APEM journal

Advances in Production Engineering & Management
Volume 20 | Number 3 | September 2025 | pp 351–368
https://doi.org/10.14743/apem2025.3.545

ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Two-echelon drone–truck collaborative TSP-based routing for humanitarian logistics with time windows and stochastic demand

Xiao, N.[a,*], **Lan, H.**[a]

[a]School of Economics and Management, Beijing Jiaotong University, Haidian District, Beijing, P.R. China

A B S T R A C T

In humanitarian logistics emergency material transportation and distribution, trucks offer large load capacity and long driving range, whereas drone transportation is independent of ground road conditions but constrained by battery life and payload capacity. The coordination of the two can therefore provide complementary advantages. In this paper, the traveling salesman problem is formulated for a two-echelon emergency material distribution process, spanning transportation from the central warehouse to the distribution center and then to the demand points. In the first stage, transportation from the central warehouse to the distribution center is performed by trucks. In the second stage, trucks and drones collaboratively carry out material distribution from the distribution center to the demand points. Based on the above scenario, this paper aims to minimize the total cost of completing all distribution tasks. The model considers capacity constraints at distribution centers, time window constraints at demand points, and stochastic demand, and establishes a two-echelon traveling salesman problem for humanitarian logistics with truck–drone collaboration. Based on the particle swarm optimization (PSO) framework, a heuristic algorithm named PSO-VD is proposed, which transforms the discrete traveling salesman problem into a continuous encoding and integrates drone routes into truck routes using the 2-opt method. In small-scale instances, the solutions obtained by PSO-VD are compared with those of commercial solvers, demonstrating that the proposed algorithm achieves high accuracy with low computational time. For instances with up to 12 demand points, the algorithm obtains solutions within 150 seconds, with an accuracy deviation of less than 10 % compared to exact solution methods. The applicability of the algorithm proposed in this paper has been demonstrated through large-scale numerical examples. Sensitivity analyses are conducted on key parameters, including the time window penalty coefficient, drone speed, and drone battery capacity, yielding practical managerial insights.

A R T I C L E I N F O

Keywords:
Humanitarian logistics;
Two-echelon routing;
Drone–vehicle collaboration;
Stochastic demand;
Time windows;
Capacity constraint;
Vehicle routing problem (VRP);
Travelling salesman problem (TSP);
Heuristic algorithm;
Particle Swarm Optimization (PSO)

**Corresponding author:*
20113045@bjtu.edu.cn
(Xiao, N.)

Article history:
Received 1 September 2025
Revised 6 October 2025
Accepted 9 October 2025

1. Introduction

Drones are playing a crucial role in humanitarian logistics by enabling faster, more efficient aid delivery to remote and disaster-stricken areas. In Africa, Zipline operates autonomous drones to deliver blood, vaccines, and medical supplies to clinics in Rwanda, Ghana, Nigeria, and beyond, cutting delivery times from hours to minutes. Following Hurricane Maria in Puerto Rico, drones were deployed to assess damage and deliver aid to isolated communities. Similarly, during the COVID-19 pandemic, Matternet's drones transported test kits, PPE, and vaccines between hospitals in the U.S. and Switzerland, reducing exposure risks. The World Food Programme tested drone deliveries for food aid in famine-affected Madagascar, while search and rescue teams used drones after the 2015 Nepal earthquake to locate survivors and assess damage. The UNHCR leveraged drone technology for refugee camp planning and monitoring in Greece and Bangladesh, improving logistical coordination. Additionally, drones have been used in Australia and California for wildfire monitoring and evacuation planning. These innovations are transforming humanitarian logistics by reducing response times, lowering costs, and improving access to life-saving aid. With the rapid development of Industry 4.0 and artificial intelligence, automation technologies have increasingly penetrated manufacturing and service industries, leading to the widespread adoption of drones [1-2]. It has a wide range of applications, including but not limited to industrial manufacturing, logistics and distribution, healthcare, rescue exploration, etc. [3].

The use of drones in humanitarian logistics presents significant advantages, including rapid delivery of life-saving supplies by bypassing infrastructure challenges—as demonstrated by Zipline's medical drone operations in remote Africa, which reduced delivery times from hours to minutes. Drones excel in accessing hard-to-reach areas affected by disasters, conflict, or geography, while also minimizing risks to human responders in hazardous environments. Their cost efficiency, scalability, and real-time data capabilities (evident in post-2015 Nepal earthquake assessments and wildfire monitoring in California and Australia) enhance disaster response and planning. Additionally, eco-friendly operations and improved security monitoring further position drones as transformative tools. However, barriers such as restrictive aviation regulations, limited payload capacities, short battery life, high upfront costs, and technical challenges like weather vulnerability and operator expertise hinder widespread adoption. Privacy concerns, public skepticism, and infrastructure gaps in remote areas further complicate deployment. Addressing these limitations demands collaborative efforts to technology [4], streamline regulations, increase investment, and integrate drones with traditional logistics strategies, ensuring their full potential is realized in delivering timely, efficient, and sustainable humanitarian aid.

A hybrid truck-drone distribution model combines the efficiency of trucks for bulk transport with the agility of drones for last-mile delivery, offering a transformative solution for humanitarian aid, emergency response, and e-commerce logistics. In this system, trucks act as mobile hubs, carrying large quantities of supplies such as food, medicine, or emergency kits while serving as launchpads and recharging stations for drones. Once trucks reach centralized locations—like disaster relief camps or regional hubs—drones take over to deliver smaller, critical items such as vaccines, medical kits, or communication devices to remote, inaccessible, or disaster-stricken areas blocked by rough terrain or infrastructure damage. Operational strategies like parallel delivery (simultaneous truck and drone operations), relay systems (drones picking up payloads from stationary trucks), and on-demand deployment (dynamic launches based on real-time needs) optimize speed and coverage. The model enhances efficiency by allowing trucks to handle bulk logistics while drones bypass traffic and geographical barriers, drastically reducing delivery times and costs. It also improves accessibility in isolated regions and scales rapidly, with multiple drones deployed from a single truck to cover vast areas. However, challenges such as limited drone battery life require frequent recharging at truck hubs, regulatory restrictions on airspace use, and payload capacity constraints compared to trucks must be addressed. By integrating AI-driven route optimization, real-time tracking, and automated scheduling, this hybrid approach promises to revolutionize supply chains, making them faster, more resilient, and adaptable to urgent demands in crisis scenarios or everyday logistics.

This study addresses the emergency material distribution vehicle routing problem from central warehouse to distribution centers and subsequently to demand points. Given that the transportation duration from central warehouse to distribution centers influences the scheduling of last-mile deliveries to demand points, a two-echelon vehicle routing approach is proposed. The first stage, termed trunk transportation, employs large-capacity trucks to transfer materials from central warehouse to distribution centers. The second stage, referred to as last-mile delivery, utilizes coordinated small-scale drones and trucks from distribution centers to demand nodes. Each distribution center stores truck and drone, with each truck carrying one drone to service multiple emergency demand points before returning to its origin. Notably, the trunk transportation stage permits direct delivery by large drones to either distribution centers or demand points without truck-drone coordination.

The research establishes a truck-drone collaborative two-echelon travelling salesman problem model for emergency logistics, prioritizing total mission completion time minimization. The model incorporates operational constraints including payload capacities, mileage limitations, and road accessibility restrictions. A heuristic algorithm is developed to solve this NP-hard problem. The paper is structured as follows: Section 2 presents the literature review, Section 3 details the mathematical formulation, Section 4 describes the heuristic algorithm, Section 5 provides empirical validation, and Section 6 concludes with findings and implications.

2. Literature review

2.1 Vehicle routing problem with drone

Murray and Chu (2015) pioneered a truck-drone collaborative delivery system, framing the problem as the Flying Sidekick Traveling Salesman Problem (FSTSP), where each customer must be served either by a driver-operated truck or a drone coordinated with the truck [5]. Carlsson and Song (2018) demonstrated the efficiency of this system [6], while Agatz *et al.* (2018) developed an exact algorithm termed TSP-D [7]. Yurek and Ozmutlu (2018) proposed an iterative decomposition-based algorithm to minimize the completion time of TSP-D deliveries [8]. Ha *et al.* (2018) introduced a TSP-D variant aimed at minimizing operational costs, including transportation expenses and idle time penalties incurred by interdependencies between trucks and drones [9]. Kim *et al.* (2018) further advanced this with a robust optimization method to address battery duration uncertainties [10]. Roberti and Ruthmair (2020) employed dynamic programming for TSP-D solutions [11], and Tamke and Buscher (2021) incorporated waiting time into the objective function, proposing a novel mixed-integer linear programming (MILP) model enhanced with valid inequalities [12]. Vásquez *et al.* (2021) decomposed the problem into two stages: selecting and sequencing truck-served customers, then assigning remaining nodes to drones [13]. Kang and Lee (2021) devised an exact algorithm based on Benders Decomposition for logistical optimization [14].

In heterogeneous truck-drone routing research, Ham (2018) extended the problem by integrating delivery and pickup tasks with multiple vehicles and depots [15]. Kitjacharoenchai *et al.* (2019) advanced multi-vehicle drone-truck coordination, framing it as a multi-traveling salesman problem with drones (MTSP-D) and developing insertion-based heuristics for large-scale instances (up to 100 nodes) [16]. Murray and Raj (2020) proposed a heuristic decomposition approach for a single truck and multiple drones, breaking the problem into three subproblems [17]. Roberti and Ruthmair (2021) introduced a branch-and-cut algorithm [18], while Mohamed *et al.* (2022) formulated a MILP model for coordinated last-mile delivery using a truck and heterogeneous drones, minimizing completion time via a hybrid simulated annealing and variable neighborhood search algorithm [19]. Dynamic decision-making in operations was explored by Chen *et al.* (2021), who developed a Markov Decision Process (MDP) model for e-commerce logistics, dynamically adjusting delivery rates and pricing strategies under elastic demand [20]. Recent advancements in routing optimization include Lichau *et al.* (2025), who proposed an exact algorithm combining dynamic programming and branch-and-cut pricing to enhance two-echelon truck-drone (2E-VRP-D) efficiency [21]. Mbiadou Saleu *et al.* (2022) extended the parallel drone scheduling TSP (PDSTSP) with a MILP model for multi-vehicle coordination, though computational complexity limited scalability. In intelligent algorithms for complex scenarios [22], Ermağan *et al.* (2022) integrated machine learning with column generation for drone charging station routing, dynamically inserting charging nodes to address range constraints [23]. Stodola *et al.* (2024) designed an adaptive ant colony optimization algorithm (AACO-NC-D) for multi-depot heterogeneous fleets (MDVRP-D), achieving 15-20 % time savings via node clustering and pheromone evaporation [24]. Notably, humanitarian logistics research by Ramadhan *et al.* (2025) introduced a truck-drone-boat (TSP-DB) model for flood disasters, coordinating drones (for dry/flooded zones) and inflatable boats (for flooded areas) to enhance rescue efficiency, though single-drone capacity per mission remains a limitation [25].

This evolving body of work underscores the growing sophistication of hybrid truck-drone systems, driven by algorithmic innovation, multi-objective optimization, and adaptability to diverse operational contexts.

2.2 Emergency logistics vehicle routing problem

Recent advancements in humanitarian logistics have focused on multimodal transportation planning and uncertainty management, with scholars delving into multi-commodity rebalancing, dynamic routing optimization, coordinated delivery, and intelligent algorithm development. In the realm of uncertainty modeling and equity optimization, Gao *et al.* (2021) proposed a bi-objective stochastic mixed-integer nonlinear programming (BOSMINP) model for large-scale disasters, balancing equity and efficiency by minimizing the weighted proportion of unmet demand while integrating priority weights into multi-commodity rebalancing for the first time [26]. Expanding on this, Gao *et al.* (2024) employed a Wasserstein distance-based distributionally robust optimization (DRO) framework to address pre-disaster facility location-inventory prepositioning and post-disaster resource allocation, enhancing robustness against road disruption risks [27]. Similarly, Ahmadi *et al.* (2015) utilized two-stage stochastic programming to manage post-earthquake network failures under the "golden 72-hour" constraint, validating the feasibility of dynamic multi-trip vehicle routing using GIS data [28]. Hou and Liu (2024) used the maximum regret value and Lyapunov central limit theorem to describe the uncertainty of demand and transportation time and established a robust optimization model for cold chain multimodal transportation routes considering mixed uncertainty of carbon emissions [29].

Multimodal coordination and time-sensitive optimization have emerged as key research themes. Yang *et al.* (2025) developed a two-stage stochastic model for truck-drone collaborative scheduling, enabling drones to reload at distribution centers to address single-node surge demand scenarios [30]. Lu *et al.* (2023) further incorporated time-varying weather impacts on drone safety and efficiency, designing a hybrid multi-objective evolutionary algorithm (HMOEA) that demonstrated weather-sensitive decision-making efficacy in the Henan floods case [31]. Alternatively, Yin *et al.* (2023) investigate a robust vehicle routing problem with trucks and drones under uncertain demands and travel times, highlighting optimization strategies for collaborative delivery under humanitarian logistics conditions [32].

Algorithmic innovation and dynamic response strategies have seen notable breakthroughs. Kyriakakis *et al.* (2022) proposed a hybrid tabu search-variable neighborhood descent algorithm (HTS-VND), achieving new optimal solutions for cumulative capacitated vehicle routing problems [33]. Van Steenbergen *et al.* (2023) pioneered the application of deep reinforcement learning (RL) to multi-vehicle routing under uncertain travel times, demonstrating that replacing 50 % of trucks with drones improved objective values by 11-56 % [34]. Amirsahami *et al.* (2025) integrated fuzzy programming with an improved NSGA-II algorithm (M-NSGA-II-AVNS), synchronizing decentralized facility expansion and drone routing in urban rescue chains, reducing costs by 75 % while shortening waiting times [35]. These studies collectively highlight the growing integration of advanced analytics, real-time adaptability, and multimodal synergies in addressing the complexities of humanitarian logistics.

3. Mathematical model

3.1 Problem description

This paper considers the two-stage TSP problem of humanitarian logistics in an emergency scenario. The first stage is from a central warehouse to multiple transit distribution centers, and the second stage is from the transit station to the demand point. At the same time, in the second stage, consider the use of drones and trucks for coordinated delivery, that is, each distribution center has one truck and one drone. At the beginning of the second stage, the truck sets off with cargos and drones to serve the demand point, and the truck is the take-off and landing platform of the drone. When the truck serves the demand point, the drone takes off from the truck and serves other demand points. After the service is completed, when the truck serves another demand point, the drone lands back to the truck. For the humanitarian logistics scenario, we consider the uncertainty of demand, the capacity limit of distribution center and the time window of demand point. Therefore, when the capacity limit of the distribution center is exceeded, or the materials cannot be delivered to the demand point, a penalty coefficient should be given.As shown in figure 1, in stage I, the truck departs from the warehouse loaded with all required supplies and follows the route 1→2→3 to replenish each distribution center sequentially. Upon arriving at distribution center 1, the truck stationed at that center—carrying supplies and drone—departs to serve demand point A. Simultaneously, the drone takes off from the truck to serve demand point B. Subsequently, both the truck and the drone rendezvous at demand point C, after which the drone lands back on the truck. The same delivery procedure is repeated for distribution centers 2 and 3. After completing all delivery tasks, the truck and drone return to their respective distribution centers.

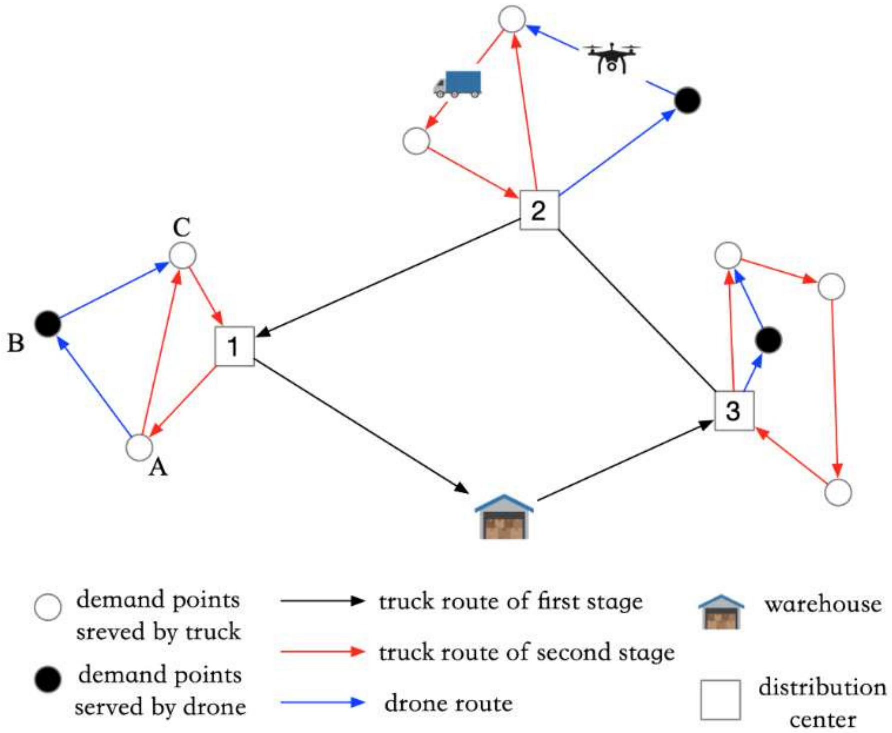

Fig. 1 Two-echelon distribution mode of collaboration between drones and trucks

This problem can be viewed as an extension of the Traveling Salesman Problem (TSP). The model incorporates both an assignment subproblem and a routing subproblem, which involves partitioning demand points into those served by the truck and those served by the drone. Furthermore, it is necessary to decide which demand points will be served by which distribution center. The truck is treated as a traveling salesman, and its route planning constitutes a two-stage TSP problem: the first stage addresses the TSP among distribution centers, while the second stage deals with the TSP among the demand points. Simultaneously, the model must determine the drone's flight route, including the demand point from which the drone launches from the truck, the demand point it serves after launch, and the rendezvous demand point where it returns to the truck after service completion. The synchronization constraint is a critical consideration in truck-drone collaborative delivery, mandating that once the drone launches from the truck to perform a service, it must reunite with the truck before initiating the next service. This necessitates precise coordination between the truck and the drone in both time and space.

3.2 Decision variables and parameters

Decision variables

$x_{ij}^{p} \in \{0,1\}, p \in M, i, j \in N$, whether the truck departing from the p-th distribution center passes through i, j, x_{0j}^{p} indicates whether the truck departs from the distribution center p and serves the demand point j, x_{i0}^{p} indicates whether to return from demand point i to distribution center p.

$y_{ijk}^{p} \in \{0,1\}, p \in M, i, j \in N$, whether the drone departing from the p-th distribution center passes through i, j, k.

$z_{pq} \in \{0,1\}, p, q \in M$, indicates whether there is a path between distribution centers p, q.

$W_{j}^{p} \in \{0,1\}, p \in M, j \in N$, whether the demand point j is served by the drone.

$u_{i}^{p} \in N^{+}, i \in N$, the sequence of truck service demand points.

$ArriveV_{j}^{p} \in R^{+}$, denotes the time when the truck departing from p arrives at j.

$ArriveU_{j}^{p} \in R^{+}$, denotes the time when the drone departing from p arrives at j.

Parameters

M	the set of distribution centers
N	the set of demand points
p, q	index of the distribution center
i, j, k	index of the demand point
d_{ij}	the distance between the points
tv_{ij}	the travel time of the truck between i, j
tu_{ij}	the travel time of the drone between i, j
cv_{ij}	the cost of using the truck between i, j
cu_{ij}	the cost of using the drone between i, j
π_s	probability of occurrence of scenario s
d_i^s	the demand quantity at point i under scenario s
l_i	indicates the latest time at which point of demand i needs to be reached

3.3 Mathematical model

The objective function in Eq. 1 indicates that the total distribution cost is the lowest as the objective. In this article, we use time as the cost of the delivery route. When the speed of the transportation vehicle is one unit, the cost of delivery between two points is equal to the distance between the two points. In addition, pd denotes the capacity penalty cost of the distribution center, representing the penalty incurred when unmet demand requires goods to be transferred from other locations, while pt represents the time-window penalty cost associated with violations of time-window constraints. The parameters α and γ represent the coefficients of cost, respectively. The values of these parameters are determined by two main factors. The first factor is their relative importance: a higher value should be assigned to α when greater emphasis is placed on meeting demands within the required time window, while γ should be assigned a higher value when there is a need to balance distribution demands among multiple distribution centers with limited capacities. Additionally, the magnitude scaling of different cost components should be taken into consideration. Both α and γ can serve to adjust for differences in the scales of various costs, thereby preventing any single cost item from becoming excessively large or small and ensuring that it does not disproportionately influence the total distribution cost.

$$\min \sum_{i,j\in N,p\in M} x_{ij}^{p}\, cv_{ij} + \sum_{i,j\in N} y_{ijk}^{p}\left(cu_{ij} + cu_{jk}\right) + \sum_{i,j\in N} z_{pq} cv_{pq} + pd + pt \tag{1}$$

$$pd = \alpha \sum_{p\in M} \max\left\{C_p - \sum_{s\in S,i\in N} \pi_s d_i^s \left(x_{ij}^{p} + y_{ijk}^{p}\right), 0\right\} \tag{1-1}$$

$$pt = \gamma \sum_{i\in N,p\in M} \max\left\{ArriveV_i^p - l_i, 0\right\} + \max\left\{ArriveU_i^p - l_i, 0\right\} \tag{1-2}$$

$$\sum_{p\in M\cup 0} z_{pq} = 1, q \in M \tag{2}$$

$$\sum_{q\in M\cup 0} z_{pq} = 1, p \in M \tag{3}$$

$$\sum_{p\in M\cup 0} z_{pq} = \sum_{o\in M\cup 0} z_{qo}, \forall\, q \in M \tag{4}$$

$$\sum_{i=0}^{n}\sum_{p=1}^{m} x_{ij}^{p} = 1 - W_j^p, j \in N \tag{5}$$

$$\sum_{j=1}^{n+1}\sum_{p=1}^{m} x_{ij}^{p} = 1 - W_j^p, i \in N \tag{6}$$

$$\sum_{i=0}^{n} x_{ij}^{p} = \sum_{k=1}^{n+1} x_{jk}^{p}, \forall\, j \in N, \forall\, v \in V \tag{7}$$

$$u_i^p - u_j^p + 1 \leq (c+1)\left(1 - x_{ij}^{p}\right), \forall\, i \in N_{-}, j \in N^{+}, p \in M \tag{8}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{u=1}^{m} y_{ijk}^{p} = W_j^p, \forall\, j \in N \tag{9}$$

$$u_i^p - u_k^p < m\left(1 - \sum_{j=1}^{n} y_{ijk}^{p}\right) \tag{10}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{u=1}^{m} y_{ijk}^{p} \leq \sum_{h=1}^{n}\sum_{v=1}^{m} x_{hi}^{v}, \forall\, i \in N \tag{11}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{u=1}^{m} y_{ijk}^{u} \leq \sum_{h=1}^{n}\sum_{v=1}^{m} x_{kh}^{v}, \forall\, k \in N \tag{12}$$

$$ArriveV_j^p \geq ArriveV_i^p + tv_{ij} - M\left(1 - x_{ij}^{p}\right), \forall\, i,j \in N, p \in M \tag{13}$$

$$ArriveU_j^p \geq ArriveU_i^p + tu_{ij} - M\left(1 - y_{ijk}^{p}\right), \forall\, i,j,k \in N, p \in M \tag{14}$$

$$ArriveU_k^p \geq ArriveU_j^p + tu_{jk} - M\left(1 - y_{ijk}^{p}\right), \forall\, i,j,k \in N, p \in M \tag{15}$$

$$ArriveU_i^p \geq ArriveV_i^p - M\left(1 - \sum_{j=1}^{n} y_{ijk}^{p}\right), \forall\, i,k \in N, p \in M \tag{16}$$

Constraints Eqs. 2-4 represent the path from the central warehouse to each distribution center and realize the flow balance constraint. Constraints Eqs. 5-6 indicate that the route of the truck will exist only if the demand point is satisfied by the truck. Constraint Eq. 7 represents the flow balance of the distribution truck in the second stage. Constraint Eq. 8 removes the sub-cycle constraint for the truck in the second stage. Constraint Eq. 9 indicates that the route of the drone will exist only when the demand point is met by the drone. Constraint Eq. 10 limits the delivery direction of drone, that is, the direction of drone delivery demand point should be consistent with the direction of truck travel. Constraints Eqs. 11-12 indicate that only the demand point passed by the truck can allow the drone to take off and land. For example, only the truck serves point i, and point i can be used as the take-off point of the drone. The constraint Eq. 13 is used to calculate the time for the truck to reach point k. Constraints Eqs. 14-15 denote the calculation of the time for the drone to arrive at the served demand point j and the landing point k. Constraint Eq. 16 represents the temporal coordination between the truck and the drone, that is, if the drone takes off at the demand point i, the take-off time of the drone should not be earlier than the time when the truck arrives at the demand point i.

4. Algorithm

4.1 Headings and subheadings

Based on the framework of particle swarm optimization (PSO), this paper proposes a particle swarm optimization heuristic algorithm suitable for the collaborative distribution of truck and drone (PSO-VD) to solve the mixed integer programming model. In each iteration of the algorithm, first, the use of drones is not considered, that is, all demand points are assigned to trucks, and the path of two-stage multi- trucks is calculated. Then, dynamic programming (neighborhood search) is used to reassign some demand points originally served by truck to drone services and a take-off point and landing point are assigned to the drone.

When arranging the route for trucks, the first stage is the sequence from the warehouse to each distribution center, while the second stage each distribution center needs to serve multiple demand points. It is necessary to determine the allocation of distribution centers (that is, which demand point is served by which distribution center) and the two-stage routing arrangement at the same time, so that the total route is the shortest. Because PSO is usually used for continuous space problems, while TSP is a discrete optimization problem. At this time, we need to consider how to represent the solution of these two-stage problems and how to update the position and velocity of particles. Therefore, this paper transforms the route coding of TSP into a continuous mode, which makes it more suitable for using PSO to solve.

In terms of the structure of the solution, the truck route needs to include the visit order of the first stage distribution centers, the allocation of demand points, and the visit order of demand points within each distribution center. The position vector $X[i] = [m + 2n]$ of each particle includes m (number of distribution centers) real numbers for distribution center order, and $2n$ real numbers for the allocation and order of demand points. The total dimension is $m + 2n$. The first m real values are arranged from small to large to get the access order of the distribution center. For example, if the real value corresponding to the distribution center P1 is 0.3, P2 is 0.5, and P3 is 0.1, the order is P3 (0.1), P1 (0.3), and P2 (0.5). demand point assignment and order section, each demand point corresponds to two real numbers, the first determines which transit station to be assigned to, and the second determines the order of access in that distribution center. For each demand point i, the $(m + 2i - 1)$-th real value is used to map to the distribution center. For example, scale the real value x_i to $[0, 1]$, and then determine the distribution center p according to m equal partitions. Assuming that there are 2 distribution centers, then $x(m + 2i - 1) \in (0, 0.5)$ means that the demand point i is in charge of the first distribution center, and $x(m + 2i - 1) \in (0.5, 1)$ means that the demand point i is in charge of the second distribution center. In demand point order calculation, for each distribution center p, all demand points assigned to s are collected, and the order of visits at this distribution center is determined according to the order of size of their $(m + 2i)$-th real value.

After determining the current truck route, it is necessary to clarify the execution sequence of drone tasks, determine the demand points served by drone, select appropriate take-off points and landing points, and then update the path for the next stage of optimization. In terms of route calculation of drone, it needs to be considered that drones can only serve one demand point at a time, and each mission must take off from a certain demand point of the truck and land at another demand point of the truck after service. Every time a mission is completed, the landing point of the drone becomes part of the truck route, and the take-off point of the next mission must be after this landing point. For example, if the drone takes off from demand point 1, serves demand point 2 and lands at demand point 4, then the next route of the truck is 1 → 4, after which the drone can only take off from node 4 or subsequent nodes (such as 5, 9, etc.) to perform the next mission. When performing calculations, we first determine the following parameters, K is the maximum allowable drone route length, t is the number of task calculations, $x_{ij,t} \in \{0, 1\}$, and whether the truck travels from i to j during t calculations, its initial solution is the current position of the particle; at the same time, we denote the take-off point of the t-th mission by $i \in N, k \in N$ represents the landing point of the t-th task, and $j \in N$ represents the demand point served by the t-th iterative task. We need to find a drone path (i, j, k) in the t-th iteration such that $max\{(cv_{ij} + cv_{jk}) - max\{cv_{ik}, cu_{ij} + cu_{jk}\} > 0$, take point j as the point served by the drone and delete it from the path of the delivery truck, when the maximum flight distance of the drone is reached, or $max\{(cv_{ij} + cv_{jk}) - max\{cv_{ik}, cu_{ij} + cu_{jk}\} < 0$, then stops the calculation and updates the truck route.

After the calculation of each particle is completed, the global optimal solution $gbset$ and the optimal solution $pbest[i]$ of each particle can be obtained. At this time, in order to ensure the availability of PSO in the next stage, the optimal solution of the particle should be changed to the truck route under this fitness function, that is, the position of the particle at this time is still $x[i] = x[m + 2n]$, which only includes the distribution center allocation and truck route. For example, before inserting the demand point served by drone, the route of the truck is 1-2-3-4-5.

At this time, enter the next iteration, and use the velocity update formula and position update formula in PSO to update the particles.

Speed update:

$$v_i(t+1) = v_i(t) + r1\big(pbest - x_i(t)\big) + r2\big(gbest - x_i(t)\big)$$

Location update:

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Now, based on the above analysis, the steps of designing PSO in this paper are as follows:

Step1. Initialize the particle swarm, the position vector of each particle includes m (number of distribution centers) real numbers for transit station order, $2n$ (n demand points, two real numbers per demand point: allocation and order) real numbers. The total dimension is $m + 2n$.

Step 2. For each particle, decode its position:
a) The first m real numbers determine the access order of the distribution center: arrange them according to their values from small to large to get a sequential list of distribution centers.
b) For each demand point i, calculate its assigned distribution centers: take the real value of the $(m + 2i - 1)$-th dimension, map it to $[0, 1]$, and then divide it into m intervals to determine the corresponding distribution centers.
c) For each distribution centers p, collect all demand points assigned to p, and then determine their access order at the distribution center according to the magnitude of the real value of the $(m + 2i)$-th dimension of these demand points.

Step 3. Insert the route of the drone into the route of each delivery and update the route of the truck.

Step 4. Calculate fitness (total path length):
a) Calculate the first stage path: start from the warehouse, visit it in sequence according to the decoded distribution center order, and then return to the TSP route length of the warehouse.
b) For each distribution center p, calculate the TSP route length of its demand point visit order: start from distribution center p, visit according to the decoded truck and drone route sequence, and then return to the route length of p.
c) The total fitness is the sum of the route lengths of the two stages.

Step 5. Update $pbest$ and global $gbest$ for each particle.

Step 6. Update the velocity and position of each particle according to the PSO velocity and position update formula.

Step 7. Repeat steps 2-6 until the termination condition is met.

5. Numerical experiments

5.1 Algorithm performance

In this section, we use three groups of cases to conduct numerical experiments, 15 random trials are conducted for each group of cases and directly compare the solution results of the heuristic algorithm with those of CPLEX, so as to verify the effectiveness of our proposed algorithm. In terms of parameter generation, in the three groups of cases, we assume that there is a (1000, 1000) area, (0, 0) is the central warehouse, and there are two distribution centers with random locations and capacity of (10, 50). There are 8, 10, and 12 demand points in the area, with random locations. Assuming the demand at each point follows a uniform distribution with a range of (1, 10), we use the mathematical expectation of demand as the actual demand for the point to handle the scenario of random demand, and the time window is a random integer of 0-3000. The speed of the drones is twice that of the trucks, and the travel time of the trucks is 1 unit. We use the distance between two points as the transportation cost, and both the time penalty cost coefficient and the capacity penalty coefficient are set to 1. In the setting of particle swarm optimization parameters, the algorithm uses Python for encoding, the number of particles is set to 50, and the number of iterations is 1000.

Table 1 Comparison of CPLEX and PSO-VD solution results with 8 demand points

Instance	N	MIP		PSO-VD		δ (%)
		Obj.	Time (s)	Obj	Time (s)	
1	8	4867.34	64.35	5281.22	63.91	7.84
2		4565.41	19.65	4630.15	62.06	1.40
3		4864.18	15.06	5078.80	60.78	4.23
4		5333.14	14.66	5905.38	60.12	9.69
5		6930.11	76.57	7914.67	55.43	12.44
6		5599.33	25.06	5740.81	48.25	2.46
7		4995.21	18.90	6631.94	58.93	24.68
8		6825.00	88.75	8344.14	60.10	18.21
9		4816.82	10.26	4819.56	59.22	0.06
10		4219.41	13.23	4906.92	54.36	14.01
11		4287.20	59.33	4544.19	55.67	5.66
12		4737.35	63.07	5615.66	54.82	15.64
13		4932.30	37.49	4998.69	57.99	1.33
14		6093.29	37.25	7096.12	62.15	14.13
15		4582.94	20.67	4643.63	51.87	1.31
AVE			37.62		57.71	8.87

Table 2 Comparison of CPLEX and PSO-VD solution results with 10 demand points

Instance	N	MIP		PSO-VD		δ (%)
		Obj.	Time (s)	Obj.	Time (s)	
16	10	4696.49	24.00	4915.70	99.33	4.46
17		22747.20	143.69	39044.15	94.14	41.74
18		5821.90	492.41	6259.63	98.58	6.99
19		5508.59	139.35	6073.98	86.99	9.31
20		10257.00	435.76	12484.22	109.54	17.84
21		5423.98	916.37	5515.46	104.04	1.66
22		4679.57	144.78	4911.54	104.69	4.72
23		7391.52	529.75	10495.42	96.81	29.57
24		4399.35	20.55	5400.09	96.25	18.53
25		5424.60	135.92	5895.59	83.22	7.99
26		10379.20	597.43	14956.40	86.23	30.60
27		5303.12	121.43	5449.65	96.76	2.69
28		4986.49	130.42	5437.93	81.88	8.30
29		5967.45	1449.71	6078.02	90.70	1.82
30		4988.55	684.39	5941.99	103.77	16.05
AVE			397.73		95.53	13.49

Table 3 Comparison of CPLEX and PSO-VD solution results with 12 demand points

Instance	N	MIP		PSO-VD		δ (%)
		Obj.	Time (s)	Obj	Time (s)	
31	12	5779.91	3832.27	6086.39	168.70	5.04
32		4969.05	2104.76	5133.52	132.09	3.20
33		5268.61	1633.16	5901.28	117.92	10.72
34		6631.93	5145.78	7712.96	155.69	14.02
35		5098.44	905.76	5657.63	211.16	9.88
36		5878.99	4869.61	7220.25	180.32	18.58
37		6611.72	5560.58	8571.16	183.96	22.86
38		6408.76	1229.02	7113.89	150.48	9.91
39		12457.80	4639.87	20612.16	162.72	39.56
40		7074.39	9653.25	7312.15	162.68	3.25
41		5779.91	4543.35	6086.39	140.69	5.04
42		22321.60	3581.06	25099.58	168.81	11.07
43		5787.84	7668.67	6760.20	125.29	14.38
44		5273.36	1822.55	5593.84	166.34	5.73
45		5678.51	1349.59	5993.26	147.47	5.25
AVE			3902.62		158.29	11.90

Tables 1 to 3 show the comparative analysis of three sets of experiments with a total of 45 cases. In these tables, MIP represents the scenario of direct solution using the CPLEX commercial solver; PSO-VD represents the scenario of solution using the heuristic algorithm designed in this paper; N denotes the number of demand points, Obj. stands for the objective value of the solution result; and Time(Sec) indicates the runtime of the code; δ is the gap between the optimal solution and the PSO-VD solution results, and AVE is the average solution time and average solution gap of 15 sets of cases. When there are only 8 demand points, the optimal solution of CPLEX can be obtained in a short time, the runtime of the code of PSO-VD is slightly longer than that of CPLEX, and the gap between the heuristic algorithm and the exact optimal solution is 8.87 %. When there are 10 demand points, the average runtime of the code of CPLEX is 397.73 seconds, the runtime of the code of PSO-VD is 95.53 seconds, and the gap between the heuristic algorithm and the exact optimal solution is 13.49 %. When there are 12 demand points, the runtime of the code of CPLEX increases to 3902.62 seconds, which is almost not allowed in emergency logistics, because there is a certain suddenness, and emergency logistics often needs to respond quickly. The runtime of the code of PSO-VD is about 150 seconds, and the increase of the runtime of the code is basically linear with the increase of demand point. The difference between the heuristic algorithm and the exact optimal solution is 11.90 %. An increase in the number of demand points does not lead to a deterioration in solution quality. When the number of demand points becomes large, directly applying commercial solvers such as CPLEX results in excessive computational time. The two-stage particle swarm heuristic algorithm proposed in this paper can greatly shorten the solution time and ensure the solution quality. It provides a decision-making tool for the use of drones and trucks for emergency supplies distribution after an emergency event.

Table 4 Solution results with 50 demand points

Instance	Obj.	CW	CI	ND	Time (s)
1	4944.76	1777.80	0.00	8	147.19
2	6085.32	2020.49	0.00	7	150.33
3	4805.52	1574.88	274.87	8	167.93
4	4265.48	1105.71	246.01	9	138.14
5	21223.89	2706.64	7123.06	9	143.62
6	3530.08	1057.61	0.00	5	119.41
7	23939.49	1293.60	10134.23	7	136.86
8	6754.44	1177.63	1298.65	8	127.89
9	20663.29	1469.28	7883.07	8	160.12
10	7938.67	2749.23	31.18	7	152.60

Table 5 Solution results with 100 demand points

Instance	Obj.	CW	CI	ND	Time (s)
1	48957.25	6480.77	16208.92	9	537.90
2	64206.56	8591.16	21824.23	9	788.62
3	57583.76	7630.96	19532.70	11	848.06
4	76037.76	5468.44	31158.02	9	863.16
5	31473.14	5417.34	8685.06	13	760.33
6	30708.04	5924.46	7807.30	14	694.09
7	44349.27	6020.26	13501.16	13	822.34
8	64874.75	7635.61	21775.38	14	843.22
9	48225.69	7221.32	14519.01	11	812.16
10	48986.75	5096.46	16697.15	14	779.57

In this section, following the procedure of the heuristic algorithm, we first conducted computations for scenarios involving large-scale demand points and presented the results, thereby verifying the broad applicability of the proposed algorithm. Given that the heuristic algorithm proposed in this paper includes several randomly generated steps, the experimental process was repeated ten times for different scales of demand points to ensure the robustness of the heuristic approach. Furthermore, we considered the presence of three distribution centers within the region, with the remaining parameter settings consistent with those in the previous section. Tables 4 and 5 present the case analysis results for 50 and 100 demand points, respectively. For each demand point scale, the coordinates of the demand points were randomly generated ten times. In Tables 4 and 5, the first column indicates the case number, followed by *Obj.* representing the solution outcome—specifically, the minimized delivery time, time window penalty cost, and capacity penalty cost. CW denotes the time window penalty cost incurred due to the inability to meet the time window requirements of demand points, while Cl represents the distribution center capacity penalty cost resulting from the failure to fully satisfy the demand of certain points; ND denotes the number of demand points served by drones. The last column records the code running time. From the results presented in Tables 4 through 5, it can be observed that the proposed heuristic algorithm can obtain solutions relatively quickly. In the case of 50 demand points, the algorithm requires approximately two minutes to produce a solution, whereas for 100 demand points, it takes about ten minutes. As an NP-hard problem, the computational time of the proposed algorithm significantly surpasses that of commercial solvers, albeit with a slight trade-off in solution accuracy. In the context of humanitarian logistics, there is often a pressing need for rapid decision-making. Therefore, the model and algorithm proposed in this paper demonstrate considerable practical utility.

5.2 Sensitivity analysis

Time window penalty cost coefficient α

In this set of experiments, this paper examines the impact of the time window penalty cost coefficient α on the solution outcomes. α represents the importance of the time window, and different ranges can be assigned depending on the type of goods being delivered. For instance, a higher α value can be set for urgent supplies such as water and medicines, whereas a lower α can be used for ordinary goods.

Table 6 Sensitivity analysis results of time window penalty coefficient

α	Instance	Obj.	CW	CI	ND
0.5	1	7750.12	2585.32	1199.31	7
	2	2691.99	1049.21	0	8
	3	3182.84	1343.16	0	8
	4	18007.03	1387.50	7283.77	9
	5	14235.91	1286.79	5408.63	9
	AVE	9173.58	1530.50	2778.34	8.2
1	1	12695.17	827.51	4701.94	9
	2	4114.84	1260.34	0	8
	3	13335.70	1028.90	0	8
	4	4004.98	1211.41	0	9
	5	5421.97	1884.95	0	8
	AVE	7914.53	1242.62	940.38	8.4
2	1	9028.48	1786.24	0	9
	2	3995.90	841.12	0	5
	3	7304.66	1529.77	0	7
	4	12048.78	2606.94	1116.98	8
	5	32193.36	1154.40	13536.57	8
	AVE	12914.23	1583.69	2930.71	7.4
5	1	8934.08	1210.13	0	7
	2	11150.71	938.41	0	8
	3	21912.31	1239.09	5601.84	7
	4	18108.31	1612.05	1448.62	6
	5	18903.77	3391.60	1556.16	5
	AVE	15801.83	1678.25	1721.32	6.6

As shown in Table 6, as the time window penalty cost coefficient α gradually increases, the frequency of drone usage actually decreases. When α is 0.5, an average of 8.2 demand points are served by drones. At α = 2, drones serve 7.4 demand points, and when α reaches 5, only 6.6 demand points are served by drones. We analyze that this is likely because demand points with stricter time requirements are more sensitive to delivery reliability, making trucks a more suitable choice to ensure dependable service. Since drones operate in coordination with trucks—requiring pickup from trucks before continuing delivery—they may not be able to handle more complex or time-critical demands. Therefore, in practical humanitarian emergency logistics and the distribution of relief supplies, drones may still play only a supporting role. For the delivery of urgent supplies, trucks remain the most reliable option.

Drone battery capacity

This study investigates the impact of drone battery capacity on delivery outcomes. To simplify the problem and control for the influence of cargo weight on the results, the maximum travel distance of the drone is used as a substitute for battery capacity, while both cargo weight and the drone's own weight are disregarded.

Table 7 Sensitivity analysis results of drone battery capacity

Drone battery capacity	Instance	Obj.	CW	CI	ND
50	1	26631.77	1477.61	11147.77	3
	2	25720.24	2701.90	14243.07	5
	3	27182.15	1364.25	16512.97	5
	4	22107.84	992.58	9161.18	5
	5	32106.39	1210.91	14111.32	6
	AVE	26749.68	1549.45	9035.26	4.8
100	1	23161.46	1458.10	9173.11	7
	2	28082.45	1845.50	11258.03	7
	3	26697.90	2346.76	5058.88	6
	4	25201.54	1307.82	9462.57	7
	5	25044.44	1007.07	10622.33	8
	AVE	25637.56	1593.05	9114.98	7
500	1	25791.63	1668.10	10513.68	9
	2	22520.63	1818.97	7480.15	7
	3	21125.52	1307.82	7462.57	6
	4	23893.09	1007.07	8622.33	8
	5	25044.44	1007.07	10622.33	8
	AVE	23675.06	1361.81	8940.21	7.6
1000	1	19771.22	1060.81	8164.78	9
	2	23722.23	2267.25	13853.96	7
	3	17212.17	1507.37	6118.70	6
	4	23629.79	1807.33	7371.58	8
	5	26705.28	1241.84	11451.42	7
	AVE	22208.14	2074.64	9392.09	7.4
2000	1	23809.98	1909.41	9297.58	7
	2	22668.43	2184.73	8418.20	9
	3	27672.10	1512.33	11126.80	8
	4	13709.28	1075.12	4824.58	8
	5	18238.48	447.06	7443.35	7
	AVE	21219.65	1425.73	8222.10	7.8

The battery capacity—expressed as the maximum allowed travel distance—is set at 50, 100, 500, 1000, and 2000 units, respectively. As shown in Table 7, since terrain and road conditions do not impose constraints, the operational cost of drones remains relatively low. When the battery capacity increases, drones are able to serve more points, thereby reducing the overall delivery cost. However, it should also be noted that once the battery capacity reaches a certain threshold, the frequency of drone deployment no longer increases. Specifically, when the battery capacity rises from 50 to 500, the number of demand points served by drones increases from 4.8 to 7.6. Beyond this point, further increases in battery capacity result in almost no change in the number of demand points served. Therefore, in humanitarian logistics, rationally selecting drones with appropriate parameters can enhance logistics efficiency while ensuring cost-effectiveness.

Drone speed

Since drone flight is not constrained by terrain and can generally follow a straight-line path and given that drones typically have a higher speed than trucks, even though drones serve only as an auxiliary delivery tool in this distribution mode, this paper posits that an increase in drone flight speed can significantly reduce total delivery costs. Therefore, this subsection focuses on drone speed as the subject of sensitivity analysis. Assuming four speed ratios between the truck and the drone—1:1, 1:2, 1:5, and 1:10—the model was solved for five randomly generated sets of demand point coordinates to examine the impact of drone speed on the solution outcomes. The speed of the truck is set to 1 unit. Each demand point set comprises 30 demand points, with all other parameters consistent with previous settings.

Table 8 presents the solution results. When the speed ratio between the truck and the drone is 1:1, the average delivery cost is 26,720.68, and on average only 6.6 demand points are served by drones per delivery. When the speed ratio increases to 1:2, the total delivery cost decreases to 25637.56, and drone utilization improves, with an average of 8.2 demand points served by drones.

Table 8 Sensitivity analysis results of drone speed

Truck speed: Drone speed	Instance	Obj.	CW	CI	ND
1:10	1	18373.86	1472.61	6921.84	9
	2	23188.60	1335.77	9424.22	8
	3	19753.17	1574.46	7477.95	8
	4	15435.91	1251.83	5292.83	9
	5	30953.23	1675.55	12832.72	8
	AVE	21540.95	1462.04	8389.91	8.4
1:5	1	18331.24	1143.74	2979.78	8
	2	17939.15	1626.49	6360.60	10
	3	18242.59	1874.39	5790.26	9
	4	24846.43	2333.73	8905.28	6
	5	24472.16	1347.13	9955.68	9
	AVE	20766.31	1665.09	6798.32	8.4
1:2	1	23579.75	1788.26	8952.47	8
	2	20085.72	1380.06	7987.17	10
	3	20023.24	1263.56	7978.57	8
	4	10703.64	1114.93	3323.57	7
	5	19371.06	1877.38	6872.05	8
	AVE	18752.68	1484.84	7022.77	8.2
1:1	1	23089.46	2205.97	8395.21	8
	2	26293.74	1424.03	10894.12	6
	3	22341.35	1940.27	8398.79	6
	4	29984.24	972.74	13358.06	7
	5	31894.64	1028.21	14253.35	6
	AVE	26720.69	1514.24	11059.91	6.6

However, as drone speed continues to increase—specifically when the speed ratio of drone to truck reaches 1:5 and even 1:10—the solution outcomes do not show further improvement, nor does the frequency of drone usage increase significantly. This implies that appropriate transportation modes should be selected based on practical circumstances, considering factors such as technological upgrade costs or procurement expenses. A blind pursuit of higher performance parameters may not yield substantial practical benefits.

6. Conclusion

This study investigates a two-echelon truck-drone collaborative routing problem for humanitarian logistics under stochastic demand and time-window constraints. The problem is structured in two stages: first, trucks transport emergency supplies from a central warehouse to multiple distribution centers; second, each distribution centre deploys a truck equipped with a drone to serve demand points in a coordinated manner. The drone may take off from the truck when it stops at a certain demand point, to deliver to another point, and later land back on the truck when the truck arrives at a subsequent point, thereby enhancing delivery flexibility and coverage. To address this problem, we formulate a mixed-integer programming model that aims to minimize the total distribution cost, incorporating transportation cost, capacity-shortage penalty, and time-window violation penalty. Building on the particle swarm optimization framework, we propose a novel heuristic algorithm named PSO-VD, which integrates particle swarm optimization with a drone-insertion mechanism based on the 2-opt method. This approach transforms the discrete routing problem into a continuous encoding scheme, enabling an efficient search within the solution space. Numerical experiments on small-scale instances demonstrate that, compared with CPLEX, PSO-VD achieves near-optimal solutions with an average optimality gap between 8.87 % and 13.49 %, while significantly reducing computational time. For larger-scale instances, the algorithm remains practical, obtaining feasible solutions within minutes—a crucial capability for emergency response scenarios. Furthermore, sensitivity analyses are conducted on parameters such as the time-window penalty coefficient, drone battery capacity, and drone speed, yielding several managerial insights. Stricter time windows reduce the deployment of drones, indicating that trucks are preferred for time-critical deliveries. Drone battery capacity improves delivery efficiency up to a certain threshold, beyond which further capacity increases yield diminishing returns. Although higher drone speed moderately enhances system performance, excessively high speeds do not lead to substantial improvements, suggesting that cost-effective technology selection is essential.

Future research could focus on enhancing the algorithm through hybrid meta-heuristics, incorporating more dynamic or realistic constraints (e.g., weather-dependent drone status), or developing decomposition-based exact methods for medium-scale instances.

References

[1] Perez-Grau, F.J., Martinez-de Dios, J.R., Paneque, J.L., Acevedo, J.J., Torres-González, A., Viguria, A., Astorga, J.R., Ollero, A. (2021). Introducing autonomous aerial robots in industrial manufacturing, *Journal of Manufacturing Systems*, Vol. 60, 312-324, doi: 10.1016/j.jmsy.2021.06.008.

[2] Gomes, O., Lins de Moraes, M. (2024). Forks in the road: Modelling the economic prospects of artificial intelligence, *Economic Computation and Economic Cybernetics Studies and Research*, Vol. 58, No. 3, 113-128, doi: 10.24818/18423264/58.3.24.07.

[3] Zhang, H.Q. (2024). Optimizing obstacle avoidance path planning for intelligent mobile robots in multi-obstacle environments, *Advances in Production Engineering & Management*, Vol. 19, No. 3, 358-370, doi: 10.14743/apem2024.3.512.

[4] Goyal, S.B., Rajawat, A.S., Solank, R.K., Patil, D., Potgantwar, A. (2024). A trustable security solution using XAI for 5G-enabled UAV, *Journal of Logistics, Informatics and Service Science*, Vol. 11, No. 4, 73-86, doi: 10.33168/JLISS.2024.0404.

[5] Murray, C.C., Chu, A.G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery, *Transportation Research Part C: Emerging Technologies*, Vol. 54, 86-109, doi: 10.1016/j.trc.2015.03.005.

[6] Carlsson, J.G., Song, S. (2018). Coordinated logistics with a truck and a drone, *Management Science*, Vol. 64, No. 9, 4052-4069, doi: 10.1287/mnsc.2017.2824.

[7] Agatz, N., Bouman, P., Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone, *Transportation Science*, Vol. 52, No. 4, 965-981, doi: 10.1287/trsc.2017.0791.

[8] Yurek, E.E., Ozmutlu, H.C. (2018). A decomposition-based iterative optimization algorithm for traveling salesman problem with drone, *Transportation Research Part C: Emerging Technologies*, Vol. 91, 249-262, doi: 10.1016/j.trc.2018.04.009.

[9] Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H. (2018). On the min-cost traveling salesman problem with drone, *Transportation Research Part C: Emerging Technologies*, Vol. 86, 597-621, doi: 10.1016/j.trc.2017.11.015.
[10] Kim, S.J., Lim, G.J., Cho, J. (2018). Drone flight scheduling under uncertainty on battery duration and air temperature, *Computers & Industrial Engineering*, Vol. 117, 291-302, doi: 10.1016/j.cie.2018.02.005.
[11] Roberti, R., Ruthmair, M. (2021). Exact methods for the traveling salesman problem with drone, *Transportation Science*, Vol. 55, No. 2, 315-335, doi: 10.1287/trsc.2020.1017.
[12] Tamke, F., Buscher, U. (2021). A branch-and-cut algorithm for the vehicle routing problem with drones, *Transportation Research Part B: Methodological*, Vol. 144, 174-203, doi: 10.1016/j.trb.2020.11.011.
[13] Vásquez, S.A., Angulo, G., Klapp, M.A. (2021). An exact solution method for the TSP with drone based on decomposition, *Computers & Operations Research*, Vol. 127, Article No. 105127, doi: 10.1016/j.cor.2020.105127.
[14] Kang, M., Lee, C. (2021). An exact algorithm for the heterogeneous drone-truck routing problem, *Transportation Science*, Vol. 55, No. 5, 1088-1112, doi: 10.1287/trsc.2021.1055.
[15] Ham, A.M. (2018). Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming, *Transportation Research Part C: Emerging Technologies*, Vol. 91, 1-14, doi: 10.1016/j.trc.2018.03.025.
[16] Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J.M.A., Brunese, P.A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach, *Computers & Industrial Engineering*, Vol. 129, 14-30, doi: 10.1016/j.cie.2019.01.020.
[17] Murray, C.C., Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones, *Transportation Research Part C: Emerging Technologies*, Vol. 110, 368-398, doi: 10.1016/j.trc.2019.11.003.
[18] Roberti, R., Ruthmair, M. (2021). Exact methods for the traveling salesman problem with drone, *Transportation Science*, Vol. 55, No. 2, 315-335, doi: 10.1287/trsc.2020.1017.
[19] Salama, M.R., Srinivas, S. (2022). Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 164, Article No. 102788, doi: 10.1016/j.tre.2022.102788.
[20] Chen, H., Hu, Z., Solak, S. (2021). Improved delivery policies for future drone-based delivery systems, *European Journal of Operational Research*, Vol. 294, No. 3, 1181-1201, doi: 10.1016/j.ejor.2021.02.039.
[21] Lichau, S., Sadykov, R., François, J., Dupas, R. (2025). A branch-cut-and-price approach for the two-echelon vehicle routing problem with drones, *Computers & Operations Research*, Vol. 173, Article No. 106869, doi: 10.1016/j.cor.2024.106869.
[22] Saleu, R.G.M., Deroussi, L., Feillet, D., Grangeon, N., Quilliot, A. (2022). The parallel drone scheduling problem with multiple drones and vehicles, *European Journal of Operational Research*, Vol. 300, No. 2, 571-589, doi: 10.1016/j.ejor.2021.08.014.
[23] Ermağan, U., Yıldız, B., Salman, F.S. (2022). A learning-based algorithm for drone routing, *Computers & Operations Research*, Vol. 137, Article No. 105524, doi: 10.1016/j.cor.2021.105524.
[24] Stodola, P., Kutěj, L. (2024). Multi-depot vehicle routing problem with drones: Mathematical formulation, solution algorithm and experiments, *Expert Systems with Applications*, Vol. 241, Article No. 122483, doi: 10.1016/j.eswa.2023.122483.
[25] Ramadhan, F., Irawan, C.A., Salhi, S., Cai, Z. (2025). The truck traveling salesman problem with drone and boat for humanitarian relief distribution in flood disaster: Mathematical model and solution methods, *European Journal of Operational Research*, Vol. 322, No. 1, 270-291, doi: 10.1016/j.ejor.2024.10.022.
[26] Gao, X., Jin, X., Zheng, P., Cui, C. (2021). Multi-modal transportation planning for multi-commodity rebalancing under uncertainty in humanitarian logistics, *Advanced Engineering Informatics*, Vol. 47, Article No. 101223, doi: 10.1016/j.aei.2020.101223.
[27] Gao, Y., Ding, X., Yu, W. (2024). Distributional robustness based on Wasserstein-metric approach for humanitarian logistics problems under road disruptions, *Operations Research Perspectives*, Vol. 13, Article No. 100317, doi: 10.1016/j.orp.2024.100317.
[28] Ahmadi, M., Seifi, A., Tootooni, B. (2015). A humanitarian logistics model for disaster relief operations considering network failure and standard relief time: A case study on San Francisco district, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 75, 145-163, doi: 10.1016/j.tre.2015.01.008.
[29] Hou, D.N., Liu, S.C. (2024). Optimization of cold chain multimodal transportation routes considering carbon emissions under hybrid uncertainties, *Advances in Production Engineering & Management*, Vol. 19, No. 3, 315-332, doi: 10.14743/apem2024.3.509.
[30] Yang, X., Cao, W., Wang, K., Yin, H., Wu, J., Wu, L. (2025). Integrated scheduling of truck and drone fleets for cargo transportation in post-disaster relief: A two-stage stochastic optimization approach, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 196, Article No. 104015, doi: 10.1016/j.tre.2025.104015.
[31] Lu, Y., Yang, J., Yang, C. (2023). A humanitarian vehicle routing problem synchronized with drones in time-varying weather conditions, *Computers & Industrial Engineering*, Vol. 184, Article No. 109563, doi: 10.1016/j.cie.2023.109563.
[32] Yin, Y., Yang, Y., Yu, Y., Wang, D., Cheng, T. C. E. (2023). Robust vehicle routing with drones under uncertain demands and truck travel times in humanitarian logistics, *Transportation Research Part B: Methodological*, Vol. 174, Article No. 102781, doi: 10.1016/j.trb.2023.102781.

[33] Kyriakakis, N.A., Sevastopoulos, I., Marinaki, M., Marinakis, Y. (2022). A hybrid Tabu search–variable neighborhood descent algorithm for the cumulative capacitated vehicle routing problem with time windows in humanitarian applications, *Computers & Industrial Engineering*, Vol. 164, Article No. 107868, doi: 10.1016/j.cie.2021.107868.

[34] Van Steenbergen, R., Mes, M., Van Heeswijk, W. (2023). Reinforcement learning for humanitarian relief distribution with trucks and UAVs under travel time uncertainty, *Transportation Research Part C: Emerging Technologies*, Vol. 157, Article No. 104401, doi: 10.1016/j.trc.2023.104401.

[35] Amirsahami, A., Barzinpour, F., Pishvaee, M.S. (2025). A fuzzy programming model for decentralization and drone utilization in urban humanitarian relief chains, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 195, Article No. 103949, doi: 10.1016/j.tre.2024.103949.