using type 1 and 2 drones are known.

  The travel time matrices are carefully constructed based on the actual characteristics of the system. It is important to note that these matrices are defined based on the actual flight times of drones with two different payload capacities (light and heavy). Furthermore, using travel time instead of geometric distances is a fully realistic modeling choice, motivated by the practical nature of the problem and specific flight constraints. This significantly enhances the generalizability and applicability of the results. The purpose of omitting geometric coordinates was to keep the focus on temporal optimization and algorithmic performance, since travel time, rather than spatial distance, is the key operational factor in drone-based delivery systems.

- The customer and depot nodes of each zone are known.
- The demand of each customer, its weight value, and also the depots among all zones that contain this demand are known. Therefore, the set of customers that can be satisfied from their zone, and the set of customers that should be satisfied from Zone 3, are known.
- It is believed that items not found in the depots in Zones 1 and 2 are kept in the depots in Zone 3. Consequently, the depots in Zone 3 satisfy the needs of customers who desire a product that is unavailable in their locality.
- We know the maximum number of customers that each depot can serve in each zone if it is open.
- The drones are moved between two zones by a cargo plane.
- Each route starts from a depot and ends at the same depot. Therefore, each drone must start from and return to only one depot.

This structure aligns with emergency drone-based medical delivery applications, such as delivering blood units or critical medicines to remote or underserved areas. In such operations, missions are commonly single-package and direct, and full drone availability at hubs is a practical assumption due to preparedness protocols. This simplification enables a more controlled and fair comparison among different metaheuristics, as it reduces the number of confounding variables that affect performance. In this study, the demand of each customer, along with the designated supply zone, is considered as part of the problem assumptions. This assumption was made to reduce the complexity of the model and to focus on optimizing hub locations and drone allocation. Thus, the optimization model is responsible for hub location, depot allocation, and routing decisions. In this study, the physical details of drone performance, such as flight range, energy consumption, and battery capacity, were deliberately not incorporated into the model. Only the charging operation was simply considered as handling at the hubs. Including these physical details could have increased the complexity of the model and reduced the focus of the research.

Figure 1, depicts an example of the network we wish to create and shows a network with 12 customers, six depots and three hubs. Table 1 shows all the movement paths of Figure 1 from the depot to the customer.

The main sets, parameters, and decision variables used in the mathematical model of this study are listed in Tables 2 and 3.

2.2 Problem formulation

Based on the above-mentioned assumptions and the notations introduced in Tables 2 and 3, the following bi-objective MILP model is introduced for the problem described at the beginning of this section:

Objective function 1:

$$\begin{aligned} OF_1 = \min \sum_{i\in I}\sum_{r\in R}\sum_{j\in J_1} (T_{ir}+T_{rj}+T_{jr}+T_{ri})x^1_{irj} \\ + \sum_{i\in I}\sum_{r\in R}\sum_{j\in J_1} (T'_{ir}+T'_{rj}+T'_{jr}+T'_{ri})x^2_{irj} + \sum_{i\in I}\sum_{j\in J_2} (T_{ij}+T_{ji})x'^1_{ij} + \sum_{i\in I}\sum_{j\in J_2} (T'_{ij}+T'_{ji})x'^2_{ij} \end{aligned} \tag{1}$$

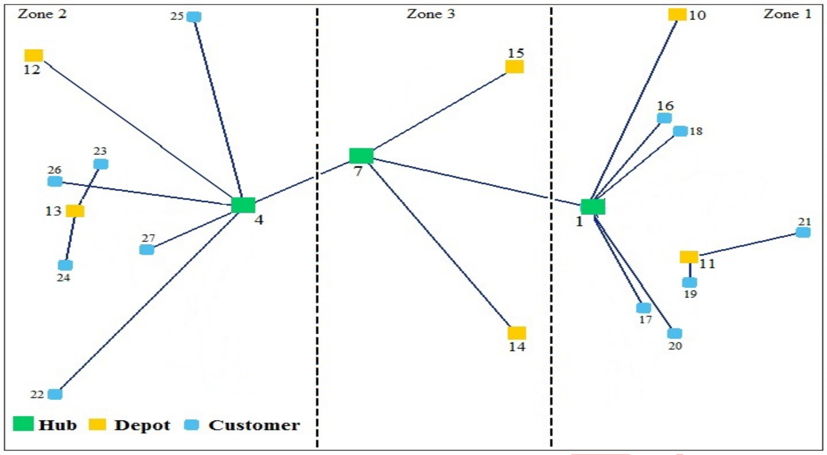

Figure 1: Example of the network

Table 1: The movement paths of Figure 1 from the depot to the customer

Receiving service using a type 1 drone	Receiving service using a type 2 drone
$25 \rightleftharpoons 4 \rightleftharpoons 12$	
$22 \rightleftharpoons 4 \rightleftharpoons 12$	
	$20 \rightleftharpoons 1 \rightleftharpoons 10$
	$16 \rightleftharpoons 1 \rightleftharpoons 10$
$19 \rightleftharpoons 11$	
$23 \rightleftharpoons 13$	
	$24 \rightleftharpoons 13$
	$21 \rightleftharpoons 11$
$17 \rightleftharpoons 1 \rightleftharpoons 7 \rightleftharpoons 14$	
	$18 \rightleftharpoons 1 \rightleftharpoons 7 \rightleftharpoons 15$
	$27 \rightleftharpoons 4 \rightleftharpoons 7 \rightleftharpoons 14$
$26 \rightleftharpoons 4 \rightleftharpoons 7 \rightleftharpoons 15$	

$$+ \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_1}\sum_{j\in J_3}(T_{ck}+T_{rj}+T_{jr}+T_{kc})y^1_{ckrj}+\sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_2}\sum_{j\in J_3}(T_{ck}+T_{rj}+T_{jr}+T_{kc})y^1_{ckrj}$$
$$+ \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_1}\sum_{j\in J_3}(T'_{ck}+T'_{rj}+T'_{jr}+T'_{kc})y^2_{ckrj}+\sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_2}\sum_{j\in J_3}(T'_{ck}+T'_{rj}+T'_{jr}+T'_{kc})y^2_{ckrj}$$

Objective function 2:

$$
\begin{aligned}
OF_2 = min\ \ & c'_1(\sum_{a\in I_1}\sum_{r\in R_1}\sum_{g_{111}\in J_1} x^1_{arg111} + \sum_{b\in I_2}\sum_{r\in R_2}\sum_{g_{221}\in J_1} x^1_{brg221} \\
& + \sum_{a\in I_1}\sum_{g'_{11}\in J_2} {x'}^1_{ag'_{11}} + \sum_{b\in I_2}\sum_{g'_{21}\in J_2} {x'}^1_{bg'_{21}} + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_1}\sum_{g_{131}\in J_3} y^1_{ckrg_{131}} \\
& + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_2}\sum_{g_{231}\in J_3} y^1_{ckrg_{231}}) + c'_2(\sum_{a\in I_1}\sum_{r\in R_1}\sum_{g_{112}\in J_1} x^2_{arg_{112}} \\
& + \sum_{b\in I_2}\sum_{r\in R_2}\sum_{g_{222}\in J_1} x^2_{brg_{222}} + \sum_{a\in I_1}\sum_{g'_{12}\in J_2} {x'}^2_{ag'_{12}} + \sum_{b\in I_2}\sum_{g'_{22}\in J_2} {x'}^2_{bg'_{22}} \\
& + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_1}\sum_{g_{132}\in J_3} y^2_{ckrg_{132}} + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_2}\sum_{g_{232}\in J_3} y^2_{ckrg_{232}}) \\
\\
& + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_1}\sum_{j\in J_3}\sum_{l\in L} c''_{kr} y^l_{ckrj} + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_1}\sum_{j\in J_3}\sum_{l\in L} c''_{rk} y^l_{ckrj} \\
& + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_2}\sum_{j\in J_3}\sum_{l\in L} c''_{kr} y^l_{ckrj} + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_2}\sum_{j\in J_3}\sum_{l\in L} c''_{rk} y^l_{ckrj} \\
& + \sum_{a\in I_1}\sum_{r\in R_1}\sum_{j\in J_1}\sum_{l\in L} c_r x^l_{arj} + \sum_{b\in I_2}\sum_{r\in R_2}\sum_{j\in J_1}\sum_{l\in L} c_r x^l_{brj} \\
& + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_2}\sum_{j\in J_3}\sum_{l\in L} c_k y^l_{ckrj} + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_1}\sum_{j\in J_3}\sum_{l\in L} c_k y^l_{ckrj} + \sum_{i\in I} c'''_i s_i
\end{aligned}
\tag{2}
$$

subject to

$$\sum_{r\in R} w'_r = 3, \tag{3}$$

$$\sum_{r\in R_z} w'_r = 1, \qquad \forall z \in Z \tag{4}$$

$$\sum_{r\in R_z} w_{tr} = 1, \qquad \forall z \in Z, \forall t \in T_z \tag{5}$$

$$w_{tr} \leqslant w'_r, \qquad \forall z \in Z, \forall r \in R_z, \forall t \in T_z \tag{6}$$

$$x^l_{arg_{11l}} \leqslant w'_r, \qquad \forall l \in L, \forall r \in R_1, \forall a \in I_1, \forall g_{11l} \in J_1 \tag{7}$$

$$x^l_{brg_{22l}} \leqslant w'_r, \qquad \forall l \in L, \forall r \in R_2, \forall b \in I_2, \forall g_{22l} \in J_1 \tag{8}$$

$$y^l_{ckrg_{13l}} \leqslant \frac{1}{2}(w'_k + w'_r), \qquad \forall l \in L, \forall k \in R_3, \forall r \in R_1, \forall c \in I_3, \forall g_{13l} \in J_3 \tag{9}$$

$$y^l_{ckrg_{23l}} \leqslant \frac{1}{2}(w'_k + w'_r), \qquad \forall l \in L, \forall k \in R_3, \forall r \in R_2, \forall c \in I_3, \forall g_{23l} \in J_3 \tag{10}$$

$$\sum_{a\in I_1}\sum_{r\in R_1} x^l_{arg_{11l}} = 1, \qquad \forall l \in L, \forall g_{11l} \in J_1 \tag{11}$$

$$\sum_{b\in I_2}\sum_{r\in R_2} x^l_{brg_{22l}} = 1, \qquad \forall l \in L, \forall g_{22l} \in J_1 \tag{12}$$

$$\sum_{a\in I_1} {x'}^l_{ag'_{1l}} = 1, \qquad \forall l \in L, \forall g'_{1l} \in J_2 \tag{13}$$

$$\sum_{b\in I_2} {x'}^l_{bg'_{2l}} = 1, \qquad \forall l \in L, \forall g'_{2l} \in J_2 \tag{14}$$

$$\sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_1} y^l_{ckrg_{13l}} = 1, \qquad \forall l \in L, \forall g_{13l} \in J_3 \tag{15}$$

$$\sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_2} y^l_{ckrg_{23l}} = 1, \qquad \forall l \in L, \forall g_{23l} \in J_3 \tag{16}$$

$$\sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_1} y^l_{ckrg_{13l}} = \sum_{r\in R_1} w_{g_{13l}r}, \qquad \forall l \in L, \forall g_{13l} \in J_3 \tag{17}$$

$$\sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_2} y^l_{ckrg_{23l}} = \sum_{r\in R_2} w_{g_{23l}r}, \qquad \forall l \in L, \forall g_{23l} \in J_3 \tag{18}$$

$$y^l_{ckrg_{13l}} \leqslant \frac{1}{2}(w_{ck} + w_{g_{13l}r}), \qquad \forall l \in L, \forall r \in R_1, \forall k \in R_3, \forall c \in I_3, \forall g_{13l} \in J_3 \tag{19}$$

$$y^l_{ckrg_{23l}} \leqslant \frac{1}{2}(w_{ck} + w_{g_{23l}r}), \qquad \forall l \in L, \forall r \in R_2, \forall k \in R_3, \forall c \in I_3, \forall g_{23l} \in J_3 \tag{20}$$

$$\begin{aligned} &\sum_{g_{111}\in J_1}\sum_{r\in R_1} w_{g_{111}r} + \sum_{g_{112}\in J_1}\sum_{r\in R_1} w_{g_{112}r} + \sum_{g_{221}\in J_1}\sum_{r\in R_2} w_{g_{221}r} + \sum_{g_{222}\in J_1}\sum_{r\in R_2} w_{g_{222}r} \\ &+ \sum_{a\in I_1}\sum_{g'_{11}\in J_2} {x'}^1_{ag'_{11}} + \sum_{a\in I_1}\sum_{g'_{12}\in J_2} {x'}^2_{ag'_{12}} + \sum_{b\in I_2}\sum_{g'_{21}\in J_2} {x'}^1_{bg'_{21}} + \sum_{b\in I_2}\sum_{g'_{22}\in J_2} {x'}^2_{bg'_{22}} = f', \end{aligned} \tag{21}$$

$$\begin{aligned} &\sum_{a\in I_1}\sum_{r\in R_1}\sum_{g_{111}\in J_1} x^1_{arg_{111}} + \sum_{a\in I_1}\sum_{r\in R_1}\sum_{g_{112}\in J_1} x^2_{arg_{112}} + \sum_{a\in I_1}\sum_{g'_{11}\in J_2} {x'}^1_{ag'_{11}} + \sum_{a\in I_1}\sum_{g'_{12}\in J_2} {x'}^2_{ag'_{12}} \\ &+ \sum_{b\in I_2}\sum_{r\in R_2}\sum_{g_{221}\in J_1} x^1_{brg_{221}} + \sum_{b\in I_2}\sum_{r\in R_2}\sum_{g_{222}\in J_1} x^2_{brg_{222}} + \sum_{b\in I_2}\sum_{g'_{21}\in J_2} {x'}^1_{bg'_{21}} + \sum_{b\in I_2}\sum_{g'_{22}\in J_2} {x'}^2_{bg'_{22}} = f', \end{aligned} \tag{22}$$

$$\sum_{g_{131}\in J_3}\sum_{r\in R_1} w_{g_{131}r} + \sum_{g_{132}\in J_3}\sum_{r\in R_1} w_{g_{132}r} + \sum_{g_{231}\in J_3}\sum_{r\in R_2} w_{g_{231}r} + \sum_{g_{232}\in J_3}\sum_{r\in R_2} w_{g_{232}r} = f, \tag{23}$$

$$\begin{aligned} &\sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_1}\sum_{g_{131}\in J_3} y^1_{ckrg_{131}} + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_1}\sum_{g_{132}\in J_3} y^2_{ckrg_{132}} \\ &+ \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_2}\sum_{g_{231}\in J_3} y^1_{ckrg_{231}} + \sum_{c\in I_3}\sum_{k\in R_3}\sum_{r\in R_2}\sum_{g_{232}\in J_3} y^2_{ckrg_{232}} = f, \end{aligned} \tag{24}$$

$$\sum_{r\in R_1}\sum_{j\in J_1}\sum_{l\in L} x^l_{arj} + \sum_{j\in J_2}\sum_{l\in L} {x'}^l_{aj} \leqslant q_1\ s_a, \qquad \forall a \in I_1 \tag{25}$$

$$\sum_{r\in R_2}\sum_{j\in J_1}\sum_{l\in L} x^l_{brj} + \sum_{j\in J_2}\sum_{l\in L} {x'}^l_{bj} \leqslant q_2\ s_b, \qquad \forall b \in I_2 \tag{26}$$

$$\sum_{k\in R_3}\sum_{r\in R_1}\sum_{j\in J_3}\sum_{l\in L} y^{l}_{ckrj} + \sum_{k\in R_3}\sum_{r\in R_2}\sum_{j\in J_3}\sum_{l\in L} y^{l}_{ckrj} \leqslant q_3\, s_c, \qquad \forall c \in I_3 \tag{27}$$

$$w_{tr}, w'_{r}, x^{l}_{irj}, y^{l}_{ikrj}, x'^{l}_{ij}, s_i \in \{0,1\}, \ \ \forall t\in T_z, \forall r \in R_z, \forall i \in I, \forall j \in J, \forall k \in R_3, \forall l \in L \tag{28}$$

In model (1)-(28), objective function (1) is used to minimize the total route traversing time using type 1 and 2 drones, while objective function (2) is considered to minimize the overall cost associated with the cost of using the drones, the cost of transferring the drones between the hubs, the cost of handling operations in the hubs, and the opening costs of the depots. Constraint (3) is considered to locate exactly three hubs. Constraints (4) ensure that exactly one hub is located in each zone. Constraints (5) guarantee that each non-hub node (customers and depots that need to use a hub to connect) must be allocated to a hub. Constraints (6) allow a non-hub node to be assigned to a hub only if that hub is available. Constraints (7)-(10) ensure that if a hub is located, then the demands can be sent from that node. Constraints (11)-(16) satisfy that each customer is visited exactly once by a drone. Constraints (17) and (18) ensure that for each customer who receive service from third zone, a drone is transferred from the hub of third zone to the hub of the customer's zone. Constraints (19) and (20) indicate that the flow of goods between hubs exists if the corresponding allocations of a depot and a customer is done to those hubs. Constraints (21) and (22) refer to the number of customers receiving service from their own zones, while constraints (23) and (24) refer to the number of customers receiving service from the third zone. The f and f' representing the known numbers of customers served from Zone 3 and their own zones respectively, are fixed inputs to the model. The purpose of these constraints is to establish logical consistency among the assignment variables connecting customers to hubs and depots, ensuring that the total allocations exactly match the known counts of customers in each category. Constraints (25)-(27) indicate the capacity limit of the depots. Finally, Constraints (28) determine the nature of the decision variables.

Our proposed model is designed to minimize drone flight time and overall costs. The model's primary focus is on optimizing drone operations and associated costs, and cargo plane transfer time is considered part of the supporting logistics process outside the scope of our model. Cargo plane transfer times are typically long and highly variable across operational contexts, and no consistent, empirically validated data are available. Including these uncertain parameters would unnecessarily increase model complexity and reduce tractability. Therefore, cargo plane transfer time is excluded to maintain focus on drone operations and preserve model solvability.

The main difference between the model proposed in this paper and classical p-hub models lies in the nature of the hub location decision and the network structure. In typical p-hub models, the goal is to select p-hubs from all candidate sites without regional restrictions, allowing multiple hubs in one area and none in another. In contrast, our model divides the network into three predefined zones and includes constraints that enforce the selection of exactly one hub per zone, resulting in a total of three hubs. The integration of both drones and cargo aircraft for delivery, the consideration of heterogeneous drone capacities, and the inter-zone supply policy introduce additional operational complexities that distinguish this model from classical p-hub formulations. Thus, the model determines the optimal subset of candidate hubs that minimizes the total drone flight time and overall costs. In other words, the model identifies which candidate location in each zone should be chosen as a hub, as well as the assignment of depots to customers and the optimal routing.

Table 2: Notations used in the formulation of the proposed problem 1

Notation	Type	Description
Z	Set	Set of zones (indexed by u,v,z=1,2,. . . ,Z)
R	Set	Set of candidate nodes (indexed by r=1,2,. . . ,R)
R_z	Set	Set of candidate nodes in Zone z (indexed by r=1,2,. . . , R_z)
T_z	Set	Set of customers and depots located in Zone z that should be connected via the hub of the zone
L	Set	Set of drone types
I	Set	Set of depots
I_z	Set	Set of depots located in Zone z
J	Set	Set of customers
J_1	Set	Set of customers who receive their demand from their zone using the hub
J_2	Set	Set of customers who receive their demand from their zone directly from a depot
J_3	Set	Set of customers who receive their demand from the third zone
g_{uvl}	Index	Index used for a customer located in Zone u receiving service from Zone v using a type l drone that uses a hub to connect to the depot
g'_{ul}	Index	Index used for a customer located in Zone u receiving service from Zone u using a type l drone that does not use a hub to connect to the depot
f	Parameter	Total number of customers receiving service from another zone; provided as input
f'	Parameter	Total number of customers receiving service from their zone provided as input
q_z	Parameter	Maximum capacity of each depot in Zone z
T_{mn}	Parameter	Travel time of type 1 drone from node m to node n
T'_{mn}	Parameter	Travel time of type 2 drone from node m to node n
c'_l	Parameter	Cost of using type l drone
c_r	Parameter	Cost of handling operations in hub r for one drone
c'''_i	Parameter	Operational cost of depot i if it is active (serves at least one customer)
c''_{kr}	Parameter	Cost of transferring a drone from hub k to hub r

Table 3: Notations used in the formulation of the proposed problem 2

w_{tr}	Variable	A binary variable, its value is 1 if non-hub node t is allocated to hub r, and 0 otherwise
$w_{g_{uvl}r}$	Variable	A binary variable, its value is 1 if customer located in Zone u receiving service from Zone v using a type l drone is allocated to hub r, and 0 otherwise
w'_r	Variable	A binary variable, its value is 1 if node r is a hub, and 0 otherwise
x^l_{irj}	Variable	A binary variable, its value is 1 if customer j receives the demand from depot i through hub r with drone of type l, and 0 otherwise
$x^l_{irg_{uvl}}$	Variable	A binary variable, its value is 1 if customer located in Zone u receiving service from Zone v using a type l drone, is supplied from depot i through hub r, and 0 otherwise
y^l_{ikrj}	Variable	A binary variable, its value is 1 if customer j receives the demand from depot i through hub k to hub r with drone of type l, and 0 otherwise
$y^l_{ikrg_{uvl}}$	Variable	A binary variable, its value is 1 if customer located in Zone u receiving service from Zone v using a type l drone, is supplied from depot i through hub k to hub r , and 0 otherwise
x'^l_{ij}	Variable	A binary variable, its value is 1 if customer j receives the demand from depot i with drone of type l, and 0 otherwise
$x'^l_{ig'_{ul}}$	Variable	A binary variable, its value is 1 if customer located in Zone u receiving service from Zone u using a type l drone, is supplied from depot i, and 0 otherwise
s_i	Variable	A binary variable, its value is 1 if depot i is opened, and 0 otherwise

2.3 On the complexity of the proposed formulation

Solution complexity of the formulation (1)-(28) can be studied experimentally in this sub-section. For this aim, three instances of this problem with 42, 115, and 219 nodes are considered. The objective functions are integrated by a global criterion method and the instances are solved by CPLEX solver of GAMS software on a PC with an Intel Core i7 and 2.8 GHz processor. The results are summarized in Table 4. It can be seen from this table that the instances with 42 and 115 nodes are solved optimally in a few seconds but when the size is increased to 219 nodes, GAMS is unable to solve it even after 15 hours and still 5.75 % optimality gap exists. Therefore, it can be concluded that for larger size instances, running time and optimality gap may be increased. To ensure fair and consistent comparisons, all metaheuristic algorithms were run under equal time limits. This approach enables practical and effective solution of large-scale problems where exact methods fail.

Table 4: The solutions found by GAMS for the proposed formulation

Test No.	Nodes	Constraint	Variables	*CPU running time*	Optimality gap (%)	Objective function value
1	42	1016	7666	2 *s*	0	0.0653
2	115	4983	46,696	4 *s*	0	0.0658
3	219	36,480	249,475	$>$ 15 *hours*	5.75	0.1072

3 Solution methodology

In this section, the formulation presented in Section 2 is solved. According to the experiments and results of Table 4, the proposed formulation has a high degree of complexity to be solved. Therefore, use of exact solution approaches of the available optimization solvers may not be efficient for large size instances of this problem. Therefore, in this section, some meta-heuristic solution approaches are proposed to solve the formulation (1)-(28) effectively. In order to apply meta-heuristic solution approaches to the formulation (1)-(28), first an effective encoding-decoding scheme and some neighborhood search operators are proposed. Then, some classical meta-heuristic approaches such as GA, DE, SA are considered, and their hybrid versions are developed. These steps are explained in the rest of this section.

3.1 Encoding-decoding scheme

Encoding-decoding is an important step in developing a meta-heuristic algorithm that represents a solution properly. To generate and assess a solution for the problem of Section 2, which is formulated as the model (1)-(28), the following steps are proposed.

Step 1. An initial solution is generated to solve the proposed model. Solution is a matrix with seven rows in which the length of each row is equal to the number of customers.

Customer's number and Customer's location are shown in the first and second rows, respectively. The zone that satisfies the need of customer is shown in the third row. The type of drone that is allocated for the customer's request is shown in the fourth row. The first four rows of the solution matrix are constructed based on the assumptions of the problem.

The hubs assigned to the customers and the hubs assigned to the depots of the third zone are shown in the fifth and sixth rows, respectively. The depot assigned to each customer is presented in the seventh row. The values of these three rows in the solution matrix are randomly generated. In choosing the depot, careful consideration should be given to the capacity; so that, the solution to the problem is feasible.

Step 2. With the aid of the results of Step 1, the value of the objective function can be calculated rapidly by equations (1) and (2) in the presented model. The objective functions are integrated by a global criterion method. Suppose that H and $\overline{H}$ represent the cost objective function and a lower bound for the cost objective function, respectively. Also, let T and $\overline{T}$ represent the time objective function and a lower bound for the time objective function, respectively. The proposed model is changed into a single

Table 5: The proposed solution representation

Row1	13	14	15	16	17	18	19	20	21	22	23	24
Row2	1	1	1	1	1	1	2	2	2	2	2	2
Row3	1	1	3	1	1	3	2	3	2	3	2	2
Row4	2	2	1	1	2	1	1	2	1	2	1	2
Row5	2	2	2	0	2	2	4	4	0	4	4	4
Row6	0	0	5	0	0	5	0	5	0	5	0	0
Row7	7	7	12	8	8	11	10	12	9	11	10	9

objective function as follows:

$$Z = \frac{H - \overline{H}}{\overline{H}} + \frac{T - \overline{T}}{\overline{T}}. \tag{29}$$

In the integrated objective function (29), the lower bound of each of the original objective functions of the model (1)-(28) is obtained by solving the model considering only that objective function. The objective functions in our model are integrated using a global criterion method based on the lower bounds of each individual objective. This normalization strategy avoids distortions due to differences in the scales and units of the objectives, and enables a fair and meaningful combination of both objectives. Using the lower bound as a reference provides a scientifically sound and interpretable basis for comparison and leads to increased stability in the optimization process. The selected approach ensures full compatibility with the problem structure and enables stable and effective performance of the employed metaheuristic algorithms. Therefore, the chosen normalization method strikes an effective balance between simplicity, interpretability, computational efficiency, and alignment with the structure of the problem, and has led to reliable and valid results.

An example of a solution matrix with 24 nodes including 12 customers in the first and second zones, six depots and six candidate hub points in the first, second and third zones is illustrated by Table 5. In Table 5, consider the column related to customer 20. The second row states that the customer is located in Zone 2. The third row specifies that Zone 3 meets the demand of the customer. The fourth row shows that the type 2 drone is allocated according to the amount of the customer's request. The fifth row states that hub four is assigned to the customer. The sixth row specifies that hub five is assigned to the depots of the third zone. The seventh row states that depot 12 is assigned to the customer.

3.2 Neighborhood search operators

There is a close connection between neighborhood search operators and the performance of any metaheuristic algorithm. We apply all the operators on the seventh row of the solution matrix and in the same zone so that the solution to the problem is feasible. Due to the problem's structure and complexities related to customer assignments, heterogeneous drone types, and payload capacity constraints, the operators are restricted to act only on the seventh row and independently within each zone. This limitation is imposed to maintain model tractability and control the search space size. Restricting operators to act within each zone independently reduces computational complexity and facilitates the optimization process, while maintaining solution quality at an acceptable level, and empirical results demonstrate satisfactory performance under these constraints. For any meta-heuristic algorithm in this study, one or some of the below operators can be used.

Insertion : This operator randomly selects and eliminates point i and inserts it in position j. These two locations must exist in the same zone. Considering row 7 in Table 5, which represents the depot allocated to each customer, the two selected points must belong to the first or second or third zone. If the two selected points are related to the first zone, then i and j points, representing depots number 7 and 8, are selected from the first, second, fourth and fifth columns.

This operator effectively modifies the solution structure without violating problem constraints, enabling a more focused search within the solution space. Therefore, it is used as one of the key neighborhood search operators in all employed algorithms and accelerates convergence.

Consider the seventh row of the answer matrix in Table 5:

$$\left[a_{71}, a_{72}, a_{73}, a_{74}, a_{75}, a_{76}, a_{77}, a_{78}, a_{79}, a_{710}, a_{711}, a_{712}\right]$$

$$= \left[7, 7, 12, 8, 8, 11, 10, 12, 9, 11, 10, 9\right],$$

$$\left[a_{71}, a_{72}, a_{74}, a_{75}\right] = \left[7, 7, 8, 8\right] \xrightarrow{\text{Applying the insertion operator}} \left[7, 8, 8, 7\right].$$

Final result of the insertion operator:

$$\left[7, 8, 12, 8, 7, 11, 10, 12, 9, 11, 10, 9\right].$$

Reversion : By using this operator, two points are selected in a random order. The sequence of these two points is changed in reverse order. These two points must exist in the same zone. Considering row 7 in Table 5, which represents the depot allocated to each customer, the two selected points must belong to the first or second or third zone. If the two selected points are related to the second zone, then i and j points, representing depots number 9 and 10, are selected from the seventh, ninth, eleventh and twelfth columns.

It is worth noting that the reversion operator is applied with the same zonal restriction to ensure the feasibility of allocations. This constraint restricts the reversal of the sequence to points within the same specified zone (Zone one, two, or three), thereby preserving the logical structure and capacity constraints of the problem. This approach helps prevent infeasible solutions and allows for a more focused and efficient search within the solution space. Consequently, this operator not only maintains problem feasibility but also contributes to solution diversity and enhances the exploratory capability of the algorithms, playing a significant role across all applied methods.

$$\left[7, 7, 12, 8, 8, 11, 10, 12, 9, 11, 10, 9\right],$$

$$\left[a_{77}, a_{79}, a_{711}, a_{712}\right] = \left[10, 9, 10, 9\right] \xrightarrow{\text{Applying the inversion operator}} \left[10, 10, 9, 9\right].$$

Final result of the inversion operator:

$$\Big[7,7,12,8,8,11,10,12,10,11,9,9\Big].$$

Swap : In the swap operator, two random points i and j are chosen, and their positions are rearranged. These two points must exist in the same zone. Considering row 7 in Table 5, which represents the depot allocated to each customer, the two selected points must belong to the first or second or third zone. If the two selected points are related to the third zone, then i and j points, representing depots number 11 and 12, are selected from the third, sixth, eighth and tenth columns.

The swap operator is also applied with the same zonal restriction as other operators to ensure that allocations remain within the same zone, preventing infeasible solutions. This constraint ensures that swapping the positions of two points occurs only within a single zone. This approach guarantees a more effective and focused exploration of the search space and improves the quality of solutions. Therefore, this operator not only preserves solution feasibility but also contributes to solution diversity and plays a key role in all proposed algorithms:

$$\Big[7,7,12,8,8,11,10,12,9,11,10,9\Big],$$

$$\Big[a_{73},a_{76},a_{78},a_{710}\Big]=\Big[12,11,12,11\Big]\xrightarrow{\text{Applying the swap operator}}\Big[11,11,12,12\Big].$$

Final result of the swap operator:

$$\Big[7,7,11,8,8,11,10,12,9,12,10,9\Big].$$

Mutation : Swap is applied on a solution with mutation probability of P_m.

This mutation probability helps increase the diversity of the solution population and prevents premature convergence of the algorithm. Similar to other operators, the zonal restriction is enforced in mutation to ensure that swaps occur only within the same zone, preserving the problem structure and capacity constraints. This approach contributes to improving solution quality and the robustness of the search process, playing a significant role in the proposed hybrid algorithms:

$$\Big[7,7,12,8,8,11,10,12,9,11,10,9\Big],$$

$$\Big[a_{71},a_{72},a_{74},a_{75}\Big]=\Big[7,7,8,8\Big]\xrightarrow{\text{Applying the swap operator}}\Big[7,8,7,8\Big].$$

Final result of the mutation operator:

$$\Big[7,8,12,7,8,11,10,12,9,11,10,9\Big].$$

Crossover : In this step, two new solutions (offspring) are created with the aid of two solutions (parents). One of the zones is randomly selected, and the parents exchange the solution vectors in the same zone.

This exchange is restricted to the selected zone to ensure that the problem structure and capacity constraints within each zone are preserved. In this way, Crossover can combine beneficial features from both parents, contributing to population diversity and overall solution quality improvement. As with other operators, the zonal restriction is applied here as well to prevent the generation of infeasible solutions.

Consider the seventh row of the two-parent answer matrix:

$$\text{Parent 1:} \qquad \left[7,7,12,8,8,11,10,12,9,11,10,9\right],$$

$$\text{Parent 2:} \qquad \left[8,7,11,8,7,12,9,12,10,11,9,10\right].$$

We consider the first region in the two parents:

$$\left[a_{71},a_{72},a_{74},a_{75}\right]=\left[7,7,8,8\right], \quad \left[b_{71},b_{72},b_{74},b_{75}\right]=\left[8,7,8,7\right].$$

Final result of the crossover operator:

$$\text{offspring 1:} \qquad \left[8,7,12,8,7,11,10,12,9,11,10,9\right],$$

$$\text{offspring 2:} \qquad \left[7,7,11,8,8,12,9,12,10,11,9,10\right].$$

Finally, it is worth noting that the overall performance of the proposed algorithms, each of which utilizes at least one of these neighborhood search operators, is comprehensively analyzed and compared in the results section of this paper. These analyses demonstrate the effectiveness and efficiency of these operators in enhancing the solution.

3.3 Meta-heuristic solution approaches

Meta-heuristic approaches are of great importance to solve large-scale and complex optimization problems [1, 22, 26, 61]. Here, some classical meta-heuristic approaches such as GA, DE and SA are considered, and their hybrid versions are developed for solving the formulation (1)-(28). Any solution in each of the proposed algorithms is represented and evaluated by the encoding-decoding scheme of Subsection 3.1 Also, in any of the proposed algorithms, at least one of the proposed neighborhood search operators of Subsection 3.2 is used.

3.3.1 The SA

The SA was introduced by Kirkpatrick et al. [30]. This algorithm is based on a single solution. The SA uses an iterated local search optimization technique. At the beginning, the parameters such as initial temperature, cooling ratio, final temperature, and number of iterations in each temperature are determined. In the first iteration of the initial temperature, a random solution is generated as the current solution, and is evaluated according to the scheme of Subsection 3.1 Then, one of the swap, reversion, and insertion operators is randomly selected and applied to the current solution to generate a neighbor solution. If the neighbor solution is better than the current solution, it is saved as the best solution and also as the current solution. If not, it is just saved as the current solution with a generated probability. All iterations of the initial temperature are done, and the initial temperature is cooled down by the cooling ratio. The procedure is repeated till reaching the final temperature or reaching a time limit. Finally, the best solution is reported as the output of the algorithm.

In our implementation of the SA algorithm, the primary focus is on maintaining the feasibility of solutions and designing effective local moves. In each iteration, one of the neighborhood search operators, specifically tailored to the problem structure, is randomly selected and applied. These operators only induce changes that do not violate the core constraints of the problem, such as depot capacity limits. This design ensures that the SA algorithm operates exclusively within the feasible solution space. The stochastic nature of SA provides a strong ability to escape local optima and explore better regions with respect to the objective function. This characteristic makes SA a suitable and practical choice for solving the proposed problem. Details of the SA algorithm is exhibited by the pseudo-code in Algorithm 1.

Algorithm 1 Simulated Annealing (SA)

Set parameters:
  an initial temperature (T_0)
  cooling ratio (α)
  number of iterations of the inner loop (maxsubit)
  number of iterations of the outer loop (maxit)
Create an initial solution a_0
$a \leftarrow a_0$
Evaluate its fitness z
$T \leftarrow T_0$
while termination criterion not satisfied
  for subit = 1 to maxsubit
    Generate new solution a' from neighbor search K
    $a' = N_k(a)$
  end for
  if $z(a') < z(a)$ **then**
    $a \leftarrow a'$
  else if rand(0,1) $\leq \exp(-\Delta f/T)$ **then**
    $a \leftarrow a'$
  end if
  $T \leftarrow \alpha T$
end while
Report the best obtained solution

3.3.2 The GA

The GA was first introduced by Holland [23]. Due to great flexibility, easy operation, and simplicity, the GA is a search technique that can respond to a wide variety of complex optimization problems. It is a nondeterministic method based on the Darwin's evolutionary theory. As the GA is a population-based mechanism, in this algorithm,two child solutions are generated from any pair of parent solutions. In order to initiate the GA, a randomly generated population of solutions (chromosomes) is determined. By evolution of the chromosomes through successive iterations (generations), the algorithm converges to the best solution. By using some measures of fitness (objective function value), the chromosomes are evaluated during each generation.

In the first iteration, two parents are selected from the population using a roulette wheel mechanism, where the probability of selection is proportional to the fitness value of each chromosome, thus favoring better solutions in the generation of offspring. Two solutions (offsprings or child) are generated from the selected parents using the crossover operator (explained in Section 3.2). Then another solution is selected from the population randomly and by applying the mutation operator with probability of P_m, its neighbor solution is obtained. The generated solutions are added to the population, the population is sorted according to the fitness values, and the new population is selected [18]. The algorithm is continued for several iterations (generations) or till a time limit. The best solution of the last generation is the output of the GA. In the implementation of the genetic algorithm, the initial population is generated randomly but in compliance with problem constraints to ensure the feasibility of initial solutions. The crossover and mutation operators are designed to introduce diversity while preventing violation of core constraints such as depot capacities and regional allocation restrictions. The population-based nature and diversity mechanisms of GA make it a suitable algorithm for extensive search in the complex solution space of this problem, especially considering capacity constraints and multi-region structure that enlarge and complicate the search space. This algorithm is indicated by the pseudo-code in Algorithm 2.

Algorithm 2 Genetic Algorithm (GA)

Set parameters:
  population size ($npop$)
  crossover rate (P_c)
  mutation rate (P_m)
  maximum number of iterations ($maxit$)
Create the initial population
Evaluate the fitness value of each individual
while the termination criterion is not satisfied
  $nc = 2 \times \text{round}(P_c \times npop/2)$
  for $k = 1$ to $nc/2$
    Select first parent
    Select second parent
    Apply crossover
    Evaluate offsprings
  end for
  $nm = \text{round}(P_m \times npop)$
  for $k = 1$ to nm
    Select parent
    Apply mutation
    Evaluate mutant
  end for
  Merge the population with the offspring created by crossover and mutation operators
  Sort population
  Truncate the $npop$ number of the best chromosomes in the sorted population
end while
Report the best chromosome

3.3.3 The DE

The DE algorithm is a population-based algorithm. This algorithm was developed by Storn and Price [51]. The DE optimizes a problem using some candidate solutions. These solutions are improved using an iterative method concerning a given measure of quality [6].

The evolution process of the DE is similar to that of the GA including selection, and the crossover and mutation operators. The efficient evolutionary algorithm of the DE has a clear structure. The DE starts with a randomly generated set of solutions called initial population where each solution is generated and evaluated according to the scheme of Subsection 3.1.

For each population member (target) in each iteration, three random different solutions from the present population are mutated to create a mutant vector. Then, a new solution is obtained by crossover of target and mutant. If the new solution is better than the current one (target), the target is replaced by a new solution with better fitness value; otherwise, the current solution is passed on to the next generation. If a stopping criterion is satisfied, the algorithm terminates, and the obtained population is reported.

The mutation and crossover operators are carefully designed to maintain the feasibility of solutions, particularly respecting depot capacity limits and regional allocation restrictions. This approach ensures that the DE algorithm performs an effective search within the feasible solution space. The population-based nature of DE, combined with its mutation and crossover mechanisms, enables efficient exploration of the large and complex search space. Moreover, DE is favored for its relatively fast convergence and ease of implementation, making it a suitable and practical choice for the proposed optimization problem. The DE algorithm is illustrated by the pseudo-code in Algorithm 3.

Algorithm 3 Differential Evolution (DE)

Set parameters:
  population size ($npop$)
  lower bound of scaling factor ($\beta_{\min}$)
  upper bound of scaling factor ($\beta_{\max}$)
  crossover probability (P_{cr})
  maximum number of iterations ($maxit$)
Create the initial population
Evaluate the fitness value of each individual
while the termination criterion is not satisfied
  for each individual x in the population
    Select randomly three different solutions a, b and c from the present population
    Apply mutation to generate a temporary solution (y):
      $y = a + (b - c) \times \text{unifrnd}(\beta_{\min}, \beta_{\max})$
    Apply crossover to generate a new solution (z):
      $z = \begin{cases} x, & \text{if rand} \le P_{cr} \\ y, & \text{otherwise} \end{cases}$
    Evaluate fitness value of new solution
    if new solution is better than old solution x **then**
      Replace x with z
    else
      Pass the old solution to the next generation
    end if
  end for
end while
Report the best chromosome

3.3.4 Serial hybrid GA-SA

In order to improve performance of the GA and SA algorithms and obtain better solution for the problem of Section 2, these algorithms are hybridized here. Finding a near to best solution in short time is an advantage of the GA, but it may not escape the local minimum when size of problem is increased. On the other hand, although the SA provides effective local zone search, there is no assurance that an optimal solution can be achieved. Hybridizing the GA and SA algorithms in a serial format results in an algorithm with advantages of both of the GA and SA algorithms. In this hybrid algorithm, first the GA is executed, and then the SA is performed. It is notable that, the SA starts with the best solution obtained by the

GA instead of generating a random solution. This issue may help the SA to introduce a better solution compared to its classical format [53, 56].

The GA is suitable for the initial search phase due to its ability to explore the solution space broadly and converge relatively quickly to near-optimal solutions; however, it may get trapped in local optima as the problem size increases. On the other hand, SA performs effective local search, which can refine solutions, but starting from a random solution may result in longer convergence times or suboptimal results. The hybrid method benefits from GA's strong exploration capability and SA's effective exploitation, enhancing solution quality in less time. This hybrid design carefully considers problem constraints to ensure that solutions generated remain feasible. This approach enhances the overall efficiency of the combined algorithm and its applicability to complex logistics optimization problems with multi-regional structures.

3.3.5 Parallel hybrid GA-SA

This hybrid algorithm combines the GA and SA in parallel to obtain better solutions comparing to their classical formats. In the GA, the neighborhood search operators are used to search for the solution globally. So, exploration of solutions may be badly affected by large search spaces and premature convergence. As a result, the new solutions obtained by the GA operators may be mainly accepted, even if they are considerably inferior to other solutions of the population. This feature can lead to disruption and damaged solutions. Here, the SA algorithm is considered to improve the GA. In this parallel hybrid algorithms, the GA is performed as basic algorithm. Then, for each offspring generated by the GA, the SA is performed in order to improve the offspring. This algorithm utilizes the GA's ability to build the global solution and the SA's ability to refine each specific solution locally.

The parallel hybrid GA-SA algorithm is specifically designed to simultaneously leverage the global search capabilities of GA and the local refinement strengths of SA. This parallel approach not only improves solution quality but also maintains population diversity, preventing premature convergence traps. Additionally, it accelerates convergence while ensuring all problem-specific constraints including depot capacities and multi-regional supply requirements are strictly respected throughout both GA and SA phases to maintain solution feasibility. This hybrid achieves an optimal balance between exploration and exploitation, delivering superior performance compared to classical algorithms and effectively solving the complex problem with high-quality and efficient solutions.

3.3.6 Parallel hybrid GA-SA using SA rule

The initial population is randomly created, and the initial temperature is adjusted. We use the crossover and mutation operators to create the offspring population. After merging the population resulting from crossover and mutation, we sort it. Now, we have to separate the size of the initial population from the sorted population and compare its members with the members in the initial population, and this comparison is done with the SA model. In fact, we use the SA model to create the new generation population. The creation and integration of the offsprings must be done at a constant temperature and we reduce the temperature at the end of each iteration. The probability of accepting non-improved new solutions is reduced as the temperature is decreased.

The main innovation of this parallel hybrid approach lies in utilizing the SA model as an integrated decision-making mechanism for accepting or rejecting members of the new generation. Unlike traditional methods where SA is applied solely as a local improvement step after GA generations, in this algorithm, SA plays a direct role within the generation selection process. This integration not only enhances the quality of obtained solutions but also preserves population diversity and prevents premature convergence. The rationale behind this combination is based on the complementary strengths of the two algorithms: GA's capability for broad exploration across the solution space enables discovery of diverse regions, while SA's probabilistic acceptance mechanism allows escape from local optima and focused local refinement. Employing SA during the new generation selection means that even inferior solutions can be accepted with a certain probability, fostering exploration and maintaining diversity.

3.3.7 Serial hybrid DE-SA

In order to improve performance of the DE and SA algorithms and obtain better solution for the problem of Section 2, these algorithms are hybridized here. Finding the near to best solution with short solution time is an advantage of the DE, but it may not escape the local minimum when size of the problem is increased. Hybridizing the DE and SA algorithms in a serial format results in an algorithm with advantages of both of the DE and SA algorithms. In this hybrid algorithm, first the DE is executed, and then the SA is performed. It is notable that, the SA starts with the best solution obtained by the DE instead of generating a random solution. This issue may help the SA to introduce a better solution compared to its classical version.

The DE, with its population-based search and differential updates, efficiently finds near-optimal solutions in a short time but may get trapped in local optima in large and complex problems. On the other hand, SA, through its precise local search and probabilistic acceptance mechanism, can escape local optima but might converge more slowly if used alone. The main novelty of this serial hybrid lies in combining the strengths of the evolutionary DE algorithm and the local optimization SA algorithm in a way that minimizes their individual weaknesses while maximizing their benefits. This approach allows the algorithm to retain DE's speed and exploration power while benefiting from SA's ability to escape local optima and improve solution quality. This approach is especially practical for complex logistical problems with vast search spaces and diverse constraints, such as the drone delivery location-routing problem.

3.3.8 Parallel hybrid DE-SA

This hybrid algorithm combines the DE and SA in parallel to obtain better solutions comparing to their classical formats. In the DE, the neighborhood search structure is used to search the solution globally. So, exploration of solutions may be badly affected by large search spaces and premature convergence. Here, the SA algorithm is considered to improve the DE. In this parallel hybrid algorithms, the DE is performed as basic algorithm. Then, for each solution generated by the DE, the SA is performed in order to improve the solution. This algorithm utilizes the DE's ability to build the global solution and the SA's ability to refine each specific solution locally.

The key innovation in this parallel hybrid algorithm lies in the simultaneous integration of DE's broad, population-based global search capabilities with SA's precise local improvement power. Unlike serial execution where the algorithms run sequentially, in this method each solution generated by DE is immediately refined using SA. This parallel integration enhances solution quality. The rationale behind this combination is based on the complementary strengths of the two algorithms: DE, effectively explores promising regions of the search space but may suffer from premature convergence in large search spaces. Meanwhile, SA's precise local search and probabilistic acceptance allow it to improve local solutions and escape local optima. This parallel execution minimizes the weaknesses of both algorithms while optimally leveraging their advantages.

4 Computational study

In this section, the proposed formulation and the proposed meta-heuristic algorithms are evaluated numerically. For this aim, the proposed meta-heuristic algorithms are coded in MATLAB and are run on a PC with an Intel Core i7, 3.60 GHz processor and 32.0 GB RAM. Some test problems are generated randomly in various sizes. The Taguchi experimental design method is utilized to tune the parameters of the proposed algorithms. Moreover, finally, the final experiments are performed to evaluate the algorithms using the generated test problems.

4.1 Test problems

A total of 16 test problems are randomly generated in this experimental study. The features related to the size of the test problems are shown in Table 6. For the following reasons, benchmark cases in optimization issues are frequently produced at random: It guarantees that the assessment of algorithms is not skewed toward particular data sets or situations and is not made for any particular set. It makes it possible to create samples with varied degrees of complexity and assess how well algorithms work in various scenarios. In situations where real-world data is limited, unreliable, or unavailable, random generation provides an alternative. A variety of scenarios, including ones that don't arise in real-world datasets, are used to test algorithms. By using random generation, researchers can create samples that are representative of a range of real-world circumstances.

The datasets used in this study were synthetically / semi-realistically generated. Since no operational dataset was publicly available that matches the structure and requirements of the problem, the instances were constructed based on reasonable spatial and logistical assumptions. Therefore, the numerical results should be interpreted as performance evaluations on semi-realistic synthetic instances rather than on data obtained from an actual operational system.

Some details about the test problems are given below:

- There are 3 zones in each test problem.
- In each zone, there are some potential places to establish only one hub.
- Each non-hub node (depot or customer) must be allocated to only maximum one hub node.
- Transportation of goods demanded by the customers are occurs within and between the zones.
- The goods are transferred within the zones using drones and between the zones by means of cargo planes that include the loaded drones.
- All demands must be satisfied.

- Two types of drones are used with different capacities and speeds.
- Each drone may visit at most one customer, and each customer is served only once by a drone.
- Each route starts from a depot and ends at the same depot. Therefore, each drone must start from and return to only one depot.
- Cost of using type l drone, cost of handling operations in hub r for one drone, cost of transferring a drone from hub k to hub r and operational cost of depot i if it is active (serves at least one customer) are considered.
- The range of drone traveling times T and T' are generated randomly from the intervals [5,30] and [10,50] respectively. The values are expressed in minutes.

4.2 Parameter tuning

The Taguchi method is used to determine the best parameter values for the proposed meta-heuristic algorithms. The Taguchi experimental design was introduced by Taguchi in 1986 [52]. Before evaluating the effectiveness of the algorithms, the Taguchi method is used to tune the parameters of the proposed algorithms and find the best combination of the parameters values in each algorithm. The Taguchi method uses orthogonal arrays with associated degrees of freedom to diminish the number of required experiments. So as, to find the optimal level of the controllable factors, Taguchi method minimizes the effect of noise by use of a response value for each experiment. In order to investigate a change in the amount of the response, the signal-to-noise ratio (S/N) is computed by the following equation:

$$\frac{S}{N} = -10(log_{10}^{(objective\ function\ value)^2}). \tag{30}$$

To determine the S/N ratio, the unscaled objective function value from the related percentage deviation (RPD) formula is used:

$$RPD = \frac{|\,Algorithm\ solution - Best\ solution\,|}{Best\ solution} \times 100. \tag{31}$$

In this study, to evaluate the effect of design parameters on the performance of metaheuristic algorithms applied to the drone-based hub location problem, we employed the Taguchi design module in Minitab. The response metric used was the relative percentage deviation, which was selected for its ability to normalize and fairly compare results across different algorithms and instances.

The obtained objective function values from 5 runs are averaged to calculate the S/N ratio of each parameter. For the proposed algorithms of this study the selected levels of the parameters, the orthogonal array used for each algorithm, and the best obtained value of each parameter by the Taguchi method is reported by Tables 7 and 8. It is notable to mention that, in order to obtain the results of Tables 7 and 8, the TP10 test problem generated in the previous sub-section is considered.

The Taguchi method is designed to significantly reduce the required number of experiments while increasing the accuracy in determining the effect of each parameter. In this study, the Taguchi experimental design method was employed to tune the parameters of the metaheuristic algorithms. The experimental

Table 6: Information of each test problem

Test problem	Total number of nodes	Number of depots	*Number* of hubs	Number of customers	Number of customers who receive their demand from Zone1	Number of customers who receive their demand from Zone2	Number of customers who receive their demand from Zone3
TP1	42	13	9	20	7	6	7
TP2	42	9	9	24	8	7	9
TP3	42	7	9	26	9	8	9
TP4	42	15	9	18	6	6	6
TP5	115	42	9	64	21	20	23
TP6	115	26	9	80	22	22	36
TP7	115	21	9	85	27	28	30
TP8	115	17	9	89	28	29	32
TP9	219	84	9	126	38	34	54
TP10	219	52	9	158	49	50	59
TP11	219	42	9	168	56	56	56
TP12	219	35	9	175	57	59	59
TP13	329	128	9	192	61	65	66
TP14	329	80	9	240	76	79	85
TP15	329	64	9	256	82	86	88
TP16	329	53	9	267	83	91	93

design utilized orthogonal arrays to reduce the number of required experiments and enable simultaneous evaluation of multiple parameters at different levels. Each algorithm's parameters were considered at three different levels, and appropriate orthogonal arrays such as L9 or L27 were selected depending on the number of parameters. This selection was based on the degrees of freedom of parameters and their levels to minimize the number of experiments efficiently and logically. This contributes significantly to the efficiency and accuracy of tuning the proposed algorithms' parameters.

4.3 Final experiments

For each algorithm, the best level of the parameters obtained by the previous sub-section is applied to solve all of the test problems. To obtain more reliable results, 15 runs are carried out for each test problem. In order to compare the algorithms, all algorithms in each test problem should be run for an equal run time. We consider the longest running time among the algorithms in each test problem as T_{max}

Table 7: Best combination of the parameters 1

			Taguchi experimental design			
Algorithm	Parameter	Proposed values	Degree of freedom	Required orthogonal array	Number of experiments	Best obtained value of the parameters
SA	T_0 α maxsubit maxit	95, 100, 105 0.85, 0.95, 0.99 10,50,100 100, 150, 250	9	L_9	9	100 0.85 100 100
GA	P_c P_m n_{pop} maxit	0.6,0.7,0.8 0.3, 0.4, 0.5 20,100,150 20, 300,400	9	L_9	9	0.7 0.5 100 400
DE	$beta_{min}$ $beta_{max}$ p_{cr} n_{pop} maxit	0.1,0.2,0.3 0.7,0.8,0.9 0.1,0.2,0.3 20, 60,100 200, 300,400	11	L_{27}	27	0.1 0.7 0.3 100 400
Serial GA-SA	P_c P_m n_{pop} maxit GA maxit SA T_0 α maxsubit	0.6,0.7,0.8 0.3,0.4,0.5 20,100,150 200, 300,400 100, 150,250 95, 100, 105 0.85, 0.95, 0.99 10,50,100	17	L_{27}	27	0.8 0.5 150 400 100 100 0.85 100
Serial DE-SA	$beta_{min}$ $beta_{max}$ p_{cr} n_{pop} maxit DE maxit SA T_0 α maxsubit	0.1,0.2,0.3 0.7,0.8,0.9 0.1,0.2,0.3 20, 60,100 200, 300,400 100, 150, 250 95, 100, 105 0.85,0.95,0.99 10,50,100	19	L_{27}	27	0.1 0.7 0.3 60 400 250 105 0.99 10

and propose the following formula to determine the stopping time of the algorithms:

$$Stopping\ Time = (\lceil \frac{T_{max}}{N} \rceil \times N) - \lceil 0.75t \rceil, \tag{32}$$

Table 8: Best combination of the parameters 2

			Taguchi experimental design			
Parallel GA-SA	P_c	0.6,0.7,0.8	17	L_{27}	27	0.8
	P_m	0.3,0.4,0.5				0.3
	n_{pop}	20,100,150				150
	maxit GA	40, 60,80				80
	maxit SA	20, 30, 50				50
	T_0	95, 100, 105				105
	α	0.85, 0.95, 0.99				0.95
	maxsubit	2,10,20				2
Parallel DE-SA	$beta_{min}$	0.1,0.2,0.3	19	L_{27}	27	0.2
	$beta_{max}$	0.7,0.8,0.9				0.7
	p_{cr}	0.1,0.2,0.3				0.2
	n_{pop}	20, 60,100				100
	maxit DE	40, 60,80				80
	maxit SA	20,30, 50				30
	T_0	95, 100, 105				105
	α	0.85,0.95,0.99				0.99
	maxsubit	2,10,20				2
Parallel GA Using SA Rule	P_c	0.6,0.7,0.8	15	L_{27}	27	0.8
	P_m	0.3, 0.4, 0.5				0.5
	n_{pop}	20,100,150				100
	maxit GA	100, 150,200				100
	T_0	95, 100, 105				95
	α	0.85,0.95,0.99				0.99
	maxsubit	5,25,50				25

where, N is the number of available nodes (the number of hubs, the number of customers, and the number of depots) and $t = (\lceil \frac{T_{max}}{N} \rceil \times N) - T_{max}$. According to this formula, the obtained stopping time is a little greater than T_{max}. All algorithms in each test problem are continued to reach the stopping time of the test problem. To ensure a fair comparison among the metaheuristic algorithms employed for solving the problem, the runtime for each algorithm is set based on the stopping time. All algorithms are thus executed under the same time budget, a standard procedure that guarantees comparability and validity of performance evaluation.

For every test problem, the minimum, the maximum and the average values of the objective function by each of the proposed algorithms are shown by Tables 9,10 and 11, respectively. With regards to the results of the Tables 9-11, in about 50% of the test problems the serial DE-SA surpasses all other algorithms and has a better ability to find the appropriate value of the objective function.

According to the last row of Table 9 (minimum value), if we arrange the algorithms from smallest value to largest value, their order is as follows: serial DE-SA, serial GA-SA, GA, parallel GA using SA rule, DE, SA, parallel GA-SA, and parallel DE-SA.

According to the last row of Table 10 (maximum value), if we sort the algorithms in ascending order, the algorithms can be considered in the following order: serial DE-SA, serial GA-SA, GA, DE, parallel

Table 9: Minimum value of the objective function in 15 runs

	Meta-heuristic approaches							
Test problem	SA	GA	DE	Serial GA-SA	Serial DE-SA	Parallel GA-SA	Parallel DE-SA	Parallel GA Using SA Rule
TP1	0.32053	0.18453	0.17733	0.19093	**0.13973**	0.14693	0.17733	0.29333
TP2	0.36084	**0.24874**	0.25495	0.28583	0.26116	0.26391	0.25495	0.30237
TP3	0.33410	0.31686	0.30307	0.30307	0.30307	**0.30306**	0.30307	0.31893
TP4	0.39602	0.23383	**0.10812**	0.14464	**0.10812**	0.15320	0.15350	0.27445
TP5	0.12373	0.08349	0.11813	0.11594	**0.06788**	0.12616	0.19763	0.10007
TP6	0.08614	0.04633	0.03855	0.03374	**0.02874**	0.08948	0.14874	0.11522
TP7	0.12669	**0.10534**	0.17804	0.10688	0.11073	0.15366	0.21905	0.12073
TP8	0.10486	0.10140	0.14967	0.10351	**0.10102**	0.15563	0.19929	0.10755
TP9	0.14803	0.09311	0.13358	0.07662	**0.06442**	0.16319	0.24412	0.12532
TP10	0.11555	0.08044	0.15633	**0.07543**	0.08343	0.17082	0.22754	0.11065
TP11	0.08918	**0.05379**	0.11152	0.05821	0.05997	0.16433	0.19852	0.07173
TP12	0.14003	0.06903	0.14043	**0.06773**	0.08823	0.23813	0.27983	0.07863
TP13	0.10198	**0.04831**	0.20456	0.07689	0.05039	0.19348	0.23623	0.10423
TP14	0.09443	**0.03071**	0.14840	0.03824	0.03437	0.16139	0.20612	0.05183
TP15	0.09797	0.04687	0.12611	0.05508	**0.02943**	0.17523	0.21709	0.07655
TP16	0.09030	**0.03326**	0.13145	0.04060	0.03945	0.17788	0.22307	0.07470
Average	0.17064	0.11100	0.15501	0.11083	**0.09813**	0.17728	0.21788	0.14539

GA-SA, parallel GA using SA rule, parallel DE-SA, and SA.

With regards to the last row of the Table 11 (average value), if we arrange the algorithms in increasing order, their order is as follows: serial DE-SA, serial GA-SA, GA, DE, parallel GA using SA rule, parallel GA-SA, parallel DE-SA, and SA.

The results of Table 9-11 are also illustrated by Figures 2-5. The reported results in this section are summarized below.

- The serial DE-SA and serial GA-SA have a better ability to find a proper value for the objective function.

- In most of the test problems, the serial hybrid versions of the DE and GA algorithm shows better performance than the DE and GA.

Table 10: Maximum value of the objective function in 15 runs

	Meta-heuristic approaches							
Test problem	SA	GA	DE	Serial GA-SA	Serial DE-SA	Parallel GA-SA	Parallel DE-SA	Parallel GA Using SA Rule
TP1	0.44773	0.27733	**0.25733**	0.26613	0.25893	0.26213	0.25893	0.35653
TP2	0.51118	0.34911	0.33325	0.33532	0.34222	0.35325	**0.33187**	0.40391
TP3	0.54721	0.33203	0.31686	0.34031	0.31686	0.32996	**0.30307**	0.41479
TP4	0.63441	0.40455	**0.28965**	0.37441	0.29822	0.41917	0.32551	0.45225
TP5	0.32351	0.17107	0.20034	0.17839	**0.14595**	0.15763	0.23178	0.25521
TP6	0.29114	0.09392	0.10504	0.09892	**0.07337**	0.11614	0.16985	0.21225
TP7	0.23073	**0.12938**	0.19996	0.13477	0.13246	0.18404	0.23385	0.28438
TP8	0.32697	0.13063	0.17832	0.11101	**0.10929**	0.17813	0.22063	0.23928
TP9	0.24011	0.16180	0.23927	0.13016	**0.10586**	0.20414	0.26586	0.20173
TP10	0.21117	0.11917	0.21117	**0.10726**	0.12708	0.19831	0.24165	0.20076
TP11	0.18470	0.07991	0.17746	0.08097	**0.07746**	0.18542	0.22022	0.15152
TP12	0.26273	0.12133	0.21813	0.11883	**0.11333**	0.25813	0.30023	0.21363
TP13	0.17939	**0.10381**	0.24923	0.12923	0.11281	0.22064	0.26315	0.16123
TP14	0.13721	**0.05150**	0.19763	0.07275	0.05975	0.18030	0.21665	0.10237
TP15	0.14348	0.06384	0.17523	0.06631	**0.06011**	0.19945	0.22980	0.10036
TP16	0.13844	0.05795	0.18814	0.06621	**0.05657**	0.20144	0.23488	0.12054
Average	0.30063	0.16545	0.22106	0.16318	**0.14939**	0.22801	0.25299	0.24192

As an example Figure 6 shows a network for a solution obtained by GAMS for test problem 3 with 26 customers, seven depots and three hubs. Table 12 shows all the movement paths of Figure 6 from the depot to the customer. Since the customers served by depots 11 and 13 belong to the first group whose needs are satisfied from their zone, these depots are not connected to any hubs.

5 Concluding remarks

In this work, to address the hub location and the drone delivery problem, we presented a bi-objective integer linear programming formulation. The main objectives that we intended were (i) minimizing the overall cost, (ii) minimizing the total route traversing time using drones. Due to the high computational

Table 11: Average value of the objective function in 15 runs

	Meta-heuristic approaches							
Test problem	SA	GA	DE	Serial GA-SA	Serial DE-SA	Parallel GA-SA	Parallel DE-SA	Parallel GA Using SA Rule
TP1	0.38341	0.24634	0.21733	0.22815	**0.21034**	0.21146	0.22063	0.32613
TP2	0.43716	0.31092	**0.29419**	0.30692	0.30881	0.30511	0.30287	0.36215
TP3	0.47281	0.32251	0.30674	0.31870	0.30674	0.31640	**0.30307**	0.36830
TP4	0.49823	0.29460	**0.19852**	0.29031	0.21259	0.30269	0.22446	0.37756
TP5	0.23103	0.12853	0.15974	0.14342	**0.11026**	0.14691	0.21601	0.18347
TP6	0.16997	0.07465	0.06967	0.06677	**0.05003**	0.10188	0.16015	0.14962
TP7	0.18578	0.11775	0.19136	**0.11751**	0.12029	0.16698	0.22732	0.16646
TP8	0.21446	0.11151	0.16847	0.10804	**0.10570**	0.16476	0.21221	0.15759
TP9	0.18159	0.11865	0.17570	0.11235	**0.08779**	0.18772	0.25498	0.16197
TP10	0.18243	0.09865	0.18194	**0.08977**	0.10676	0.18599	0.23593	0.14305
TP11	0.14400	0.06942	0.14134	0.07177	**0.06840**	0.17380	0.21139	0.10775
TP12	0.21031	0.10122	0.17433	**0.09517**	0.10161	0.25081	0.29289	0.14313
TP13	0.14360	0.08645	0.23160	0.09670	**0.07410**	0.20503	0.25129	0.13326
TP14	0.11352	**0.04114**	0.17486	0.05092	0.04630	0.17436	0.21153	0.08375
TP15	0.12203	0.05686	0.14929	0.05945	**0.04742**	0.19039	0.22462	0.08918
TP16	0.11987	**0.04253**	0.15299	0.05147	0.04796	0.19016	0.23024	0.08873
Average	0.23813	0.13885	0.18675	0.13796	**0.12531**	0.20465	0.23622	0.19013

complexity of the formulation, traditional exact methods are unable to solve medium- and large-scale instances efficiently; therefore, metaheuristic algorithms were employed.

Eight algorithms, including GA, DE, SA, and their hybrid variants, were implemented. Before conducting the computational experiments, the Taguchi method was used to determine the optimal parameter settings. A set of 16 test instances was designed to evaluate the performance of the algorithms. The results showed that the serial DE–SA algorithm achieved the best objective function values in approximately 50% of the test problems. Its superior performance is attributed to the complementary combination of DE's strong global search capability and SA's effective local refinement, enabling a more balanced exploration and exploitation of the solution space.

In terms of scalability, the proposed metaheuristic approaches significantly reduce computation time

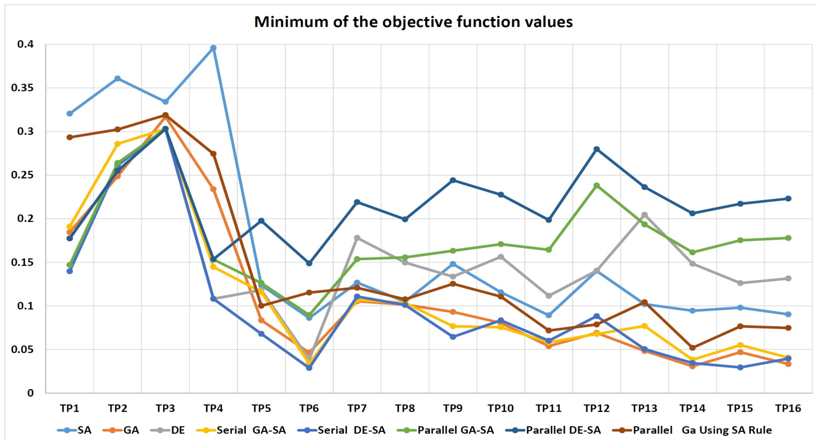

Figure 2: The graph of minimum values

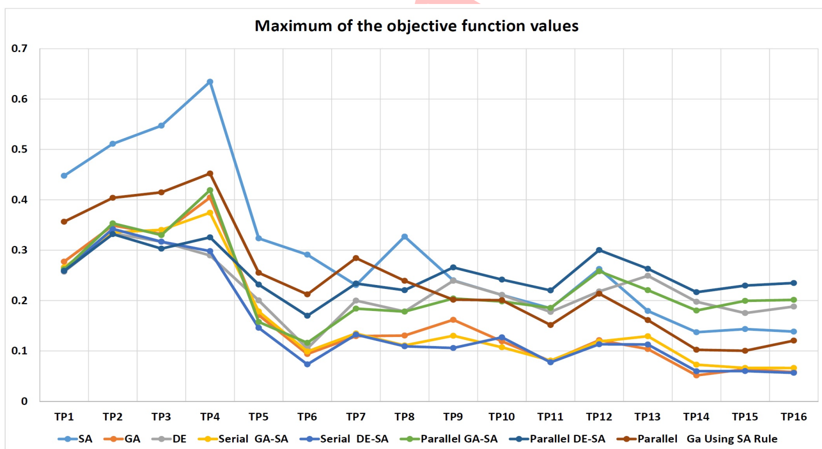

Figure 3: The graph of maximum values

compared to exact methods and can efficiently solve medium-sized instances. However, very large-scale cases remain challenging due to the exponential growth of the solution space. Future research may focus on parallel implementations, improved data-handling techniques, and the incorporation of dynamic and stochastic elements to better reflect real-world operating conditions and enhance practical applicability.

References

[1] M. Abdolhosseinzadeh, M.M. Alipour, *A simulated annealing algorithm for the restricted stochastic traveling salesman problem with exponentially distributed arc lengths*, J. Math. Model. **8(3)** (2020) 279–290.

[2] S. Alumur, B.Y. Kara, *Network hub location problems: The state of the art*, Eur. J. Oper. Res. **190(1)** (2008) 1–21.

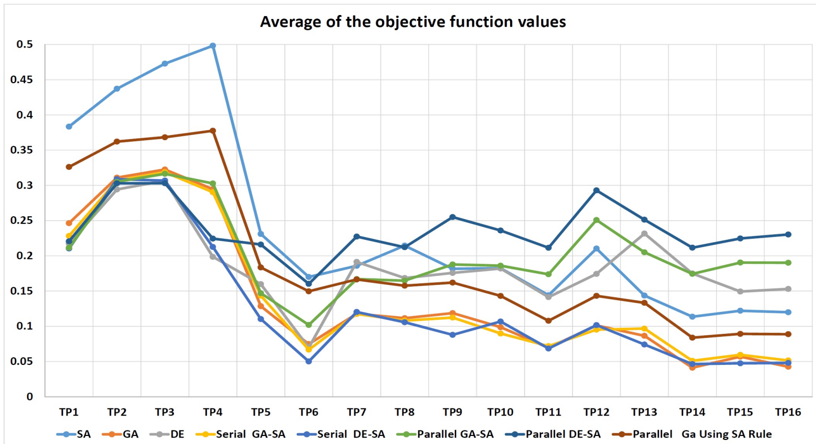

Figure 4: The graph of average values

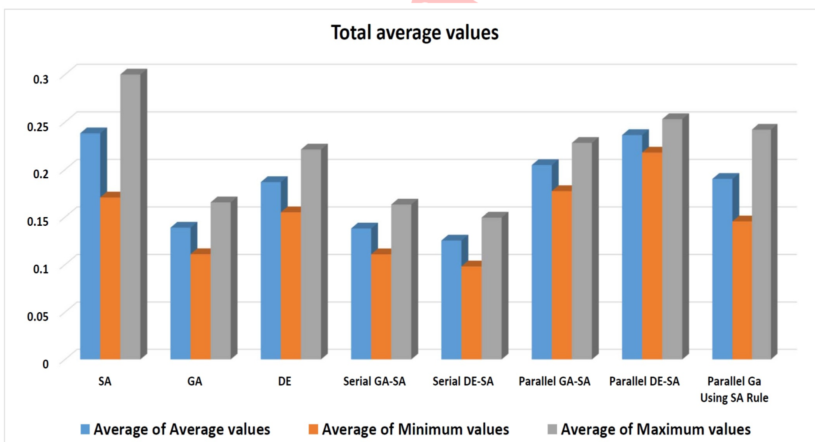

Figure 5: The graph of total average values

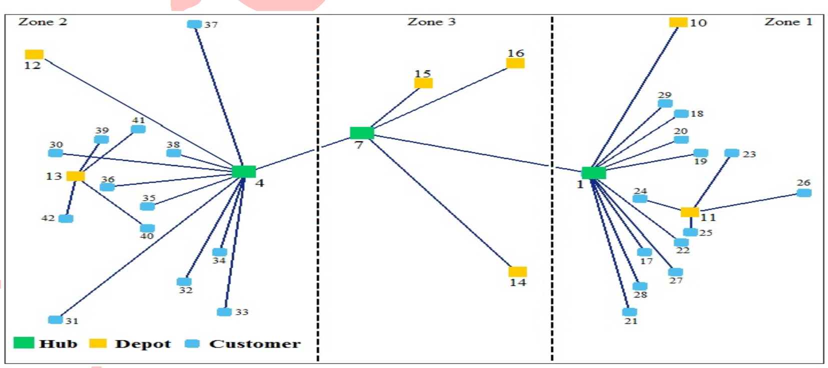

Figure 6: Network design

Table 12: The movement paths of Figure 6 from the depot to the customer

Receiving service using a type 1 drone	Receiving service using a type 2 drone
$10 \rightleftharpoons 1 \rightleftharpoons 17$	
$10 \rightleftharpoons 1 \rightleftharpoons 18$	
	$10 \rightleftharpoons 1 \rightleftharpoons 27$
	$10 \rightleftharpoons 1 \rightleftharpoons 28$
	$10 \rightleftharpoons 1 \rightleftharpoons 29$
$11 \rightleftharpoons 23$	
$11 \rightleftharpoons 24$	
	$11 \rightleftharpoons 25$
	$11 \rightleftharpoons 26$
$12 \rightleftharpoons 4 \rightleftharpoons 33$	
$12 \rightleftharpoons 4 \rightleftharpoons 34$	
	$12 \rightleftharpoons 4 \rightleftharpoons 35$
	$12 \rightleftharpoons 4 \rightleftharpoons 36$
$13 \rightleftharpoons 39$	
$13 \rightleftharpoons 40$	
	$13 \rightleftharpoons 41$
	$13 \rightleftharpoons 42$
$14 \rightleftharpoons 7 \rightleftharpoons 4 \rightleftharpoons 37$	
$14 \rightleftharpoons 7 \rightleftharpoons 4 \rightleftharpoons 38$	
	$14 \rightleftharpoons 7 \rightleftharpoons 4 \rightleftharpoons 31$
	$14 \rightleftharpoons 7 \rightleftharpoons 4 \rightleftharpoons 32$
$15 \rightleftharpoons 7 \rightleftharpoons 1 \rightleftharpoons 20$	
	$15 \rightleftharpoons 7 \rightleftharpoons 1 \rightleftharpoons 21$
	$15 \rightleftharpoons 7 \rightleftharpoons 1 \rightleftharpoons 22$
	$15 \rightleftharpoons 7 \rightleftharpoons 4 \rightleftharpoons 30$
$16 \rightleftharpoons 7 \rightleftharpoons 1 \rightleftharpoons 19$	

[3] S.A. Alumur, J.F. Campbell, I. Contreras, B.Y. Kara, V. Marianov, M.E. O'Kelly, *Perspectives on modeling hub location problems*, Eur. J. Oper. Res. **291(1)** (2021) 1–17.

[4] J.P. Aurambout, K. Gkoumas, B. Ciuffo, *Last mile delivery by drones: An estimation of viable market potential and access to citizens across European cities*, Eur. Transp. Res. Rev. **11(1)** (2019) 1–21.

[5] G. Baloch, F. Gzara, *Strategic network design for parcel delivery with drones under competition*, Transp. Sci. **54(1)** (2020) 204–228.

[6] O. Bozorg-Haddad, M. Solgi, H.A. Loáiciga, *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization*, John Wiley & Sons, 2017.

[7] L. Budd, S. Ison, *Air Transport Management*, Routledge, 2017.

[8] J.F. Campbell, *Integer programming formulations of discrete hub location problems*, Eur. J. Oper. Res. **72(2)** (1994) 387–405.

[9] D. Chauhan, A. Unnikrishnan, M. Figliozzi, *Maximum coverage capacitated facility location problem with range constrained drones*, Transp. Res. Part C Emerg. Technol. **99** (2019) 1–18.

[10] A.P. Chobar, H. Bigdeli, N. Shamami, M. Abolghasemian, *Optimizing hub location for military equipment: A robust mathematical model for uncertainty and meta-heuristic approaches*, Fuzzy Optim. Model. J. **5(4)** (2024) 44–59.

[11] S. Chowdhury, A. Emelogu, M. Marufuzzaman, S.G. Nurre, L. Bian, *Drones for disaster response and relief operations: A continuous approximation model*, Int. J. Prod. Econ. **188** (2017) 167–184.

[12] B.N. Coelho, V.N. Coelho, I.M. Coelho, L.S. Ochi, R. Haghnazar, D. Zuidema, M.S.F. Lima, A.R. da Costa, *A multi-objective green UAV routing problem*, Comput. Oper. Res. **88** (2017) 306–315.

[13] I. Contreras, M. O'Kelly, *Hub location problems*, Loc. Sci. (2019) 327–363.

[14] K. Dorling, J. Heinrichs, G.G. Messier, S. Magierowski, *Vehicle routing problems for drone delivery*, IEEE Trans. Syst. Man Cybern. Syst. **47(1)** (2016) 70–85.

[15] A.T. Ernst, M. Krishnamoorthy, *Efficient algorithms for the uncapacitated single allocation p-hub median problem*, Loc. Sci. **4(3)** (1996) 139–154.

[16] A.T. Ernst, M. Krishnamoorthy, *Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem*, Eur. J. Oper. Res. **104(1)** (1998) 100–112.

[17] F. Farajzadeh, A. Moadab, O.F. Valilai, M. Houshmand, *A novel mathematical model for a cloud-based drone enabled vehicle routing problem considering multi-echelon supply chain*, IFAC-PapersOnLine **53(2)** (2020) 15035–15040.

[18] H. Garg, *A hybrid GSA-GA algorithm for constrained optimization problems*, Inf. Sci. **478** (2019) 499–523.

[19] S. Gelareh, S. Nickel, *Hub location problems in transportation networks*, Transp. Res. Part E Logist. Transp. Rev. **47(6)** (2011) 1092–1111.

[20] M. Golabi, S.M. Shavarani, G. Izbirak, *An edge-based stochastic facility location problem in UAV-supported humanitarian relief logistics: a case study of Tehran earthquake*, Nat. Hazards **87** (2017) 1545–1565.

[21] Q.M. Ha, Y. Deville, Q.D. Pham, M.H. Hà, *On the min-cost traveling salesman problem with drone*, Transp. Res. Part C Emerg. Technol. **86** (2018) 597–621.

[22] M. Hajiaghaei-Keshteli, G. Rahmanifar, M. Mohammadi, F. Gholian-Jouybari, J.J. Klemeš, S. Zahmatkesh, A. Bokhari, G. Fusco, C. Colombaroni, *Designing a multi-period dynamic electric vehicle production-routing problem in a supply chain considering energy consumption*, J. Clean. Prod. **421** (2023) 138471.

[23] J.H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, Michigan Press, 1975.

[24] H. Huang, A.V. Savkin, C. Huang, *A new parcel delivery system with drones and a public train*, J. Intell. Robot. Syst. **100** (2020a) 1341–1354.

[25] H. Huang, A.V. Savkin, C. Huang, *Round trip routing for energy-efficient drone delivery based on a public transportation network*, IEEE Trans. Transp. Electrific. **6(3)** (2020b) 1368–1376.

[26] M. Imanparast, V. Kiani, *A practical heuristic for maximum coverage in large-scale continuous location problem*, J. Math. Model. **9(4)** (2021) 555–572.

[27] B.Y. Kara, *Modeling and analysis of issues in hub location problem*, Ph.D. Thesis, Bilkent University, Turkey, 1999.

[28] A. Khaleghi, A. Eydi, *Hybrid solution methods for a continuous-time multi-period hub location problem with time-dependent demand and sustainability considerations*, J. Ambient Intell. Humaniz. Comput. **15(1)** (2024) 115–155.

[29] M. Khalilzadeh, M. Ahmadi, O. Kebriyaii, *A bi-objective mathematical programming model for a maximal covering hub location problem under uncertainty*, SAGE Open **15(1)** (2025) 21582440251324335.

[30] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, *Optimization by simulated annealing*, Sci. **220(4598)** (1983) 671–680.

[31] G. Laporte, S. Nickel, F. Saldanha-da-Gama, *Introduction to Location Science*, Springer, 2019.

[32] Y. Liu, *An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones*, Comput. Oper. Res. **111** (2019) 1–20.

[33] J.E. Macias, P. Angeloudis, W. Ochieng, *Optimal hub selection for rapid medical deliveries using unmanned aerial vehicles*, Transp. Res. Part C Emerg. Technol. **110** (2020) 56–80.

[34] G. Macrina, L.D.P. Pugliese, F. Guerriero, G. Laporte, *Drone-aided routing: A literature review*, Transp. Res. Part C Emerg. Technol. **120** (2020) 102762.

[35] A.I. Mahmutogullari, B.Y. Kara, *Hub location under competition*, Eur. J. Oper. Res. **250(1)** (2016) 214–225.

[36] T. Meyer, A.T. Ernst, M. Krishnamoorthy, *A 2-phase algorithm for solving the single allocation p-hub center problem*, Comput. Oper. Res. **36(12)** (2009) 3143–3151.

[37] J. Minas, L. Mitten, *The hub operation scheduling problem*, Oper. Res. **6(3)** (1958) 329–345.

[38] C.C. Murray, A.G. Chu, *The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery*, Transp. Res. Part C Emerg. Technol. **54** (2015) 86–109.

[39] M.E. O'Kelly, *A quadratic integer program for the location of interacting hub facilities*, Eur. J. Oper. Res. **32(3)** (1987) 393–404.

[40] H. Omagari, S.I. Higashino, *Provisional-ideal-point-based multi-objective optimization method for drone delivery problem*, Int. J. Aeronaut. Space Sci. **19** (2018) 262–277.

[41] C. Ortiz-Astorquiza, I. Contreras, G. Laporte, *Multi-level facility location as the maximization of a submodular set function*, Eur. J. Oper. Res. **247(3)** (2015) 1013–1016.

[42] D. Pamucar, D. Lazarević, M. Dobrodolac, V. Simic, Ö.F. Görçün, *Prioritization of crowdsourcing models for last-mile delivery using fuzzy Sugeno–Weber framework*, Eng. Appl. Artif. Intell. **128** (2024) 107414.

[43] S. Poikonen, X. Wang, B. Golden, *The vehicle routing problem with drones: Extended models and connections*, Netw. **70(1)** (2017) 34–43.

[44] I. Rodríguez-Martín, J.J. Salazar-González, H. Yaman, *A branch-and-cut algorithm for the hub location and routing problem*, Comput. Oper. Res. **50** (2014) 161–174.

[45] B. Rostami, N. Kämmerling, J. Naoum-Sawaya, C. Buchheim, U. Clausen, *Stochastic single-allocation hub location*, Eur. J. Oper. Res. **289(3)** (2021) 1087–1106.

[46] M.R. Salama, S. Srinivas, *Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites*, Transp. Res. Part E Logist. Transp. Rev. **164** (2022) 102788.

[47] E. Shadkam, *Parameter setting of meta-heuristic algorithms: a new hybrid method based on DEA and RSM*, Environ. Sci. Pollut. Res. **29(15)** (2022) 22404–22426.

[48] E. Shadkam, S. Safari, S.S. Abdollahzadeh, *Finally, which meta-heuristic algorithm is the best one?*, Int. J. Decis. Sci. Risk Manag. **10(1-2)** (2021) 32–50.

[49] D. Skorin-Kapov, J. Skorin-Kapov, M. O'Kelly, *Tight linear programming relaxations of uncapacitated p-hub median problems*, Eur. J. Oper. Res. **94(3)** (1996) 582–593.

[50] B. Soylu, H. Katip, *A multiobjective hub-airport location problem for an airline network design*, Eur. J. Oper. Res. **277(2)** (2019) 412–425.

[51] R. Storn, K. Price, *Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces*, J. Glob. Optim. **11(4)** (1997) 341–359.

[52] G. Taguchi, *Introduction to Quality Engineering: Designing Quality into Products and Processes*, Asian Productivity Organization, Tokyo, 1986.

[53] Y. Tanabe, Y. Kitagawa, *An adaptive selection system of base-isolation devices evaluated by using a soft computing method that considers seismic performance*, 13th World Conf. Earthquake Eng., Vancouver, B.C., Canada, 2004.

[54] M. Tavana, K. Khalili-Damghani, F.J. Santos-Arteaga, M.H. Zandi, *Drone shipping versus truck delivery in a cross-docking system with multiple fleets and products*, Expert Syst. Appl. **72** (2017) 93–107.

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

[56] B. Urazel, Y. B. Sahin, *Solving a cubic cell formation problem with quality index using a hybrid meta-heuristic approach*, Gazi Univ. J. Sci. **36(2)** (2023) 752–771.

[57] X. Wang, S. Poikonen, B. Golden, *The vehicle routing problem with drones: several worst-case results*, Optim. Lett. **11** (2017) 679–697.

[58] Z. Wang, J. B. Sheu, *Vehicle routing problem with drones*, Transp. Res. Part B Methodol. **122** (2019) 350–364.

[59] G. Wu, N. Mao, Q. Luo, B. Xu, J. Shi, P N. Suganthan, *Collaborative truck-drone routing for contactless parcel delivery during the epidemic*, IEEE Trans. Intell. Transp. Syst. **23(12)** (2022) 25077–25091.

[60] Y. Yadav, A. Narasimhamurthy, *Algorithms for solving the vehicle routing problem with drones*, Proc. 2017 Ninth Int. Conf. Adv. Pattern Recognit. (ICAPR), 2017.

[61] M. Ziaee, M. Imanparast, V. Khodabakhshi, *Multi-agent single machine scheduling problem with transportation constraints*, J. Math. Model. **10(3)** (2022) 367–385.