Scientia Iranica D (2022) 29(2), 632–644

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions D: Computer Science & Engineering and Electrical Engineering
http://scientiairanica.sharif.edu

Ground vehicle and UAV collaborative routing and scheduling for humanitarian logistics using random walk based ant colony optimization

S. Bansal[a], R. Goel[b,*], and R. Maini[c]

a. *Computer Science and Engineering Department, Maharishi Markandeshwar (Deemed) University, Mullana, India.*
b. *Computer Science Department, Government College, Naraingarh, Ambala, India.*
c. *Computer Science Engineering Department, Punjabi University, Patiala, India.*

Received 12 May 2021; received in revised form 20 August 2021; accepted 1 November 2021

KEYWORDS
Humanitarian logistics;
UAV;
Truck-drone delivery;
Ant colony optimization.

Abstract. A well-planned humanitarian logistics aiming to rescue people and provide on-time lifesaving facilities in disaster-affected areas can significantly mitigate the repercussions of disasters. However, damaged bridges and blocked roads can hinder last-mile deliveries in disaster-affected areas to ground vehicles only. In this regard, the present study attempts to propose Ground Vehicle (GV) and Unmanned Air Vehicle (UAV) collaborative delivery system to be implemented in such areas. To this end, a fleet of homogenous ground vehicles, each equipped with a certain number of UAVs, was deployed for last-mile deliveries. UAVs make the flight from GVs, deliver to the end locations, and return to the GV for battery replacement and/or start another flight. The main objective of the proposed model is to minimize the total delivery time within UAV flight endurance and payload constraints. First, K-means algorithm was used to cluster the disaster-affected region into different sectors. Then, GV-touring and UAV-routing were scheduled using the nearest neighbor heuristics to serve the ground approachable locations and UAV served locations, respectively. Finally, the levy flight-based Ant Colony Optimization-based (ACS_RW) algorithm was developed to further optimize the overall travel time. Experimentation results show the potential superiority of the proposed algorithm over other available truck-drone collaborative transportation models.

1. Introduction

Large-scale disasters have major impacts on our lives that cause damages and loss to human and animal lives, buildings, materials, etc. In recent years, several natural and man-made disasters have hit different parts of the world, caused significant losses to economic growth, and claimed a number of human lives. For instance, Tsunami (Indian Ocean, 2005), Hurricane Katrina (hitting coastal areas in the US, 2005), Haiti earthquake (2011), flooding (India, 2013), and forest wildfire (Australia, 2019) were some of the natural disasters that shocked the world in the past decades. Casualties as well as loss of homes and communities are the direct and immediate aftermaths of such natural disasters. Furthermore, people living in the disaster-affected areas also suffer shortage of food, fresh/clean water, medicines, and other basic necessities. In the absence of proper planning, governments and NGOs

*. *Corresponding author. Tel.: +91-9896024216*
E-mail addresses: sandhya12bansal@gmail.com (S. Bansala); rajeev2040.he@hry.gov.in (R. Goel); research_raman@yahoo.com (R. Maini)

doi: 10.24200/sci.2021.58309.5664

are often unable to meet the demands of disaster-affected areas and cater for the needs of people in such areas. Generally, it takes days for response teams to reach disaster-affected areas. Transportation activities initiated in response to natural/ manmade disasters are known as humanitarian logistics.

Currently, Unmanned Air Vehicles (UAVs generally called drones) are frequently used to supply the ordered products to end-users. Given that drones are not restricted by road transportation, they are widely used for timely deliveries. By enjoying this potential, drones can play an important role in humanitarian logistics, as well. However, the major challenge in the application of UAVs as a means of transportation is their low power source which limits their flight range to comparatively shorter distances. The other disadvantages are their small time-intervals and limiting payloads.

Since a well-planned routing and scheduling of truck-drone cooperative delivery can significantly reduce the delivery timing and total transportation cost, researchers have turned their attention to such issues during the last five years. The majority of the existing works in this field have focused on the commercial package delivery using a combined fleet of drones and trucks, where some of the customers are served by drones while the remaining ones are served by trucks. In [1–3], joint routing was considered using truck-assisted UAVs. However, only single UAVs were used for last-mile deliveries. As a result, they are not very efficient in covering multiple locations simultaneously. Moreover, they may not be able to deliver items promptly which can be a critical issue in humanitarian logistics. Keeping this in mind, we proposed a multi-truck-multi-UAV last-mile delivery of emergency supplies in disaster-affected areas in this study which could be expected to significantly reduce the transportation time. To this end, the humanitarian logistics was first modeled as Multi-Vehicle-UAV collaborative Humanitarian Logistics (MVUHL) and then, a meta-heuristic was developed to solve that model efficiently and effectively. In this study, first, the endpoints (disaster-affected locations) in different sectors were clustered and then, at least, one location in each sector that could be served using the ground vehicle, i.e., truck, was identified. This specific location is called "anchor point" that is used to launch the drone(s) to serve the customers of that cluster, retrieve them back on the truck, and recharge the batteries after delivering the items. Second, a routing plan was scheduled to visit all these anchor points using an available fleet of trucks. Third, a routing plan of UAVs was prepared from each anchor point for last-mile deliveries. Fourth, the above-mentioned two routing plans were optimized to meet the temporal requirements of trucks and drones.

The methods for solving the delivery problems are broadly classified into two categories:

(i) Exact algorithms;

(ii) Approximate algorithms [4–6].

Exact algorithms such as branch and cut, mixed-integer linear programming, dynamic programming, etc. can solve small-sized instances within a reasonable time, while the ideal solution to large-sized delivery problems requires a significant and expensive amount of computation time. The inability of exact algorithms to solve large-sized problems has motivated researchers to develop approximate methods that can solve such NP-Hard problems effectively and efficiently in a reasonable time [5,7–9]. Approximate methods namely the genetic algorithms, particle swarm optimization, simulated annealing, and swarm intelligence-based Ant Colony Optimization (ACO) algorithms are widely used in solving delivery problems [4–6,8,10–13]. However, many of these methods including GA, PSO, SA, Tabu search, etc. suffer premature convergence due to local optima and lack of population diversity [4,7–10]. On the other hand, ACO-based algorithms are found more successful in solving complex delivery problems owing to their simple population generation and balance between the exploration and exploitation capabilities. Moreover, asynchronous agent cooperation at the colony level gives ACO algorithms their distinct ever [5,12,14–16].

To the best of the authors' knowledge, this study is the first of its kind that considers multiple ground vehicles and UAVs simultaneously for transportation as well as their applications to last-mile emergency supply deliveries. The rest of this paper is organized as follows. Section 2 discusses other existing studies on the application of UAVs in humanitarian logistics. Section 3 proposes the model considering the application of GV-UAV for disaster relief transportation. Section 4 elaborates on the proposed approach. Section 5 analyzes the results obtained from the proposed algorithm on certain datasets. Finally, Section 6 ends the paper with the concluding remarks and suggests the future direction.

2. UAVs in humanitarian logistics

The first 72 hours after the disasters are of critical importance for preserving human life; therefore, Search And Rescue (SAR) operations must be scheduled effectively and efficiently. In this context, the American Red Cross report supports the application of UAVs as a powerful and robust tool for SAR operations [11]. This report explored the significant role of drones in disaster planning, preparation, and response stages based on different policy recommendations, use cases, platforms, and different payloads. Use of emergent technologies in developing an automated emergency response system was elaborated in [12]. A Cyber-Physical System (CPS) was also developed where heterogeneous vehicle fleets (fixed-wing aircraft, ground vehicles, and drones) were combined, controlled, and coordinated via cyberspace to address complex humanitarian disaster response operations.

Murray and Chu [3] modeled the disaster relief transportation as:

(i) Flying Sidekick Traveling Salesman Problem (FSTSP);

(ii) Parallel Drone Scheduling TSP (PDSTSP).

In PDSTSP, drones can fly from the depot to deliver parcels to customers and return to the depot. However, in case the delivery locations are not in the range of drones from the central depot, both drone and truck work cooperatively for last-mile deliveries as FSTSP. Ferrandez et al. proposed a truck-drone system in tandem delivery networks in [2]. They employed K-means algorithm to identify the launching points of drones from trucks and further used a genetic algorithm to route the trucks among these launching points. They also discussed the benefits of using multiple drones despite the fact that drones are not limited by flight endurance. Chowdhury, et al. [13] modeled the disaster relief operations as a continuous approximation problem. They aimed to minimize the overall distribution cost (trucks and drones) through optimal selection of the locations of the distribution centers, optimal assignment of the serving locations, and efficient prediction of the ordering quantities.

Luo et al. [1] modeled truck-drone delivery as a two-echelon cooperated routing problem where Ground Vehicles (GV) travel on road networks and UAVs serve customers far beyond the roads and launch and land on GVs. Rabta et al. developed a new mathematical model to simulate last-mile deliveries in post-disaster scenarios in [14]. The model was constructed considering UAV payload, flight endurance, and disaster-specific circumstances as the function values. The numerical examples showed that prioritization of customers and optimal assignment of the recharging stations could significantly extend the operational distance of the drones. Researchers in [15] proposed a procedure to assist decision-makers in setting up disaster aid distribution networks using UAVs and Geographical Information Systems (GIS). The overall process was divided into five different stages. Chauhan, et al. presented an Integer Linear Programming (ILP) model to identify the facility locations and assign drones for last-mile deliveries in post-disaster scenarios [16]. Three different solution methodologies namely MIP solver, greedy approach, and three-Stage Heuristic (3SH) were taken into account and compared with each other. According to their findings, 3SH can achieve balanced performance on high coverage areas by economically deploying drones. Kitjacharoenchai et al. [17] formulated MTSP-UAV routing as Mixed Integer Programming (MIP) to minimize the delivery time of both trucks and UAVs. Recently, Murray and Raj [18] formulated the last-mile parcel deliveries as a multiple FSTSP (mFSTSP) problem using a single truck and multiple UAVs. They also proposed a three-phased heuristic to solve the problem. Analysis of the results revealed that additional drones would yield diminishing marginal make-span improvements and large flight endurance UAVs would be more beneficial when used to serve customers in larger areas. Liu et al. [19] proposed a new two-echelon routing using truck-drone cooperative scheduling. Their proposed model considered energy consumption as a function of payload weight. Furthermore, simulated annealing with a Tabu search-based approach was proposed to enhance the quality of the obtained results.

Although considerable research has been done on the application of drones in disaster relief systems, use of multi-UAVs and Multi-GVs in this area has not been properly explored yet. In this regard, this study explored the cooperative application of multiple GVs and multiple UAVs in humanitarian logistics.

3. Problem formulation

The proposed problem consists of a set of target locations situated in disaster-affected areas, each of which is to be served with a kit of daily need items, as shown in Figure 1a. Given that no-land deliveries are possible for all locations due to blocked/damaged roads, a fleet of GVs, each with a fixed number of UAVs, was employed for last-mile deliveries. The locations which are directly approachable by land roads are referred to as *Direct_Points* and those for which no land deliveries are possible are regarded as *Remote_Locations*, as shown in Figure 1b. While all *Remote_Locations* can be served by UAVs only,

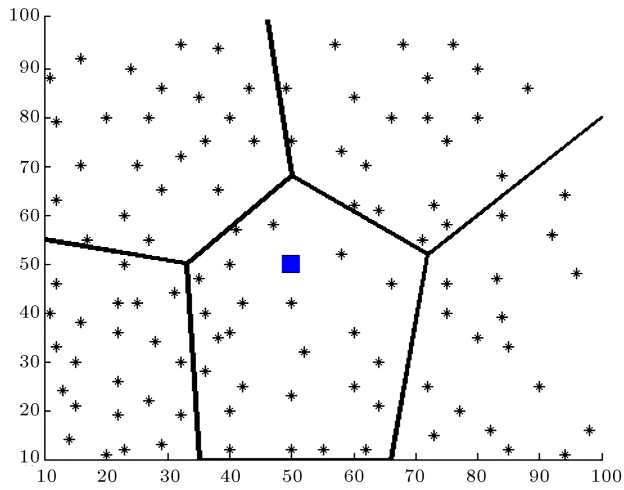

Figure 1a. Cluster sectors.

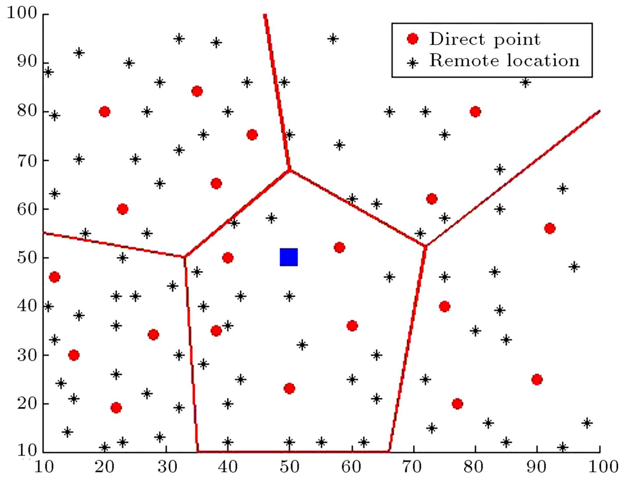

Figure 1b. Identification of DPs and RLs.

Direct_Points are served by GVs as well as UAVs.

Each location receives a packet (containing food, water, medical kit, and hygiene items) weighing approximately q units, and each UAV once fully charged can fly up to $\mathbb{D}_{\max}(= 7$ km) with a payload of $\mathbb{Q}$ units. The flight endurance limit of a fully charged UAV is $\mathbb{T}_{\max}(= 30$ minutes). However, without loss of generality, it is assumed that GVs (usually trucks) are not restricted by any capacity, distance, and time constraints. For simplicity, other assumptions are summarized below:

- Coordinates of all locations are known in advance;
- Both GV and UAV travel at constant speeds;
- Speed of UAV (Vel_U) is 1.25 times that of GV, i.e., $Vel_U = 1.25^* Vel_V$);
- UAVs are capable of automatic launch, fly, package delivery, and rendezvous back to GV;
- The required time to deliver packages and replace batteries is negligible.

Let $G(\mathbb{V}, E)$ be the set of vertices ($\mathbb{V}$) and set of edges (E), respectively. $\mathbb{V}$ is divided into two subsets of (i) $\mathbb{V}_a = \{a_1, a_2, \cdots, a_m\}$ as a finite set of m Direct_Points (DP) and (ii) $\mathbb{V}_c = \{c_1, c_2, \cdots, c_n\}$ as a finite set of n Remote_Locations (RL), such that:

$$\mathbb{V}_c \cup \mathbb{V}_a = \mathbb{V}, \tag{1}$$

and:

$$\mathbb{V}_c \cap \mathbb{V}_a = \emptyset. \tag{2}$$

Eqs. (1) and (2) imply that all nodes must be classified either as DP or *RL*, and no node should belong to both classes, respectively.

A special node acts as a central depot where GVs start their tour and return to the same depot. However, different notations a_0 and a_{m+1} are employed to represent the starting and ending nodes, respectively. A distance matrix $D = \{d_{ij}\} \forall i, j \in \mathbb{V}$ with $d_{ij} = d_{ji}$ is defined on E which corresponds to the distance between nodes i and j.

Furthermore, a set of GVs denoted by $V = \{V_1, V_2, \cdots V_K\}$ is available at the central depot, each equipped with a finite set of UAVs denoted by $U = \{U_1, U_2, \cdots, U_N\}$. The set of routes traveled by GVs to cover DPs is denoted by $r = \{r_1, r_2, \cdots r_K\}$ such that $r_i = \{a_0, a_1, \cdots a_k, \cdots a_{m+1}\}$, $a_i \in \mathbb{V}_a$. Similarly, the set of routes to cover all the *RL*s allocated to a particular *DP* (a_i) is denoted by $\Re = \{\Re_i^1, \Re_i^2, \cdots, \Re_i^K\}$ where $\Re_i^j = \{a_i, c_1, \cdots c_k, \cdots a_{i'}\}$, $c_i \in \mathbb{V}_c$. Let $x_{ij}^k = \{0, 1\}$ be a binary variable with the value of 1 if the kth ground vehicle visits the *DP* a_j after serving a_i; otherwise, it takes the value of 0. Similarly, $y_{ij}^k = \{0, 1\}$ is another binary variable with the value of 1 if the kth UAV visits node j after serving i $(i, j \in \mathbb{V})$; otherwise, it takes the value of 0. Let the binary variable $\psi(i, j, k)$ check whether or not the arc $(i - j)$ $(i, j \in \mathbb{V})$ belonging to the route started from the *anchor_point* (a_j). Here, *anchor_point* is a special DP from which UAVs start their route to cover all the RLs associated with those DPs. In addition, $\Psi(i, j, k, l)$ is used as a binary variable representing whether or not lth UAV of kth GV was launched from the anchor point (a_i) and recollected at the anchor point (a_j). Let the binary variable $\psi(i, j, k)$ check whether or not the edge $(i - j)$ belonging to the route started from the anchor point (a_j).

The arrival time of the kth GV at the jth DP is calculated as the sum of departure time from the ith DP and travel time between the ith and jth DP provided that the kth GV travels directly to the jth DP from the ith DP.

$$AT_j^k(GV) = [DT_i^k(GV) + d_{ij}/Vel_V]^* x_{ij}^k$$
$$\forall\, i, j, \quad i \neq j, \in \mathbb{V}_a. \tag{3}$$

The launch time (or departure time) of the lth UAV (associated with the kth GV) at the jth DP is equal to the maximum of the arrival time of GV or UAV at that DP. In other words:

$$LT_j^{kl}(UAV) = \max\left(AT_j^k(GV), AT_j^{kl}(UAV)\right). \tag{4}$$

The arrival time of the lth UAV (associated with kth GV) at the jth RL is equal to the sum of the departure time from the ith DP and the time taken to travel up to jth DP provided that the kth GV is scheduled to serve the jth DP.

$$AT_j^{kl}(UAV) = \Bigg[\Bigg\{ DT_i^{kl}(UAV)$$

$$\left. + \frac{\sum\limits_{a\in\mathbb{V}}\sum\limits_{b\in\mathbb{V}} {y^l_{ab}}^* \psi(a,b,i)^* d_{ab}}{Vel_U} \right\} * \Psi(i,j,k,l) \Bigg]$$

$$+ [(1-\Psi(i,j,k,l))^* AT_j^k(GV)]$$

$$\forall\, i,j, \quad i \neq j, \in \mathbb{V}_a. \tag{5}$$

Recollection time of the lth UAV (associated with kth GV) at the jth anchor point after serving all the end customers on the route $\Re_j^l$ is calculated as follows:

$$RT_j^{kl}(UAV) = LT_j^{kl}(UAV) + \frac{\sum\limits_a \sum\limits_b {y^l_{ab}}^* \psi(a,b,j)^* d_{ab}}{Vel_U}. \tag{6}$$

The departure time of the kth GV from the jth DP is the maximum recollection time of all the UAVs that are scheduled to be recollected at the jth DP which is given by:

$$DT_j^k(GV) = \max(RT_j^{kl}(UAV)) \qquad \forall\, l \in U. \tag{7}$$

The MIP formulation of the proposed humanitarian logistic is as follows:

$$\min(\max(AT_i)) \qquad \forall\, i \in \mathbb{V}, \tag{8}$$

where AT_i is the arrival time (i.e., the delivery timing) at the ith node calculated as:

$$AT_i = \begin{cases} AT_i^k(GV) & \text{if} \quad i \in \mathbb{V}_a \\ AT_i^{kl}(UAV) & \text{otherwise} \end{cases} \tag{9}$$

s.t.:

$$\sum_{k\in V}\sum_{i\in\mathbb{V}_a} x_{ij}^k \leq |V| \qquad \forall\, j \in \mathbb{V}_a, \quad i \neq j, \tag{10}$$

$$\sum_{j\in\mathbb{V}_a} x_{a_0 j}^k = \sum_{j\in\mathbb{V}_a} x_{j a_{m+1}}^k \leq 1 \qquad \forall\, k \in V, \tag{11}$$

$$\sum_{i\in\mathbb{V}_a\cup a_0}\ \sum_{\substack{j\in\mathbb{V}_a,\\ i\neq j}} x_{ij}^k = \sum_{i\in\mathbb{V}_a}\ \sum_{\substack{j\in\mathbb{V}_a\cup a_{m+1},\\ i\neq j}} x_{ij}^k \qquad \forall\, k \in V, \tag{12}$$

$$\sum_{\forall\ k\in V}\ \sum_{\substack{j\in\mathbb{V}_a\cup a_{m+1},\\ i\neq j}} \Psi(i,j,k,l) \leq |U_i| \qquad \forall\, i\in\mathbb{V}_a\cup a_0, \tag{13}$$

$$2\Psi(i,j,k,l) = \sum_{\substack{h\in\mathbb{V}_a,\\ i\neq h}} x_{hi}^k + \sum_{\substack{h\in\mathbb{V}_a,\\ j\neq h'}} x_{jh'}^k$$

$$\forall\, k \in, \quad \forall\, l \in U i, j \in \mathbb{V}_a, \quad i \neq j, \tag{14}$$

$$\mathbb{D}(\Re_i^l) = \left(\sum_{a\in\mathbb{V}}\sum_{b\in\mathbb{V}} y_{ab}^l * \psi(a,b,i) * d_{ab}\right)$$

$$* \Psi(i,j,k,l) \leq \mathbb{D}_{\max} \qquad \forall\, j \in \mathbb{V}_a, \quad \forall\, k \in V, \tag{15}$$

$$\mathbb{T}(\Re_j^l) = RT_j^{kl}(UAV) \leq \mathbb{T}_{\max}$$

$$\forall j \in \mathbb{V}_a, \quad \forall\, k \in V, \tag{16}$$

$$\left(\sum_{a\in\mathbb{V}}\sum_{\substack{b\in V\\ b\neq j}} y_{ab}^l * \psi(a,b,i) * q\right) * \Psi(i,j,k,l) \leq \mathbb{Q}$$

$$\forall\, j \in \mathbb{V}_a, \quad \forall\, k \in V, \tag{17}$$

$$\left(\sum_{a\in\mathbb{V}}\ \sum_{\substack{b\in V\\ b\neq j}} y_{ab}^l * \psi(a,b,i) * q\right) * \Psi(i,j,k,l) \leq \mathbb{Q}$$

$$\forall j \in \mathbb{V}_a, \quad \forall k \in V, \tag{18}$$

$$x_{ij}^k = \{0,1\} \qquad \forall\, i,j \in \mathbb{V}_a k \in V, \tag{19}$$

$$y_{ij}^k = \{0,1\} \qquad \forall\, i,j \in \mathbb{V}_c k \in U, \tag{20}$$

$$\Psi(i,j,k,l) = \{0,1\} \qquad \forall\, i,j \in \mathbb{V}_c k \in V, \quad l \in U, \tag{21}$$

$$\psi(i,j,k) = \{0,1\} \qquad \forall\, i,j \in \mathbb{V}_c k \in U. \tag{22}$$

Objective function (9) minimizes the delivery timing of all locations Inequ. (10) restricts the maximum number of the employed GVs, and Eqs. (11) and (12) ensure that a GV that starts from the central depot must return to it. Eq. (12) guarantees that in case a GV visits a DP, it must leave for the next node except in the case of the last node, i.e., the central depot. Eq. (13) ensures the limitation on the number of UAVs launching from any GV at any anchor node. Eq. (14) ensures that in case a UAV launching from the ith anchor node is recollected at the jth anchor node, the GV must visit the jth anchor node after visiting the ith anchor node. Eqs. (15), (16), and (17) guarantee the flight endurance, total flight time, and capacity constraints of UAVs, respectively. Constraints (18)–(21) determine the types and ranges of the variables.

4. Proposed MVUHL approach

The above-mentioned problem is more complex than other common NP-hard VRP problems since it consists of two-level VRP scheduling, i.e., cooperative routing using GVs and UAVs. Therefore, it can be concluded that exact mathematical methods such as Branch & Bound, column generation, etc. are capable of solving only small-size problems (up to 10 nodes) in a reasonable time. The computational efforts and time increase exponentially as the problem size increases to the medium (50 nodes) or large (100 or more nodes) sizes. For such types of middle- and large-size problems, we proposed an Ant Colony System (ACS) optimization-based metaheuristic to solve the proposed model.

The overall routing and scheduling process is divided into three phases:

(i) Phase-I: Initial solution construction;

(ii) Phase-II: Optimization using the proposed ACS_RW approach;

(iii) Phase-III: Post processing.

At the first phase, GV tours and UAVs routes are initialized and then, optimized at the second phase using the proposed metaheuristic. The obtained solutions (delivery timings) are further optimized at the third phase by implementing a route optimization strategy.

4.1. Initial solution construction

Since last-mile delivery in disaster-affected regions is a very challenging task, pre-processing of test data such as clustering delivery locations is very beneficial that can boost the convergence capabilities of the routing algorithm [13,20,21]. In this regard, in the case of the proposed approach, the nearby locations were first clustered into different sectors.

Here, an initial GV-UAV cooperative routing was scheduled based on K-means clustering and nearest-neighbor policy, as suggested by Abbatecola et al. [20], Nalepa and Blocho [21], and Arparslan and Science [22]. Generally, the very first step in humanitarian logistics is to divide the overall disaster-affected area into different clusters (called sectors) so that different teams can be hired to service different sectors [13]. In the proposed approach, the affected region is divided into $|V|$ sectors (Algorithm 1) and one GV should be scheduled for each sector. Next, GV_Tours for each sector are scheduled (Algorithm 2) to cover the locations that are directly accessible by ground vehicles. Such locations are referred to as Direct_Points in this study. Finally, UAV_Routes are scheduled for the rest of the locations (termed as Remote_Points). Of note, this phase is further subdivided into three sub-modules:

(a) Cluster_Sectors;

(b) GV_Touring;

(c) UAV_Routing.

```
Input: Direct_Points in the given sector
Output: GV_Tour to visit its Dirct_Points
Set Current_Node=Depot
Visited {Depot} = True
For i = 1|Direct_Points| in given sector
        Set Visited {i} = False
End for
Repeat
        Find Location (k) which is nearest to the current_Node.
        Set Visited {k} = True
        Set Current_Node = k
While (Visited {i} ≠ True, ∀ i)
```

Algorithm 2. GV_Touring.

Cluster_Sectors: The present study employed K-means clustering (Algorithm 1) to divide the region into $|V|$ sectors, as depicted in Figure 1a. Initially, $|V|$ centroids are randomly assigned in the disaster-affected region. Each location is assigned to its nearest centroid depending on the Euclidian distance from the centroid. Each sector/cluster centroid is updated depending on the locations assigned to that cluster. The process of assigning locations and updating centroids is repeated until centroids remain the same or no location-cluster change.

GV_Touring: After assigning all locations to different sectors, GV_Tours are generated for all the Direct_Points (DP) in each sector, as shown in Figure 1c. All GV_Tours start from and end at the central depot. In addition, *nearest-neighbor heuristics* (Algorithm 2) was used to construct such tours where a GV repeatedly visited its nearest location until visiting all the DPs assigned to it.

UAV_Routing: Here, the Remote_Locations (RL) are firstly assigned to DPs using the abovementioned K-means clustering, considering *Direct_Locations* as cluster centroids. Then, the UAV_Routes (in different clusters) are initialized using the mentioned nearest-neighbor heuristics keeping centroid (i.e., DP) as the starting and ending node, as depicted in Figure 1d.

```
Input: Geographical locations to be visited
Output: Locations clustered into sectors
Randomly generate |V| centroids in the given region.
Repeat
        Assign locations to the nearest random points based on Euclidian distances.
        Update positions of centroids in all clusters.
While (No change in the centroid position or location sector)
```

Algorithm 1. Cluster_Sectors.

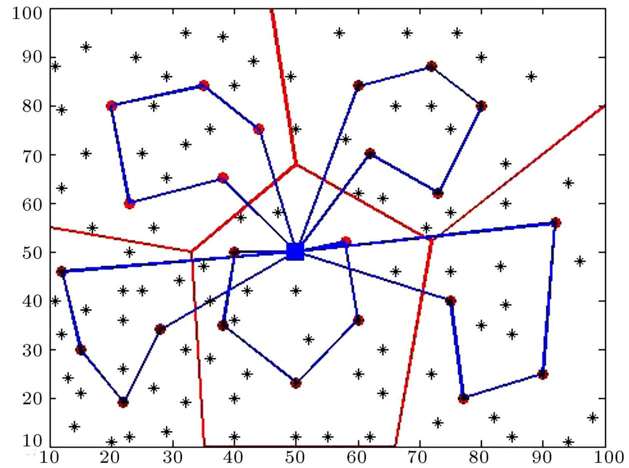

Figure 1c. GV touring.

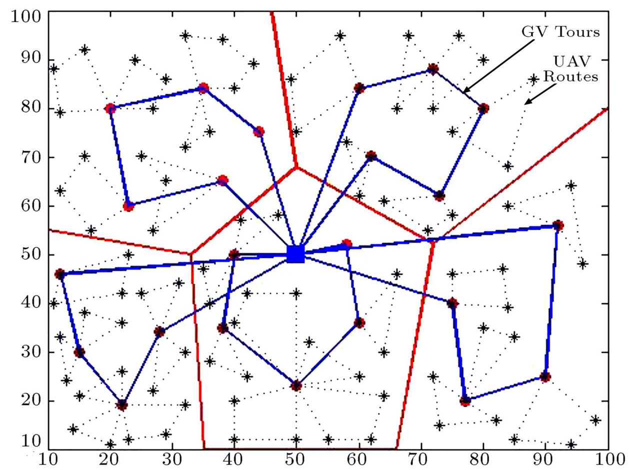

Figure 1d. UAV routing.

4.2. Optimization using RW_ACS

The initialized UAV routes were improved using the ACS-based optimization process. In addition, to visit Aps, GV (or Truck) touring was done using the same algorithm. ACS is a nature-inspired metaheuristic algorithm inspired by ants in search of food [23]. It uses exploration and exploitation strategies to solve combinatorial optimization problems. Initially, the ants start their search randomly. They lay a natural chemical (called pheromone) on the paths they follow. The pheromone also gets evaporated over time. To be specific, the more ants following the path, the greater the pheromone concentration on the path. The pheromone deposited by the ants on different paths guides other ants to identify the path for the food source. Hence, the leading ants will *exploit* the ants to search for food source. However, the ants can also use their own strategy for choosing the paths using the solution quality of the chosen path that assist them in exploring the new search space. The ants determine the next path selection probabilistically based on both *exploration* and *exploitation* strategies. In this respect, the ACS-based algorithms were successfully applied to different optimization problems including the routing problems [9,10,23–28]. Despite its capability to solve a wide variety of combinatorial optimization problems, ACS may be trapped into local optima. As a result, it converges prematurely due to stagnation in local optima. In this regard, the current study employed ACS with *Random Walk* along with ACS metaheuristic to avoid the local optima stagnation problem.

Random Walk: Random walk is a random process where the next consecutive steps are randomly selected. Mathematically, we have:

$$W_n = \sum_{i=1}^{n} s_i, \tag{23}$$

where W_n is the nth random walk, and s_i the random step chosen among a random distribution. Dependence of the nth random walk on its previous $(n-1)$th random walk is illustrated by:

$$W_n = \sum_{i=1}^{n} s_i = \sum_{i=1}^{n-1} s_i + s_n = W_{(n-1)} + s_n, \tag{24}$$

indicating that the next state (i.e. W_n) depends only on the (i) current state $W_{(n-1)}$ and (ii) current step size s_n. Here, a series of random walks to reach for a final position x_n after starting from an initial position x_0 is defined as follows:

$$x_n = x_0 + \alpha_1 s_1 + \alpha_2 s_2 + \alpha_3 s_3 + \alpha_n s_n = x_0, \tag{25}$$

where $\alpha_i > 0$ is a parameter that controls s_i. In the present scenario, Cauchy distribution was taken into account to choose the step size since this distribution had infinite variance which could help take higher jumps to come out from the local minima.

Now, the Random walk-based ACS approach can be given by the RW_ACS algorithm (Algorithm 3).

4.3. Post processing

This phase is used for further optimization of the obtained solution by either (i) serving a *Direct_Point* using UAV or (ii) updating the UAV collection (i.e., Rendezvous) node explained as *Tour_Optimization* and *Route_Optimization* strategies, respectively.

4.3.1. Tour_Optimization

Given that $Vel_{UAV} > Vel_V$, in case some of the *Direct Points* are served by UAVs instead of GVs, there could be a substantial improvement in the delivery timings. However, to avoid complexity, only those DPs in each sector were selected such that the number of *RLs* allocated to that DP was less than half of the average number of *RLs* allocated to that DP. In the case of finding such a DP, all *RLs* of that DP along with such

Step 1: (a) Place "m" ants at the starting location of the route,
(b) Initialize the pheromone of each path joining nodes a and b as:

$$\tau_{ab} = 1/d_{ab}. \tag{25}$$

Step 2: Repeat Steps 3–7 for the required number of steps;
Step 3: Find the probability of choosing all the approachable nodes (J) from the current node i based on the following rule:

$$p_{ij} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{j\in J}\tau_{ij}^{\alpha}\eta_{ij}^{\beta}} & j \in J \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

Step 4: Select the next node to be visited among these approachable nodes using the following transition formula:

$$j = \begin{cases} \arg\max \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{j\in J}\tau_{ij}^{\alpha}\eta_{ij}^{\beta}} & \text{if } q \le q_0 \text{ exploitation} \\ j & \text{otherwise exploration} \end{cases} \tag{27}$$

where η_{ij} is the heuristic function and α, β are the parameters for determining the relative importance of pheromone and heuristic function, respectively;
Step 5: Update solution using Random walk;
Step 6: If $f(\text{best}) < f(\text{current})$, update the best solution with the current solution;
Step 7: Apply global pheromone update:

$$\tau_{ij}^{\text{new}} = (1-\rho)\tau_{ij}^{\text{old}} + \frac{Q}{L}, \tag{28}$$

where ρ is the pheromone evaporation constant and L the length of the current route.
Step 8: Output the best solution.

Algorithm 3. RW_ACS.

Step 1: For all *Direct_Points* in the given sector, find the average of the number of *Remote_Locations* assigned to that sector;
Step 2: Select *Direct_Point* ('k') with the minimum number of the allocated *Remote_Locations*;
Step 3: If the number of *Remote_Locations* allocated to 'k' is less than half of the number of *Remote_Locations* assigned to that sector, go to Step 3; otherwise, stop.
Step 4: Try to assign 'k' and its allocated *Remote_Locations* to nearby *UAV_Route/s* using the proposed *RW_ACS* approach. If possible, Remove 'k' from *GV_Tour*; otherwise, stop.

Algorithm 4. Tour_Optimization.

a DP are distributed among neighboring DPs ensuring that UAV constraints given by Eqs. (15), (16), and (17) remain satisfied. This process was accomplished using Algorithm 4.

4.3.2. Route_Optimization

Once scheduling the UAV routes, the routes might be improved by reassigning the rendezvous node (i.e., Anchor point). In Figure 2, the rendezvous node of UAV is the same as that of the launching node, i.e., the ith *Direct_Point* (shown in green color). However, if such a UAV is capable (in terms of flight endurance limit and time) of reaching the next *Direct_Point*, the GV does not have to wait for this UAV at the ith node since this GV can retrieve it at the $(i+1)$th node (shown in red color). In this case, since the GV can leave the current *Direct_Point* in advance, the overall travel time might decrease provided that this GV had recollected all other UAVs scheduled to be rendezvoused at this *Direct_Point*.

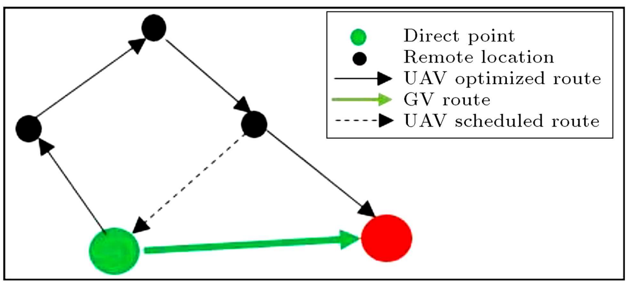

Figure 2. Route optimization.

5. Computations and results

This section examines the proposed MVHUL model and develops the ACS_RW metaheuristic using numerical examples. To the best of the authors' knowledge, no other study considered last-mile deliveries in disaster-affected areas using drone-truck collaborative routing; hence, no publically available datasets are available for the proposed problem. Therefore, the present study took eight well-known Solomon VRPTW randomly clustered (known as RC 101-108) instances for experimentation. The proposed algorithm was coded in Matlab 8.0 on a personal laptop with an i5 processor and 8Gb RAM.

Table 1. Parameter values selected for ACS_RW algorithm.

Parameter	Description	Value
m	No. of ants	10
q_0	Exploration vs. Exploitation decision	0.9
α	Relative importance of Pheromone	1
β	Relative importance of Heuristic	5
ρ	Pheromone Evaporation constant	0.1
Q	Pheromone Deposit constant	100

The success rate of any approximate algorithm depends on different parameter values. Since there are no fixed parameters for ACO, which can be universally applied to all optimization problems, it is required to find the best parameter values for the proposed algorithm with large experimentations and different combinations presented in [29–31]. Finally, the best-suited values are given in Table 1.

5.1. Comparison of the proposed ACS_RW with other approaches

This subsection presents a comparison between the proposed ACS_RW algorithm and some of the commonly used meta-heuristics (such as Genetic Algorithm, GA, Particle Swarm Optimization PSO, Grey Wolf Optimization, GWO, and basic Ant Colony System, ACS). In this regard, four different scenarios based on different percentages (10%, 20%, 30%, and 40%) of locations, which are not approachable by ground vehicles, were taken into consideration. Further, it is assumed that maximum five UAVs are available on each GV. Each algorithm is run in 5000 seconds to minimize the total travel time. Table 2 presents an average of 10 runs of each approach. In this table, #GV and #UAV represent the number of used GVs and UAVs, respectively. Here, TT is the total travel time taken to complete the overall delivery process. Bold-faced values represent the best obtained solutions among all approaches. As observed in Table 2, the TT obtained from the proposed ACS_RW significantly outperforms all other approaches for all four scenarios. It was also found that for 10%, 20%, and 30% UAV-bound locations, ACS_RW used less numbers of GV and the same number of UAVs than all other approaches. However, for 40% UAV-bound customers case, the proposed ACS_RW performed best in terms of #GV and #UAV as well.

5.2. Comparison with other truck-drone collaborative models

The performance of the proposed model was compared with that of [1,2,17], thus considering the problems that are quite similar to the present problem. However, Luo et al. [1] and Ferrandez et al. [2] used single GV (Truck) and single UAV(drone) tandem for delivery; consequently, multiple parallel instances of these algorithms were used for each zone separately to fit them to the current model. Given that Kitjacharoenchai et al. [17] considered a single UAV for each of GV, we limited one UAV per GV for a fair comparison while implementing the proposed as well as all these algorithms. A total of 150 node problems were considered with 70% of chances that a node would be served by truck (GV), as suggested by Chowdhuly et al. [13]. The maximum CPU time allocated to the run of each algorithm was set to 1500 seconds. Each of the algorithms was tested for 20 runs, and the best values of each run were chosen for comparison. Figure 3 presents the box plots of the calculated total travel time (in minutes) for completion of the overall tour. According to this figure, the proposed MVHUL and [17], compared to the other two approaches, are capable of finding the best values for the objective function (i.e., 320 min). While the first quartile of [17] varies between 320 and 340, that of our proposed approach gives a fixed value of 320 (overlapped with the second quartile), indicating the better stability of our proposed approach than that of the others. The median values of our MVHUL are also better than those of the other three approaches. Furthermore, the short-range (320–380) of the proposed algorithm is indicative of its consistency compared to other approaches (value 400 shown by the red arrow can be ignored as it is an outer point). As a result, it was proved that the proposed approach was superior to other similar approaches in terms of the best objective function as well as the consistency in finding the best solutions.

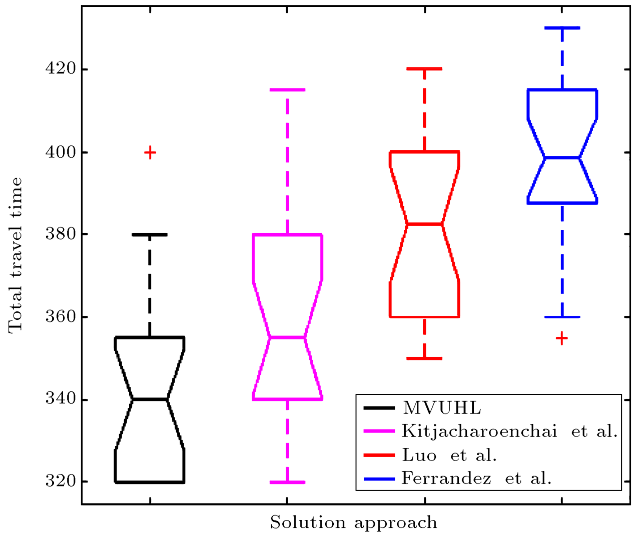

Figure 3. Box-plot comparison of the total travel time of the proposed approach with other existing approaches.

5.3. Sensitivity analysis

Sensitivity analysis was carried out to analyze the effect of the number of UAVs on the total travel

Table 2. Comparison of ACS_RW with commonly used metaheuristics.

Dataset		10%			20%			30%			40%		
		# GV	# UAV	TT	# GV	# UAV	TT	# GV	# UAV	TT	# GV	# UAV	TT
RC 101	GA	13	2	1803.22	12	2	1742.06	11	3	1695.6	11	5	1589.06
	PSO	13	2	1785.32	11	2	1710.43	11	3	1656.18	11	5	1534.81
	GWO	13	2	1702.47	11	3	1699.65	11	3	1623.42	11	4	1540.52
	ACS	13	2	1723.19	12	2	1740.92	11	3	1680.56	11	5	1560.95
	ACS_RW	**12**	**2**	**1650.02**	**11**	**2**	**1564.65**	**9**	**3**	**1520.04**	**9**	**4**	**1438.92**
RC 102	GA	12	2	1744.83	11	2	1587.14	9	3	1429.65	9	5	1499.08
	PSO	12	2	1701.12	11	2	1550.56	9	3	1416.16	9	5	1480.75
	GWO	12	2	1680.12	11	2	1510.2	9	3	1380.4	9	5	1420.16
	ACS	12	2	1710.96	11	2	1525.11	9	4	1408.19	9	5	1433.6
	ACS_RW	**11**	**2**	**1587.12**	**10**	**2**	**1471.43**	**8**	**3**	**1352.11**	**8**	**4**	**1314.06**
RC 103	GA	11	2	1276.12	11	3	1380.94	9	4	1155.68	9	4	1140.6
	PSO	11	2	1250.06	11	3	1342.16	8	4	1140.04	9	4	1114.92
	GWO	11	2	1233.95	11	3	1313.08	8	4	1110.82	9	4	1093.81
	ACS	11	2	1260.09	11	3	1346.26	9	4	1142.17	9	4	1100,02
	ACS_RW	**10**	**2**	**1156.17**	**10**	**2**	**1211.52**	**8**	**3**	**1041.46**	**8**	**3**	**1006.8**
RC 104	GA	10	2	1208.46	10	3	1199.07	8	3	1080.41	8	4	1010.16
	PSO	10	2	1180.36	10	3	1150.83	8	3	1034.16	8	4	987.24
	GWO	10	1	1156.72	10	3	1102.06	8	3	1020.86	8	4	983.06
	ACS	10	2	1160.31	10	3	1123.35	8	3	1018.04	8	4	990.42
	ACS_RW	**9**	**1**	**1120.95**	**9**	**2**	**1021.44**	**8**	**2**	**978.5**	**8**	**3**	**921.83**
RC 105	GA	14	3	1685.62	13	4	1594.57	12	4	1484.55	11	5	1388.9
	PSO	13	3	1637.98	13	4	1582.16	12	4	1482.19	11	5	1376.16
	GWO	13	3	1600.06	13	4	1555.18	11	4	1442.94	11	5	1355.84
	ACS	13	3	1596.44	13	4	1544.73	11	4	1409.02	11	5	1341.93
	ACS_RW	**12**	**2**	**1533.84**	**11**	**3**	**1476.16**	**10**	**3**	**1320.95**	**10**	4	1287.26
RC 106	GA	12	3	1506.31	11	3	1395.17	10	5	1276.18	10	5	1289.82
	PSO	11	3	1500.04	11	3	1388.23	10	5	1242.06	10	5	1254.41
	GWO	11	2	1492.44	11	2	1368.04	10	4	1229.13	10	4	1225.17
	ACS	11	2	1450.79	11	3	1327.17	10	4	1222.64	10	5	1246.05
	ACS_RW	**10**	**2**	**1379.58**	**10**	**2**	**1250.1**	**9**	**3**	**1184.48**	**9**	**3**	**1152.09**
RC 107	GA	12	2	1295.09	11	3	1266.81	9	4	1144.42	9	4	1141.55
	PSO	11	2	1252.41	11	3	1225.58	9	4	1130.06	9	3	1126.18
	GWO	10	1	1230.89	11	2	1237.14	9	4	1126.11	9	3	1110.56
	ACS	11	2	1228.11	11	3	1209.15	9	4	1116.08	9	3	1104.12
	ACS_RW	**10**	**1**	**1198.5**	**10**	**2**	**1110.44**	**8**	**3**	**1057.02**	**8**	**3**	**1029.56**
RC 108	GA	10	2	1056.8	9	3	1040.95	8	3	1008.17	8	5	989.02
	PSO	10	2	1009.43	9	3	1025.18	8	3	999.44	8	4	952.36
	GWO	10	1	997.84	9	3	1008.02	8	3	987.12	8	4	906.71
	ACS	10	2	1002.06	9	3	1004.66	8	3	990.04	8	4	922.41
	ACS_RW	**9**	**1**	**980.46**	**8**	**2**	**950.4**	**7**	**3**	**910.22**	**7**	**4**	**898.74**

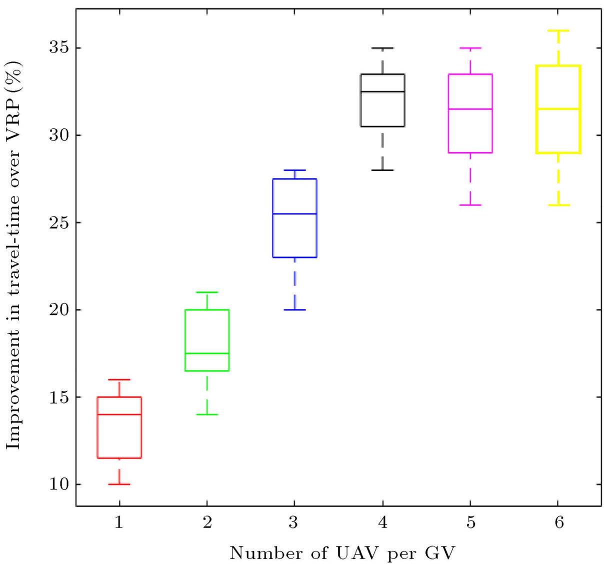

Figure 4. Sensitivity analysis of the effect of number of UAVs deployed on the total travel time.

time to cover all the customers. Figure 4 depicts the percentage improvement in the total travel time over the VRP solution obtained from using the same ACO_RW heuristic without any UAV. As expected, at first, the total travel time was significantly reduced upon increasing the number of UAVs allotted to each GV. However, adding more UAVs beyond the threshold limit (in our case, 4) did not significantly reduce the objective value. One of the possible explanations for such a behavior is that once determining the optimal number of UAVs to serve customers, adding extra UAVs may contribute to an additional distance of moving from AP to a customer and then returning to AP; however, this customer could be easily added to the tour of other (already deployed) UAVs without returning to AP.

5.4. Friedman's analysis

To statistically validate the obtained results, a non-parametric Friedman test was conducted for multiple comparisons at 0.05 level of significance. A null hypothesis H_o: "there is no performance difference between the algorithms" and an alternate hypothesis H_a: "there is a performance difference" has been postulated. The hypothesis was tested on an average of 250 runs on all the compared algorithms on the used data sets, the results of which are presented in Table 3. Since Friedman test statistics is highly significant ($\chi^2_{cal}(3) = 87.065$, $p \leq 0.05$), the null hypothesis can be rejected. After rejecting the null hypothesis, it should be determined which method works better. Therefore, multiple pair-wise comparisons were made using Nemeny procedure (which uses Friedman ranking), the results of which are reported in Table 4. According to Nemenyi post-hoc analysis for multiple comparisons, the proposed HAFA significantly differs ($p < 0.0001$) from GA, PSO, and GWO, while the other contrasts are not significant ($p > 0.05$).

Table 4. p-values of different approaches using Nemenyi post-hoc analysis on Friedman test.

	ACS_RW	**GA**	**PSO**	**GWO**
ACS_RW	1	< 0.0001	< 0.0001	< 0.0001
GA	< 0.0001	1	0.773	0.902
PSO	< 0.0001	0.773	1	0.996
GWO	< 0.0001	0.902	0.996	1

6. Conclusion and future work

A well-planned humanitarian logistic plays a significant role in the right delivery of the right material at the right time to the needy people suffering from shortage of food, clean water, chlorine tablets, and other lifesaving medicines in disaster-affected areas. However, damaged bridges and blocked roads due to debris can hinder last-mile deliveries by ground vehicles only. The collaborative GV-UAV routing was taken into account for last-mile deliveries in disaster-affected areas in this study. A Multi-Vehicle UAV collaborative Humanitarian Logistic (MVUHL) algorithm was proposed for scheduling and routing. This algorithm was broadly classified into three stages:

(a) Initial solution construction;

(b) Optimization using RW_ACS;

(c) Post processing.

A comparison of the proposed algorithm with other similar ones available in the literature confirmed the superiority of our proposed algorithm in terms of total travel time. This study also considered the optimal number of used UAVs. Our analysis revealed that increasing the number of UAVs would reduce the travel time only up to a threshold. However, the application of additional UAVs beyond a certain threshold value would not significantly contribute to improving the travel time.

Our proposed future work in this direction involves an analysis of using heterogeneous UAVs with

Table 3. Friedman statistics.

$Q_{\text{calculated}}$	Q_{critical}	Degree of freedom	p-value (one-tailed)	Level of significance
87.065	7.815	3	< 0.0001	95%

different capacity and flight endurances in last-mile deliveries as well as analysis of the effect of prioritizing the customers and/or dynamic demand-based intelligent routing.

References

1. Luo, Z., Liu, Z., and Shi, J. "A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle", *Sensors (Basel).*, **17**(5), pp. 1–17 (2017).
2. Ferrandez, S.M., Harbison, T., Weber, T., et al. "Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm", *J. Ind. Eng. Manag.*, **9**(2), pp. 374–388 (2016).
3. Murray, C.C. and Chu, A.G. "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery", *Transp. Res. Part C Emerg. Technol.*, **54**, pp. 86–109 (2015).
4. Kenneth, S. "Metaheuristics–the metaphor exposed", *Int. Trans. Oper. Res.*, **22**(1), pp. 3–18 (2015).
5. Akbarpour, N., Kia, R., and Hajiaghaei-keshteli, M. "A new bi-objective integrated vehicle transportation model considering simultaneous pick-up and split delivery", *Sci. Iran.*, **28**(6), pp. 1–32 (2020).
6. Yegane, B.Y., Kamalabadi, I.N., and Farughi, H. "Influence of two different producers in a competitive location problem", *Sci. Iran.*, **27**(5), pp. 2539-2554 (2020).
7. Peng, H., Ying, C., Tan, S., et al. "An improved feature selection algorithm based on ant colony optimization", *IEEE Access*, **6**, pp. 69203–69209 (2018).
8. Deng, W., Xu, J., and Zhao, H. "An improved ant colony optimization algorithm based on immunization strategy", *IEEE Access*, **7**, pp. 20281–20292 (2019).
9. Goel, R.K. and Bansal, S.R., *Hybrid Algorithms for Rich Vehicle Routing Problems: A Survey*, Elsevier Inc. (2020).
10. Jovanovic, R., Tuba, M., and Voß, S. "An efficient ant colony optimization algorithm for the blocks relocation problem", *Eur. J. Oper. Res.*, **274**(1), pp. 78–90 (2019).
11. American Red Cross, *Drones for Disaster Response and Relief Operations*, (April), p. 51 (2015).
12. Mosterman, P.J., Sanabria, D.E., Bilgin, E., et al. "A heterogeneous fleet of vehicles for automated humanitarian missions", *Comput. Sci. Eng.*, **16**(3), pp. 90–95 (2014).
13. Chowdhury, S., Emelogu, A., Marufuzzaman, M., et al. "Drones for disaster response and relief operations: A continuous approximation model", *Int. J. Prod. Econ.*, **188**(February), pp. 167–184 (2017).
14. Rabta, B., Wankmüller, C., and Reiner, G. "A drone fleet model for last-mile distribution in disaster relief operations", *Int. J. Disaster Risk Reduct.*, **28**, pp. 107–112 (2018).
15. Silva, L. de O., Bandeira, R.A. de M., and Campos, V.B.G. "Proposal to planning facility location using UAV and geographic information systems in a post-disaster scenario", *Int. J. Disaster Risk Reduct.*, **36**(February), p. 101080 (2019).
16. Chauhan, D., Unnikrishnan, A., and Figliozzi, M. "Maximum coverage capacitated facility location problem with range constrained drones", *Transp. Res. Part C Emerg. Technol.*, **99**(December), pp. 1–18 (2019).
17. Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., et al. "Multiple traveling salesman problem with drones: Mathematical model and heuristic approach", *Comput. Ind. Eng.*, **129**(June 2018), pp. 14–30 (2019).
18. Murray, C.C. and Raj, R. "The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones", *Transp. Res. Part C Emerg. Technol.*, **110**(February 2019), pp. 368–398 (2020).
19. Liu, Y., Liu, Z., Shi, J., et al. "Two-echelon routing problem for parcel delivery by cooperated truck and drone", *IEEE Trans. Syst. Man, Cybern. Syst.*, **51**(12), pp. 7450–7465 (2021).
20. Abbatecola, L., Fanti, M.P., Fellow, I., et al. "A new cluster-based approach for the vehicle routing problem with time windows", In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pp. 744–749 (2018).
21. Nalepa, J. and Blocho, M. "Adaptive guided ejection search for pickup and delivery with time windows", *J. Intell. Fuzzy Syst.*, **32**(2), pp. 1547–1559 (2017).
22. Alparslan, A. and Science, T. "Comparison of different clustering", *Int. J. Simul. Model*, **18**(4), pp. 574–585 (2019).
23. Liu, J., Yang, J., Liu, H., et al. "An improved ant colony algorithm for robot path planning", *Soft Comput.*, **21**(19), pp. 5829–5839 (2017).
24. Kamaruzaman, A.F. and Zain, A.M. "Levy flight algorithm for optimization problems - A literature review", *Applied Mechanics and Materials*, **421**, pp. 496–501 (2013).
25. Goel, R. and Maini, R. "A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems", *Journal of Computational Science*, **25**, pp. 28–37 (March 2018).
26. Goel, R., Maini, R., and Bansal, S. "Vehicle routing problem with time windows having stochastic customers demands and stochastic service times: Modelling and solution", *J. Comput. Sci.*, **34**, pp. 1–10 (2019).
27. Zhu, H., You, X., and Liu, S. "Multiple ant colony optimization based on Pearson correlation coefficient", *IEEE Access*, **7**, pp. 61628–61638 (2019).
28. Metawa, U.J.N., Shankar, K., and Lakshmanaprabu, S.K. "Financial crisis prediction model using ant colony optimization", *Int. J. Inf. Manage.*, **50**, pp. 538–556 (2020).

29. Bansal, S.R., Goel, R.K., and Maini, R. "An improved ant colony algorithm based on levy flight distribution", *Adv. Math. Sci. J.*, **9**(6), pp. 3907–3916 (2020).

30. Li, Y., Soleimani, H., and Zohal, M. "An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives", *J. Clean. Prod.*, **227**, pp. 1161–1172 (2019).

31. Rao, T.S. "An ant colony and simulated annealing algorithm with excess load VRP in a FMCG company", In *IOP Conference Series: Materials Science and Engineering*, **577**, *International Conference on Advances in Materials and Manufacturing Applications* (IConAMMA-2018) 16–18 August 2018, Bengaluru, India (2019).

Biographies

Sandhya Bansal is an Associate Professor at Maharishi Markandeswar University Mullana. She received her PhD degree from the same University, Jaipur (Rajasthan) and B.Tech degree from Kurukshethra University (Haryana). She supervised eight M.Tech candidates. Her areas of interest include WSN, meta heuristics, and VRP. She has about 25 research papers in international journals and conference proceedings.

Rajeev Goel is an Assistant Professor at Government College, Naraingarh (Haryana) at the Computer Science Department. He completed his Graduation and Post-Graduation at Kurukshetra University, India. He received his PhD from Punjabi University Patiala, India. His research interest includes meta heuristics and operations and mathematical research. He has published more than 20 papers in journals of international repute.

Raman Maini received his BTech (Computer Science and Engineering) from Beant College of Engineering, Gurdaspur, Punjab, India in 1999 and MTech (Computer Science and Engineering) from PAU, Ludhiana, India in 2002. He is currently working as a Professor at Punjabi University, Patiala, India. His current area of research is computer vision (especially noise reduction in medical images, edge detection, and image enhancement) and development of soft-computing-based algorithms.