

Original Research

Research on Optimization of Transport Path for Novel Coronavirus Detection Samples

Jing Han*, Yanqiu Liu

School of Management, Shenyang University of Technology, Shenyang/110870, China

Received: 15 September 2023

Accepted: 8 November 2023

Abstract

The novel coronavirus is one of the most widespread global epidemics that has harmed human life and health around the world in the past century. Nonetheless, we confront a formidable predicament: the transportation costs and transmission risks of comprehensive testing for the new coronavirus are huge. Furthermore, the extensive utilization of trucks has contributed to a surge in greenhouse gas emissions, thereby exacerbating the pressing issues of global warming and climate change. To ameliorate these challenges, we proffer an ingenious resolution: the deployment of a truck outfitted multiple drones, thereby orchestrating the transfer of samples. In our endeavor to address this intricate issue, we have devised a MIP model, with distribution cost and delivery time serving as bi-objectives. Additionally, we have developed a genetic adaptive large-scale neighborhood search algorithm (GALNS) to resolve the model. Through example testing, we draw the following conclusions: a. We verified the correctness and effectiveness of the proposed model and algorithm. b. In comparison to traditional truck transfer, the deployment of trucks ferrying multiple drones, each equipped with multi-visit functionalities, for the transportation of nucleic acid testing samples, not only proves to be a more cost-effective and efficient approach but also mitigates the risk of contagion.

Keywords: truck and drone, multi-visits, novel coronavirus sample transfer, path planning

Introduction

The eruption of the COVID-19 epidemic has engendered significant public health security perils, representing a grave menace to human well-being and vitality, while also exerting a profound impact on economic and societal spheres [1, 2]. The virus has afflicted humanity through modes of respiratory droplet and contact transmission. To mitigate the proliferation of the virus, the Chinese government has actively carried

out nucleic acid testing [3], which has also spawned research on optimizing the transport path of new coronavirus testing samples. Thus, the investigation into the conveyance path of these samples bears immense practical significance. The sample transfer process commonly entails healthcare personnel collecting samples at diverse testing sites, followed by the imperative dispatch of vehicles from the testing center to gather these samples. However, due to the variability in sample quantities, collection times, and frequencies, this results in an inefficacious utilization of vehicle cargo space and the proliferation of circuitous paths, thereby inflating transshipment costs. Certain testing sites may be inaccessible via vehicular means, necessitating

*e-mail: hanjing@smail.sut.edu.cn

manual transportation, which is not only time-intensive but also heightens the risk of infection. Furthermore, the extensive utilization of trucks has exacerbated the emission of greenhouse gases, consequently inflicting greater damage upon the predicaments of global warming and climate change. Drones have demonstrated merits across a spectrum of domains, encompassing the reduction of operational expenditures, acceleration of transportation velocity, facilitation of contactless delivery, and advocacy for the integration of renewable energy sources [4]. Additionally, owing to its reliance on battery power, the drone system operates with a complete absence of emissions, refraining from the discharge of detrimental gases, thereby contributing to the preservation of the climate and enhancement of air quality. Nevertheless, it is imperative to acknowledge the constraints inherent in their performance with regard to long-haul and large-scale cargo conveyance assignments. Consequently, we proffer an innovative resolution wherein a truck collaboratively transports nucleic acid samples in concert with multiple drones, each endowed with multi-visit capabilities. This approach not only furnishes robust reinforcement in the face of the exigencies presented by this global pandemic but also harmonizes seamlessly with the principles of sustainability and environmental preservation.

Some scholars have proposed research on using trucks and drones to collaborate on package delivery, which has gone through multiple evolutionary stages and variants so far. The Flying Sidekick Traveling Salesman Problem (FSTSP) originally proposed by Murray et al. [5] was the cornerstone of this problem. Subsequently, a profusion of offshoots materialized, among which were TSPD (Traveling Salesman Problem with Drone) [6], MFSTSP (Multiple Flying Sidekicks Traveling Salesman Problem) [7], TSP-mD (Traveling Salesman Problem with Multiple Drones) [8], VRPD (Vehicle Routing Problem with Drone) [9], to name but a few, each tailored to distinct delivery scenarios and endowed with different objective functions. Researchers have predominantly built to formulate mathematical models with delivery time [10] or delivery cost [11] as the centerpiece objective. They have proffered sundry algorithmics to tackle these quandaries, including the column generation algorithm [12], the genetic algorithm [13], the variable neighborhood search algorithm [14], and the hybrid algorithm [15], among others. Furthermore, certain studies have delved into mode of transport, such as truck-supported drone delivery [16], drone-facilitated truck delivery [17], independent truck and drone delivery [18], and hybridized delivery systems [19], to better Adapt to real-world delivery challenges. The above literature can provide methodological reference for this article, but it does not consider the impact of public health emergencies on collaborative delivery path planning of trucks and drones.

The investigation conducted by Barnawi et al. [20] centered on the utilization of drones during the COVID-19 epidemic and introduced a deep learning

model for diagnosing cases. Regrettably, this study omitted the consideration of practical concerns, such as the performance, efficiency, and energy consumption of drones in real-world deliveries. Correspondingly, Yang et al. [21] proposed a truck-supported drone joint delivery model to address the distribution challenges posed by the novel coronavirus epidemic. However, in this delivery model, the truck's cargo capacity remains underutilized, as the truck itself does not directly participate in the delivery service. In this context, Peng et al. [22] conducted research on collaborative delivery of trucks and drones, where trucks directly provide services to customers. They formulated a mixed integer programming model and employed a hybrid neighborhood search algorithm to resolve this conundrum. Wu et al.'s [23] research prioritized the minimization of delivery time and introduced a contactless package delivery conundrum entailing collaborative routing for both trucks and drones. The research by Liu et al. [24] focused on how to supply emergency supplies between high-risk and low-risk epidemic areas after the outbreak of infectious public health events, especially during the COVID-19 epidemic. In order to avoid the spread of the epidemic, they proposed the strategy of using "contactless" drone delivery in high-risk epidemic areas and truck delivery in low-risk epidemic areas. Ji et al. [25] proposed a material distribution method for closed communities based on the collaboration of drones and trucks, focusing on solving the problems of cross-infection and material efficiency. Li et al.'s [26] research centered on contactless medical supplies distribution while also addressing carbon emissions optimization. Finally, Li et al. [27] proposed a two-layer heuristic algorithm to tackle the collaborative delivery conundrum involving trucks and drones. The aforementioned literature serves as a valuable theoretical compass for material distribution within the context of the new coronavirus epidemic.

Although there have been studies focusing on collaborative delivery using trucks and drones, the existing literature has not yet delved into the specific delivery needs of nucleic acid samples. Firstly, this need is pressing as any delays may result in sample failure. Such failures can lead to substantial economic and human costs and even cause social panic. Secondly, this demand carries a high risk of infection and necessitates non-contact methods. Additionally, existing research often does not take into account the actual situation of drones visiting multiple nodes in a single trip and processing nodes whose demand is greater than the drone's load capacity. With advancing technology, we will also investigate how multiple drones can be coordinated with trucks to enhance drone utilization. Simultaneously, we will prioritize two key goals: delivery time and distribution cost. In summary, this study aims to address the urgent delivery of nucleic acid samples. We establish a bi-objective programming model that takes into account the actual situation of

a truck carrying multiple drones and the drones' visits to multiple nodes within one trip. This model is designed to minimize delivery time and costs. To tackle this crucial problem, we have devised a genetic adaptive large-scale neighbor search algorithm (GALNS) to resolve this conundrum, offering valuable insights for decision-making in virus sample transport path planning during major outbreaks.

Literature Review

In this section, we furnish a concise exposition of antecedent research intimately associated with the present paper's undertaking. Should readers desire an exhaustive and in-depth analysis, they are encouraged to peruse the studies authored by Macrina et al. [28] and Madani et al. [29]. According to the number of trucks and vehicle-mounted drones, we can divide the truck and drone collaborative delivery problems into three categories (Table 1): TSPD, TSP-mD and VRPD.

Murry et al. [5] introduced the concept of the FSTSP, which encompasses the collaborative delivery of a truck and a drone. In this context, they imposed a constraint wherein the drone could exclusively serve one customer per flight while optimizing for delivery time. To contend with the inherent NP-hard of this challenge, they formulated a mixed-integer programming model and devised a heuristic algorithm for resolution. Building upon this work, Agatz et al. [6] conceived an alternative model, denoted as TSPD, with the objective of cost-minimization in transportation. They also engineered a path-first clustering-second heuristic algorithm, effectively addressing this problem. Ha et al. [15] focused their attention on the TSPD model with the goal of minimizing operating costs, where operating costs include transportation costs and time costs caused by waiting. Tong et al. [30] harnessed a variable neighborhood search algorithm to tackle the TSPD challenge. It should be pointed out that in the above studies, the drone typically undergoes battery replacement after each flight to facilitate subsequent node services. However, Yurek et al. [31] proposed a different perspective, considering the TSPD problem under charging strategy.

Following this, Phan et al. [32] expanded the purview of the TSP-D problem to encompass real-world distribution scenarios, introducing the concept of a truck equipped with multiple drones for delivery, which they designated as the TSP-mD problem.

Campbell et al. [33] employed continuous approximation modeling techniques to assess a collaborative delivery system involving both trucks and multiple drones. The research findings underscored the economic advantages inherent in TSP-mD. Seifried et al. [34] developed an exact algorithm for resolving this problem and substantiated the NP-hard of TSP-mD. Murray et al. [7] further studied the MFSTSP, correlating the energy consumption model of the drone with the package weight, speed and operation time. It is worth noting that in the above study, the drone could only visit one customer at a time. Subsequently, Gonzalez-R et al. [35] proposed an extension of the TSP-D problem model, permitting drones to serve multiple customers in a single flight. Luo et al. [8] addressed the multi-drone multi-visit traveling salesman problem and harnessed a multi-start tabu search methodology to tackle the challenge, involving up to a hundred customers. Lastly, Mara et al. [36] devised a heuristic algorithm grounded in Adaptive Large-scale Neighborhood Search (ALNS) techniques for the resolution of this problem.

Wang et al. [9] embarked on an expansive exploration of collaborative delivery challenges both trucks and drones, transcending the confines of a single vehicle to encompass multiple vehicles, thereby birthing the VRPD problem. Their research illuminated that, even when each vehicle carries just one drone, and the drones share identical speed and distance matrices with the trucks, this collaborative delivery paradigm can slash completion time in half. Subsequently, some scholars conducted research on exact methods [37, 38] and heuristic algorithms [39, 40] to solve this problem. Some scholars have studied the collaborative delivery problem of trucks and drones in different contexts [23, 26]. In addition, some literature studies the extension of the collaborative delivery problem of trucks and drones by adding constraints. Meng et al. [41] extended the VRPD to a logistics system that incorporates dual requisites: pickup and delivery. They intricately modeled parameters such as drone travel distance and package carrying weight, conceptualizing them as pivotal facets of drone energy consumption models. Subsequently, they formulated a mixed integer programming model, replete with problem-specific inequalities. In addition, they proffered a novel two-stage heuristic algorithm to proficiently tackle this conundrum. Kuo et al. [42] directed their research towards the minimization of travel expenses, focusing on the expansion of the VRPD, this time accounting for temporal constraints in the form of time windows. They ingeniously devised a variable neighborhood search algorithm as a means to resolve this intricate challenge. Jeong et al. [43] studied the impact of package weight on drone flight duration and considered the situation where drone flights are prohibited in some delivery areas during certain periods. Furthermore, Liu et al. [11] innovatively incorporated the concepts of no-fly zones and pickup and delivery tasks into the VRPD framework. Their solution strategy was grounded in a two-stage heuristic algorithm, anchored

Table 1. Categorization of cooperative delivery challenges amidst trucks and drones.

	TSPD	TSP-mD	VRPD
Number of trucks n	$n = 1$	$n = 1$	$m > 1$
Number of vehicle-mounted drones m	$m = 1$	$m > 1$	$m^3 - 1$

in the principles of simulated annealing, offering a resolution to this multifaceted problem.

The preceding literature delves into the examination of collaborative truck and drone delivery from various perspectives and problem scenarios. While these investigations have advanced our understanding, there remains ample space for further exploration. Despite the proposal of theoretical frameworks within these studies, they must still account for the intricacies of real-world delivery operations. Moreover, it is imperative to acknowledge that this predicament is formally established as NP-Hard, necessitating additional research to devise more efficient algorithms, particularly in addressing expansive-scale challenges.

Problem Description

Given a delivery network $G=(V,A)$, $V=\{0,1,2,\dots,n,n+1\}$ represents the node set, node $0,n+1$ represents the inspection center, $C=\{1,2,\dots,n\}$ represents the detection node set; A represents the arc set, $A=\{(i,j)|i\in V,j\in V,i\neq j\}$. $U=\{1,2,\dots,u\}$ represents the collection of drones, and $P=\{1,2,\dots,p\}$ represents the collection of drone trips. The location of each detection node is known, along with the sample collection requirements, which are determined based on the number of samples that need to be collected. When orchestrating the conveyance of these samples, the formidable challenge at hand lies in the judicious planning of synergistic transport paths for both trucks and multiple drones, each endowed with multi-visit capabilities. This endeavor is aimed at augmenting sampling efficiency, mitigating the peril of contagion, and curtailing transportation costs. This study christens this intricate quandary as the Multi-Visit Multiple Flight Assistant Traveling Salesman Problem (MV-MFSTSP).

The workflow is as follows: a fleet, comprising a truck and a cohort of multi-visit drones, commences its journey from the inspection center denoted as ‘0’, methodically collecting samples at diverse detection nodes. It is imperative that each detection node receives service either from a truck or a drone, and upon the culmination of all tasks, the fleet is obliged to return to the inspection center. To facilitate management, the truck is bound to the drone it carries, i.e. they move together (Fig. 1). The drone is afforded the flexibility to launch and land either at the detection node or the inspection center. Following the completion of the sample collection task at each detection point, the drone must fulfill two prerequisites before proceeding to the subsequent detection point: its remaining cargo capacity must meet or exceed the sample collection volume requisite for the next point, and its remaining flight distance must meet or exceed the remaining flight distance for the ensuing detection point. If the

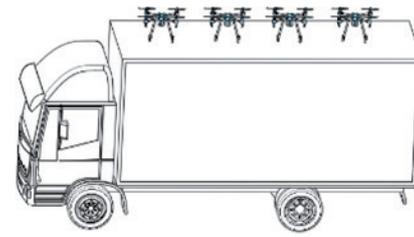


Fig. 1. Model drawings of trucks and drones.

drone fails to meet these criteria, it shall return to the nearest accessible truck parking location to await landing. Meanwhile, while the drone executes its mission aloft, the truck continues its preordained path, ensuring the uninterrupted delivery of sample packages. Furthermore, to forestall cross-contamination, it is imperative that samples undergo requisite disinfection both before collection and after delivery.

The assumptions of the MV-MFSTSP problem are as follows:

The drone traverses nearly linear trajectories, prompting its adherence to the Euclidean distance metric; while the truck needs to travel along the road network, the truck distance is defined as $d_{ij}=\eta d_{ij}$, η is a constant that represents the curvature of the road.

Both the truck and the drone always keep moving at a constant speed, and the drone is faster than the truck.

The impact of the drone’s flight speed and load capacity on the cruising range is not considered.

In the event that either the truck or the drone arrives ahead of schedule, they must await the other’s arrival before recommencing the delivery process.

No consideration is given to the time required for battery replacement when their energy levels are depleted.

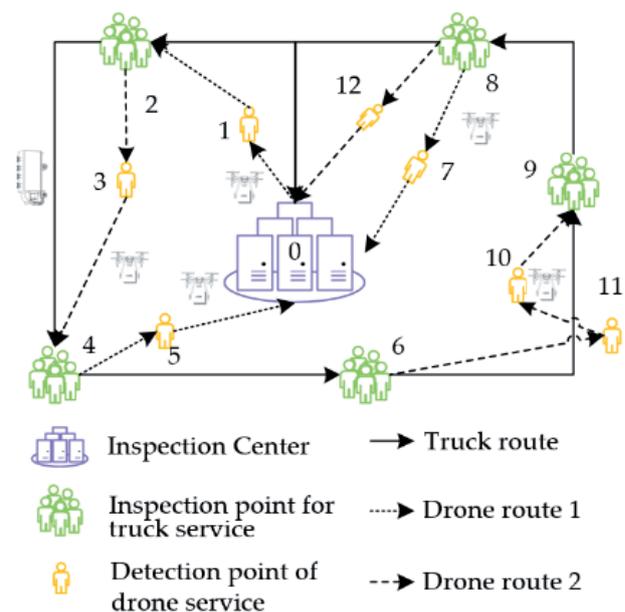


Fig. 2. A possible solution for MV-MFSTSP.

The truck possesses the capacity to transport all available drones in a single haul, and the quantity is contingent upon the volume of samples gathered at the detection node.

In this paper, MV-MFSTSP is assumed to be explored in two-dimensional (2D) space.

Fig. 2 illustrates a plausible path for a truck transporting multiple multi-visit drones collaboratively engaged in sample transportation. Further symbols and explanations pertinent to the mathematical model can be found in Table 2.

Before establishing a mathematical model, first divide the detection nodes as follows:

N^{ku} : detection nodes in this set are served by trucks,
 $N^k = \{i \mid q_i > Q \parallel d_{ij} > D_{\max} / 2, \forall i, j \in C\}$;

N^k : detection nodes in this set can be served by both drones and trucks, $N^{ku} = C - N^k$.

Mathematical Model

$$\min f_1 = C_1 \sum_{i \in V'} \sum_{j \in V'} d'_{ij} x_{ij} + C_2 \sum_{u \in U} \sum_{p \in P} \sum_{i \in V'} \sum_{j \in V'} d_{ij} y_{ij}^{up} \tag{1}$$

$$\min f_2 = \max \{t_{n+1}\} \tag{2}$$

S.T.

$$\sum_{u \in U} \sum_{p \in P} Z_i^{up} + Z_i = 1, \forall i \in N^{ku} \tag{3}$$

$$Z_i = 1, \forall i \in N^k \tag{4}$$

$$\sum_{j \in V'} y_{ij}^{up} \geq Z_i^{up}, \forall i \in V, p \in P, u \in U \tag{5}$$

Table 2. Model symbols and explanations.

Type	Symbol	Explanation
Parameter	e	Sample validity period.
	M	A very large positive integer.
	R	Maximum number of drones that can be carried on a truck.
	D_{\max}	The maximum flight distance of the drone. Because the drone serves at least one detection node each time it flies, the distance to the farthest detection node that the drone can visit is $D_{\max} / 2$.
	C_1, C_2	Truck/drone cost per unit transportation distance.
	S_i	Service time of detection node $i \in C$.
	q_i	Sample weight of detection node $i \in C, q_i = l_i c, l_i$ represents the number of samples at detection node i, c is the weight of a single sample.
	Q	Maximum load capacity of drone.
	V_1, V_2	The speed of truck/drone transportation.
	d'_{ij}, d_{ij}	Transportation distance of truck/drone from node $i \in V'$ to node $j \in V$.
	t'_{ij}, t''_{ij}	Transportation time of truck/drone from node $i \in V'$ to node $j \in V$.
Variable	$x_{ij}(=1)$	0-1 variable, which indicates that the truck travels from node $i \in V'$ to node $j \in V$.
	$y_{ij}^{up}(=1)$	0-1 variable, which indicates that the u -th ($u \in U$) drone's p -th ($p \in P$) trip along arc $(ij) \in A$.
	$h_{iup}^S(=1)$	0-1 variable, which indicates that node $i \in V$ is the launch node of the p -th ($p \in P$) trip of the u -th ($u \in U$) drone, and S is the abbreviation of "Start".
	$h_{iup}^E(=1)$	0-1 variable, which indicates that node $i \in V$ is the landing node of the p -th ($p \in P$) trip of the u -th ($u \in U$) drone, and E is the abbreviation of "End".
	$Z_i^{up}(=1)$	0-1 variable, which indicates that node $i \in V$ is served by the p -th ($p \in P$) trip of the u -th ($u \in U$) drone.
	$Z_i(=1)$	0-1 variable, which indicates that node $i \in V$ is served by a truck.
	t'_i	Continuous variable, which represents the time when the truck arrives at node $i \in V$.
	t_i^{up}	Continuous variable, which represents the time when the p -th ($p \in P$) trip of the u -th ($u \in U$) drone reaches node $i \in V$.
	g_i	Continuous variable, which represents the truck access order of node $i \in V$.
	k_i	Continuous variable, which represents the drone access order of node $i \in V$.
	r_i	Continuous variable, which represents the number of drones that are not on the truck when departing from node $i \in V$.
	t_i	Continuous variable, which represents the actual time of arrival at node $i \in V$.

$$Z_i = \sum_{j \in V} x_{ji}, \forall i \in V \tag{6}$$

$$h_{iup}^E + \sum_{j \in V} y_{ij}^{up} = h_{iup}^S + \sum_{j \in V} y_{ji}^{up}, \forall i \in V, p \in P, u \in U \tag{7}$$

$$\sum_{j \in V} y_{ij}^{up} \leq 1, \forall i \in V, p \in P, u \in U \tag{8}$$

$$\sum_{j \in V} y_{ji}^{up} \leq 1, \forall i \in V, p \in P, u \in U \tag{9}$$

$$\sum_{j \in V} x_{ij} = \sum_{j \in V} x_{ji} \leq 1, \forall i \in V \tag{10}$$

$$\sum_{u \in U} \sum_{p \in P} y_{ij}^{up} + \sum_{u \in U} \sum_{p \in P} y_{ji}^{up} \leq 1, \forall (i, j) \in A \tag{11}$$

$$Z_i^{up} + Z_j^{up} \geq 1 - M(1 - y_{ij}^{up}), \forall (i, j) \in A, p \in P, u \in U \tag{12}$$

$$h_{iup}^S \geq 1 - M(2 - y_{ij}^{up} - Z_j^{up} + Z_i^{up}), \forall (i, j) \in A, p \in P, u \in U \tag{13}$$

$$h_{iup}^E \geq 1 - M(2 - y_{ij}^{up} - Z_i^{up} + Z_j^{up}), \forall (i, j) \in A, p \in P, u \in U \tag{14}$$

$$\sum_{u \in U} \sum_{p \in P} h_{iup}^S \leq M \sum_{j \in V} x_{ji}, \forall i \in V \tag{15}$$

$$\sum_{u \in U} \sum_{p \in P} h_{iup}^E \leq M \sum_{j \in V} x_{ij}, \forall i \in V \tag{16}$$

$$h_{iup}^S + h_{iup}^E \leq 1, \forall i \in V, p \in P, u \in U \tag{17}$$

$$\sum_{i \in V} h_{iup}^S = \sum_{i \in V} h_{iup}^E \leq 1, \forall p \in P, u \in U \tag{18}$$

$$g_i - g_j \geq 1 - M(1 - x_{ij}), \forall i \in C, j \in V, i \neq j \tag{19}$$

$$k_i - k_j \geq 1 - M(1 - y_{ij}^{up}), \forall i \in C, j \in V, i \neq j, p \in P, u \in U \tag{20}$$

$$t_i = \max\{t_i^{u_max}, t_i'\}, \forall i \in V, u_max \in U, p \in P \tag{21}$$

$$t_i^{u_max} = \max\left\{\sum_{p \in P} t_i^{1,p} Z_i^{1,p}, \sum_{p \in P} t_i^{2,p} Z_i^{2,p}, \dots, \sum_{p \in P} t_i^{u,p} Z_i^{u,p}\right\}, \forall i \in V, u_max \in U, u \in U, p \in P \tag{22}$$

$$t_i^{u,p} \geq t_{i-1} + t_{i-1,i} + S_{i-1} - M(1 - y_{i-1,i}^{up}), \forall i \in V, p \in P, u \in U \tag{23}$$

$$t_i' \geq t_{i-1} + t_{i-1,i}' + S_{i-1} - M(1 - x_{i-1,i}), \forall i \in V \tag{24}$$

$$t_{ij} = d_{ij} / V_2, \forall (i, j) \in A \tag{25}$$

$$t_{ij}' = d_{ij}' / V_1, \forall (i, j) \in A \tag{26}$$

$$\sum_{p \in P} \sum_{u \in U} \sum_{j \in C} y_{0j}^{up} + r_0 = R \tag{27}$$

$$\sum_{p \in P} \sum_{u \in U} \sum_{i \in V} y_{ii}^{up} + r_i \geq \sum_{p \in P} \sum_{u \in U} \sum_{h \in V} y_{jh}^{up} + r_j - M(1 - x_{ij}), \forall i \in C, j \in C, i \neq j \tag{28}$$

$$\sum_{p \in P} \sum_{u \in U} \sum_{i \in V} y_{ji}^{up} + r_j \geq R - M(1 - x_{j,(n+1)}), \forall j \in C \tag{29}$$

$$\sum_{i \in N^{ku}} q_i Z_i^{up} \leq Q, \forall p \in P, u \in U \tag{30}$$

$$\sum_{i \in V} \sum_{j \in V} d_{ij} y_{ij}^{up} \leq D_{max}, \forall p \in P, u \in U \tag{31}$$

$$x_{ij}, y_{ij}^{up}, Z_i, Z_i^{up}, h_{iup}^E, h_{iup}^S \in \{0, 1\}, \forall i \in V, j \in V, i \neq j, p \in P, u \in U \tag{32}$$

$$1 \leq g_i \leq n + 2, 1 \leq k_i \leq n + 2, t_i \geq 0, t_i' \geq 0, t_i^{up} \geq 0, 0 \leq r_i \leq R, \forall i \in V, p \in P, u \in U \tag{33}$$

This study considers the dual optimization objectives of the transportation cost and the maximum allowable return time to the inspection center, as expressed in formulas (1-2). Constraint (3) means that each detection node can only be provided with a collection service by a truck or drone. Constraint (4) ensures that the detection points in the set N^k can only be served by trucks. Constraints (5-6) give the relationship between variables. Constraint (7) provides a traffic balance equation for the nodes visited during the drone trip, which can handle launch and landing nodes. Constraints (8-9) provide the total in-degree and total out-degree of the drone for each detection node to ensure that each detection node is only served once. Constraint (10) not only maintains the flow balance of trucks, but also ensures that trucks can visit a detection node at most once. Constraint (11) requires that each edge be visited by a drone at most once. Constraint (12) requires that each edge visited by the drone has at least one node service during this drone trip to prevent the drone from flying along the truck's path unnecessarily. Constraints (13-14) determine the launch and landing nodes of the drone trip, respectively. Constraints (15-16) ensure that the launch and landing nodes of the drone must be visited and served by the truck. Constraint (17) ensures that the launch and landing nodes of the drone cannot be the same node. Constraint (18) ensures that the

number of launch and landing nodes in each drone trip is equal, and there is at most one launch node in each drone trip. Constraints (19-20) represent subtours that eliminate the paths of trucks and drones. Constraint (21) is to calculate the actual time to reach node i . Constraint (22) represents the latest time for all drones to arrive at node i . Constraints (23-24) calculate the time for drones and trucks to arrive at node i , respectively. Constraints (25-26) calculate the transportation time of drones and trucks from node i to node j , respectively. Constraint (27) means calculating the number of drones on the truck when it leaves the inspection center. Constraint (28) means that if the truck wants to go from detection node i to j , then the sum of the number of drones on the truck when leaving node i and the number of drones from node i to other nodes is equal to the number of drones on the truck when leaving node j the sum of the number and the number of drones from node j to other nodes. Constraint (29) is a special case of constraint (28) at the inspection center. Constraint (30) represents the load constraint of the drone. Constraint (31) is the maximum flight distance constraint of the drone. Constraints (32-33) represent the value range of all variables.

Solution Approach

Main Objective Method

Contingent upon the specific circumstances of the issue at hand, it becomes imperative to ensure the timely conveyance of test samples to the inspection center, adhering to their stipulated validity periods. Consequently, this study adopts the main objective method for addressing the bi-objective programming model. In this model, we designate objective function (1) as the principal aim, relegating objective function (2) to the status of a constraint. By doing so, we effectuate a transformation of the original bi-objective programming model into a single-objective programming model. To confine the ultimate timeframe for the fleet's return to the inspection center, we refine objective function (2) into formula (34).

$$t_{n+1} \leq e \tag{34}$$

Genetic Adaptive Large-scale Neighborhood Search Algorithm

Algorithm Framework

Murry et al.'s [7] research shows that the MFSTSP problem is NP-hard, so the MV-MFSTSP problem is also NP-hard. The ALNS algorithm has been widely used to solve NP-hard problems [36], and the operators of ALNS are customizable. Hence, it can specifically optimize the paths for trucks and drones to work together. Considering that the precision of solution

and the pace of convergence in ALNS are notably influenced by the initial solution, we advocate for the adoption of a "sort first, then group" approach to constructing the initial solution, which leverages genetic algorithms. Furthermore, to mitigate the risk of ALNS getting trapped in local optima and to enhance algorithmic efficiency, we incorporate the Metropolis acceptance criterion from Simulated Annealing (SA) into ALNS. This refined algorithm is denominated as the Genetic Adaptive Large-Scale Neighborhood Search Algorithm (GALNS) [44].

The fundamental concept underlying the Metropolis acceptance criterion is to probabilistically embrace new solutions during each iteration, thus circumventing entrapment within a local optimum [45]. The probability associated with accepting these new solutions is determined by Formula (35):

$$P = \begin{cases} 1, & f(S_{new}) < f(S_{curr}), \\ e^{-[f(S_{new})-f(S_{curr})]/T}, & f(S_{new}) \geq f(S_{curr}). \end{cases} \tag{35}$$

where, $f(S_{curr})$ represents the objective function value of the current solution, and $f(S_{new})$ represents the objective function value of the new solution. As the number of iterations increases, the probability of acceptance will definitely decrease. This is because the search time needs to be reduced, that is, the temperature T decreases as the number of iterations increases. The formula is as follows:

$$T_{gen+1} = \alpha T_{gen} \tag{36}$$

where, α is the cooling factor, $0 < \alpha < 1$.

This study formulates three sets of operator pairs and employs the roulette principle for their selection. These operator pairs individually undertake the task of destroy and repair of the existing solution, guided by the principles of "randomness," "correlation," and "whole path reconstruction." Such an approach not only broadens the domain of viable solutions but also contributes to an enhanced level of solution precision. The probability associated with the selection of an operator pair is delineated in Formula (37).

$$P_j = w_j / \sum_{k=1}^{|\Omega|} w_k, j = 1, 2, \dots, |\Omega| \tag{37}$$

where, w_j represents the weight of the operator to j , and $|\Omega|$ represents the total number of operator pairs.

The weighting of operator pairs undergoes dynamic adjustments in response to performance score fluctuations throughout the iterative process. If the new solution S_{new} , following the operator pair exploration, ascends as the fresh global optimum solution, it shall receive a points. When the new solution S_{new} is better than the current solution S_{curr} , b points are given; when

the new solution S_{new} is worse than the current solution S_{curr} but accepted by the acceptance criterion, c points are given, otherwise, d points are given, where $a^3 b^3 c^3 d$. The weight w_j undergoes updating according to the formula (38).

$$w_j = \beta w_j^0 + (1 - \beta)(\Psi_j / \Pi_j), j = 1, 2, \dots, |\Omega| \quad (38)$$

where, β represents the reaction parameter, $\beta \in [0, 1]$. w_j^0 signifies the historical weight. Ψ_j denotes the score corresponding to operator pair j , and Π_j reflects the frequency of utilization of operator pair j .

The framework of GALNS is shown in Fig. 3.

Construct Initial Solution

The dimensions and length of the solution to the MV-MFSTSP problem are not predetermined, rendering the conventional approach to initial solution construction impractical. Consequently, this study introduces a novel “sort first, then group” method for the initial solution construction, leveraging genetic algorithms. The fundamental concept is as follows: Firstly, employ a genetic algorithm to ascertain the optimal truck delivery paths for all nodes. Subsequently, categorize all nodes into two groups: one comprises

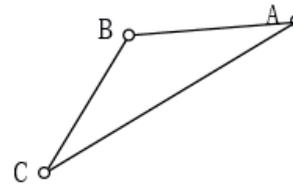


Fig. 4. Demonstration diagram of a simple path.

nodes amenable to both drones and trucks, while the other consists of nodes exclusively suitable for truck service. Thereafter, eliminate nodes of the first category from the truck path, followed by determining the nodes to be serviced by drones based on Formula (39). This formula accounts for the overall transportation cost when distinct vehicle types serve each node. In essence, we eliminate the most expensive nodes for drone delivery and insert these omitted nodes into the most economical positions along the truck’s path. This step expedites the search for the optimal solution during subsequent optimization processes. Lastly, we sequentially insert the omitted nodes at positions where the increase in total transportation cost is minimal. Following the insertion of each node, the algorithm scrutinizes whether the present path complies with the drone’s load capacity, endurance, and sample validity

Algorithm 1	GALNS
Input:	An initial solution S_0 ; Initial temperature of simulated annealing T_0 ; Cooling factor α ; Final temperature T_{min} ; Maximum iterations $MAXGEN$; Reaction parameters β ; Initial weight of destroy and repair operators w ; Initial score of destroy and repair operators $DRscore$; Initial use times of destroy and repair operators $DRuseTime$
Output:	Optimal solution S_{best}
1:	Initialization: $S_{best} \leftarrow S_0, S_{curr} \leftarrow S_0, gen \leftarrow 1, T \leftarrow T_0$
2:	while $gen \leq MAXGEN$ do
3:	while $T > T_{min}$ do
4:	Select destroy and repair operator $DR \in \Omega$ using w
5:	$S_{new} = DR(S_{curr})$
6:	if $f(S_{new}) \leq f(S_{curr})$ then
7:	$S_{curr} \leftarrow S_{new}$
8:	if $f(S_{new}) \leq f(S_{best})$ then
9:	$S_{best} \leftarrow S_{new}$
10:	$DRscore = DRscore + a$
11:	else
12:	$DRscore = DRscore + b$
13:	end if
14:	else
15:	if $Random([0,1]) < \exp(-(f(S_{new}) - f(S_{curr}))/T)$ then
16:	$S_{curr} \leftarrow S_{new}$
17:	$DRscore = DRscore + c$
18:	Else
19:	$DRscore = DRscore + d$
20:	end if
21:	end if
22:	Update w using formula (38)
23:	$T = \alpha \cdot T$
24:	end while
25:	$gen = gen + 1$
26:	end while

Fig. 3. GALNS algorithm framework.

constraints. Subsequently, it reevaluates the cost associated with each uninserted node when placed at various positions within the current path. If during the process of inserting nodes, there is a path that does not satisfy the constraints, then the algorithm will add a new drone path or add the node to the truck path to ensure that all nodes are eventually inserted into the path. As illustrated in Fig. 4, we can derive Formula (39).

$$f(B) = \begin{cases} (|AB| + |BC|) \times \eta \times C_1 \geq (|AB| + |BC|) \times C_2 + |AC| \times \eta \times C_1, & \text{Node B is serviced by drone,} \\ \text{Other,} & \text{Node B is serviced by truck.} \end{cases} \quad (39)$$

Pursuant to Formula (39), we can deduce a lemma that will provide useful help for guiding parameter selection.

Lemma: When $\eta \geq \frac{AB + BC}{AB + BC - AC} \times \frac{C_2}{C_1}$ or $\frac{C_2}{C_1} \geq \frac{AB + BC}{(AB + BC - AC) \times \eta}$ is established, node B can save transportation costs by choosing drone service.

Operator Design

(1) *randDestory* and *randRepair*

Operator *randDestory* represents randomly deleting a node from the current solution, while operator *randRepair* represents inserting the deleted node into a position that minimizes the increase in transportation cost. Its calculation method is shown in formula (40).

$$\min_{r \in \text{removed}} \Delta f_r^i = f_r^i - f_0 \quad (40)$$

where, f_0 represents the transportation cost of the destroyed solution, f_r^i represents the transportation cost of node r at the i -th position, Δf_r^i represents the transportation cost increment of node r inserted into

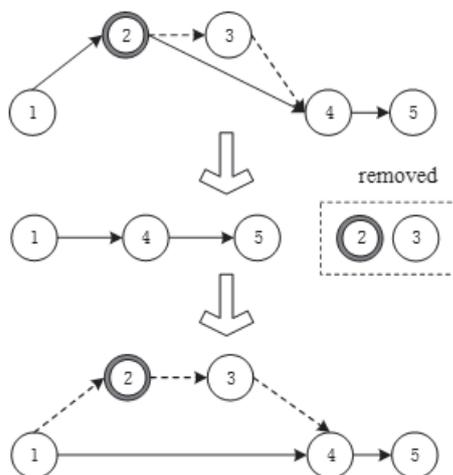


Fig. 5. *randDestory* and *randRepair*.

position i , and *removed* represents the set of removed nodes.

Taking Fig. 5 as an example, we first randomly selected the drone launch node 2 as the node to be removed. In this operation, since the drone path where node 2 is located is still serving node 3, we need to remove node 3 as well and store them as a set in the node set to be removed. Then, according to the calculation method of formula (39), we sequentially insert node 2 and node 3 into the position that minimizes the increase in transportation cost.

(2) *relatedDestory* and *relatedRepair*

Operator *relatedDestory* refers to removing several related nodes according to formula (41-42).

$$R(i, j) = 1 / (\bar{d}_{ij} + V_{ij}), \forall i \in C, j \in C, i \neq j \quad (41)$$

$$\bar{d}_{ij} = d_{ij} / \max d_{ij}, (\max d_{ij} \neq 0) \quad (42)$$

where, \bar{d}_{ij} is the standardized value of d_{ij} , $\bar{d}_{ij} \in [0, 1]$. $V_{ij} = 1$ indicates that node i and node j are not on the same path, otherwise $V_{ij} = 0$. When $R(i, j)$ is larger, the correlation between node i and node j is greater.

The operation of operator *relatedRepair* is to try to re-insert the removed node to the position with the largest regret value while satisfying the conditions of load capacity, endurance, and sample validity. The regret value refers to the cost difference between the node's optimal insertion position and the sub-optimal insertion position in the truck and drone paths, and its calculation method is as shown in formula (43).

$$\max_{r \in \text{removed}} H_r = \Delta f_r^{l2} - \Delta f_r^{l1} \quad (43)$$

where, Δf_r^i represents the transportation cost increment when node r is inserted into the position i , and $l1$ and $l2$ respectively represent the minimum and sub-minimum positions of the transportation cost increment when node r is inserted into the destroyed solution. The regret value H_r represents the cost difference of inserting node r into the destroyed solution.

Taking Fig. 6 as an example, we first selected the drone launch node 3 and landing node 4 as the nodes to be removed based on formula (41). In this operation, node 7 in the path also needs to be removed and stored in the set of nodes to be removed. Then, we will sequentially insert the nodes in the node set to be removed to the position with the largest regret value according to formula (43).

(3) *wholeUavpathDestory* and *wholeUavpathRepair*

Operator *wholeUavpathDestory* means to randomly delete a drone path from the current solution, while operator *wholeUavpathRepair* will sequentially insert the drone service nodes in the deleted drone path to the position with the smallest transportation cost increment.

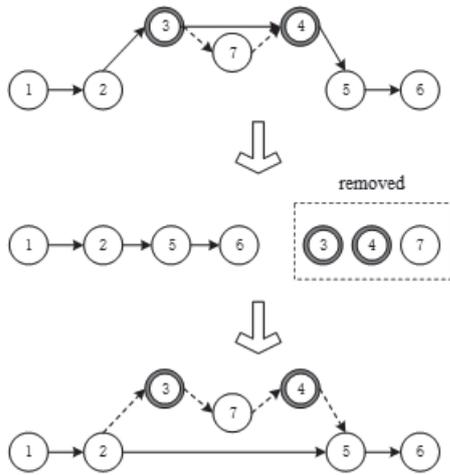


Fig. 6. *relatedDestory* and *relatedRepair*

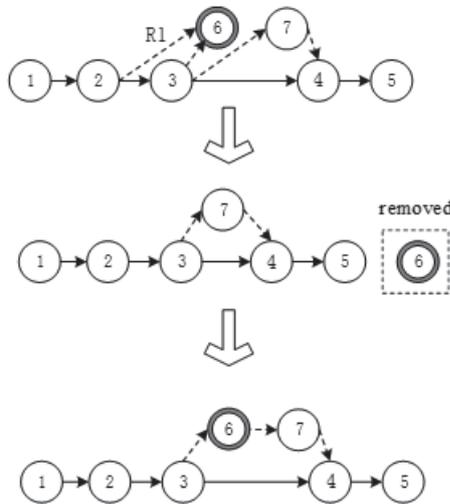


Fig. 7. *wholeUavpathDestory* and *wholeUavpathRepair*.

Taking Fig. 7 as an example, we first randomly selected a drone path R1 for deletion, and stored the node 6 served by the drone in the removed node set. Next, we insert node 6 into the appropriate location one by one according to the principle of minimum transportation cost increment.

Experiment

Because MV-MFSTSP is an extended problem of the MFSTSP, this study chooses to use the test instance of the MFSTSP [7] to evaluate the performance of our proposed model and algorithm. All test cases featured within this paper can be accessed online at the following link: <https://github.com/optimizerlab/mFSTSP>. We employ Gurobi (version 9.1.2) for resolving the MIP model, utilizing Python (version 3.9) as the programming language, and configuring a maximum runtime constraint of 3600 seconds. For algorithms such as GALNS, we utilize Matlab (version 2018b) for implementation, imposing a maximum iteration limit of 200. All coding endeavors are conducted on a server furnished with an Intel(R) Core (TM) i5-10200H CPU @ 2.40GHz. Referring to the performance parameters of multi-rotor drones [46] and the parameters of standard nucleic acid detection samples [4], the parameters are set as shown in Table 3.

Comparative Analysis of Small-Scale Instances

Owing to the inherent intricacy of the MV-MFSTSP problem, the exact solver can exclusively procure optimal solutions for small-scale instance. Consequently, we confine the node count in our instances to a maximum of 20 nodes. We then apply both the GALNS algorithm and the Gurobi solver to these instances, yielding a comparative analysis presented in Table 4. Regarding the Gurobi’s solution, this table documents the upper and lower bounds (Obj_UB, Obj_LB) pertaining to the optimal solution value, alongside its corresponding runtime (Time) and disparity between them ($GAP = (Obj_UB - Obj_LB)/Obj_UB * 100\%$). Regarding the GALNS solution, this table records the optimal solution value (Obj) and the computation time (Time) obtained via GALNS, while denoting the disparity between them as $GAP_1 = (Obj_UB - Obj)/Obj_UB * 100\%$. The test instances are categorized based on their node quantities (Num), and Table 4 documents the corresponding mean values (AVG.) for each group of test instances.

According to the results, the MV-MFSTSP challenge can be optimally addressed employing the Gurobi solver for instances featuring a modest scale of no more than

Table 3. GALNS parameters.

Parameter	Value	Parameter	Value	Parameter	Value
$NIND$	50	Q_u	15 lb	V_u	0.75 km/min
Pc	0.9	D_{max}	20km	C_u	0.078
Pm	0.05	α	0.97	V_k	0.5 km/min
ζ	100	β	0.5	C	0.78
T_{min}	10	a, b, c, d	1.5, 1.2, 0.8, 0.6	η	2
T_0	1000	s	3 min	/	/

10 detection nodes. However, as the number of nodes increases to 15 (20), the optimal solution cannot be obtained within 3600 seconds. On average, the gap between the upper and lower bounds of the solution is 18% (41%). This shows that as the node size increases, the solver performance decreases drastically and more computation time is required. In contrast, the GALNS algorithm exhibits commendable performance not only for smaller-scale predicaments but also maintains a minimal disparity with Gurobi's solutions when tackling 15 and 20-node challenges. On average, GALNS demonstrates efficiency in terms of computation time and solution quality for cases involving 5 and 10 nodes. Nevertheless, when the node count escalates to 15 and 20, GALNS performance exhibits a slight reduction, yet it continues to boast relatively expeditious computation times. In general, it is necessary to design a heuristic algorithm for MV-MFSTSP to solve.

Comparative Analysis of Medium-Scale and Large-Scale Instances

The ALNS algorithm is commonly employed for addressing the collaborative delivery path optimization problem concerning trucks and drones [36]. Hence, to assess the algorithm's efficacy, this paper applies both the GALNS and ALNS algorithms to medium- and large-scale instances, subsequently compares their solution results (Table 5). Table 5 shows the name of the calculation example (Instance), the number of nodes (Num), the optimal solution value of the algorithm (Obj_{ALNS} and Obj_{GALNS}) and the difference value between them. The computational formula is denoted as $GAP_2 = (Obj_{ALNS} - Obj_{GALNS}) / Obj_{ALNS} * 100\%$. To reduce the impact of chance, the LNS and GALNS algorithms were executed 20 times for each computational instance to ascertain the minimum value. These test instances are categorized based on their node quantities,

Table 4. Small-scale instance result comparison.

Instance	Num	Gurobi				GALNS		GAP ₁
		Obj_UB	Obj_LB	GAP	Time	Obj	Time	
20170606T113038113409	5	33.18	33.18	0	1	33.18	0.23	0
20170606T113251786976	5	28.24	28.24	0	1	28.24	0.23	0
20170606T113339368121	5	43.38	43.38	0	1	43.38	0.19	0
20170606T113427164164	5	31.95	31.95	0	1	31.95	0.21	0
20170606T113515209066	5	23.35	23.35	0	1	23.35	0.26	0
AVG.	5	32.02	32.02	0	1	32.02	0.22	0
20170606T113038113409	10	36.33	36.33	0	10	36.33	0.21	0
20170606T113251786976	10	51.84	51.84	0	205	51.84	0.22	0
20170606T113339368121	10	30.57	5.57	0.82	3600	30.57	0.62	0
20170606T113427164164	10	38.45	38.45	0	1171	38.45	0.29	0
20170606T113515209066	10	83.85	83.85	0	15	83.85	0.3	0
AVG.	10	48.21	43.21	0.16	1000.2	48.21	0.33	0
20170606T113038113409	15	55.86	53.1	0.05	3600	53.58	0.32	0.04
20170606T113251786976	15	49.39	45.81	0.07	3600	46.85	0.58	0.05
20170606T113339368121	15	47.44	43.64	0.08	3600	48.4	1.01	-0.02
20170606T113427164164	15	21.22	7.27	0.66	3600	19.35	0.25	0.09
20170606T113515209066	15	86.94	85.5	0.02	3600	87.25	0.25	0
AVG.	15	52.17	47.06	0.18	3600	51.09	0.48	0.02
20170606T113038113409	20	81.13	57.53	0.29	3600	83.28	0.72	-0.03
20170606T113251786976	20	93.3	54.54	0.42	3600	94.24	0.74	-0.01
20170606T113339368121	20	79.05	62.13	0.21	3600	81.15	0.61	-0.03
20170606T113427164164	20	92.19	31.87	0.65	3600	90.85	1.01	0.01
20170606T113515209066	20	93.08	50.7	0.46	3600	103.49	0.71	-0.11
AVG.	20	87.75	51.35	0.41	3600	90.60	0.76	-0.03

Table 5. Solution results of different algorithms.

Instance	Num	Obj_{ALNS}	Obj_{GALNS}	GAP_2
20170606T114511221132	50	123.75	117.49	0.05
20170606T114654882472	50	128.42	124.06	0.04
20170606T114840930461	50	202.6	168.97	0.20
20170606T115303341654	50	99.90	99.90	0.00
20170606T115437348436	50	143.05	123.48	0.16
AVG.	50	139.54	126.78	0.10
20170606T115823934453	100	244.41	215.89	0.13
20170606T120227545709	100	220.15	177.58	0.24
20170606T121241353494	100	252.91	201.36	0.26
20170606T121632081849	100	302.67	247.54	0.22
20170606T122019874088	100	236.13	179.33	0.32
AVG.	100	251.25	204.34	0.23

and Table 5 catalogues the corresponding mean values (AVG) for each cohort of test instances.

In the case of the 50-node instance, the ALNS algorithm yields an average optimal solution of 139.54, whereas the GALNS algorithm attains an average optimal solution of 126.78, with a mere 0.10 average disparity between them. As the node count escalates to 100, the ALNS algorithm’s average optimal solution stands at 251.25, whereas the GALNS algorithm achieves an average optimal solution of 204.34, resulting in an average difference of 0.23. In summation, when the node count remains below 100, the GALNS algorithm outperforms the ALNS algorithm. This superiority arises from the GALNS algorithm’s optimization grounded in a superior initial solution and the inclusion

of a simulated annealing acceptance criterion, which effectively circumvents local optima, thereby enhancing the algorithm’s performance.

Comparative Analysis of Different Transport Modes

If the inspection center exclusively employs trucks for operations, this mode is referred to as the traditional truck transfer mode (Truck). Conversely, when the inspection center utilizes a truck carry out multiple drones, each endowed with multi-visit capabilities, operating in tandem, this mode is designated as the truck and drone cooperative transfer mode (Truck + Drone). To delve deeper into the merits and

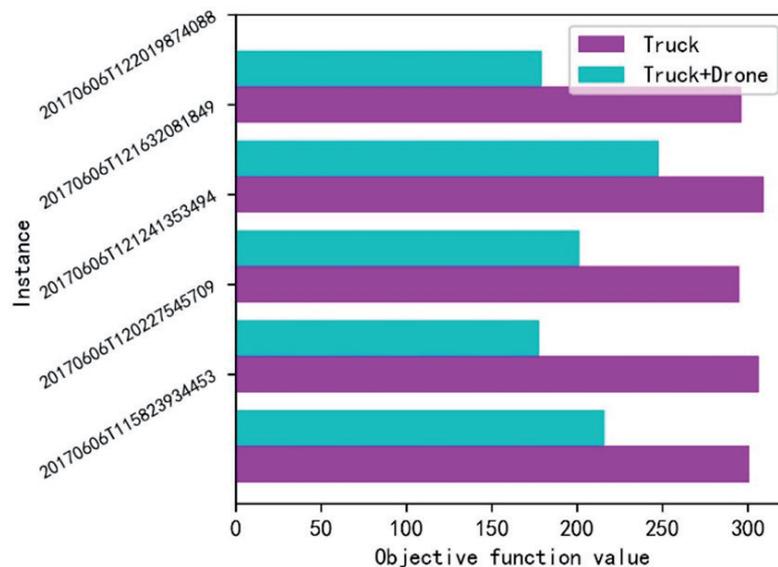


Fig. 8. Comparison chart of different working modes.

Table 6. Comparison results of different working modes.

Instance	Num	Truck	Truck + Drone	GAP ₃
20170606T115823934453	100	300.87	215.89	0.39
20170606T120227545709	100	306.29	177.58	0.72
20170606T121241353494	100	294.93	201.36	0.46
20170606T121632081849	100	309.04	247.54	0.25
20170606T122019874088	100	296.43	179.33	0.65
AVG.	100	301.51	204.34	0.50

demerits of these two operational modes, we applied them to five test instances comprising 100 nodes. The comparative outcomes are delineated in Table 6 and Fig. 8. Additionally, we selected the computation instance “20170606T115823934453” to illustrate the optimal route for traditional truck transfer (Fig. 9)

and the optimal route for truck and drone cooperative transfer (Fig. 10).

The findings demonstrate that the synergy between trucks and drones in the transport paradigm confers distinct competitive advantages when compared to the traditional truck-based transportation model,

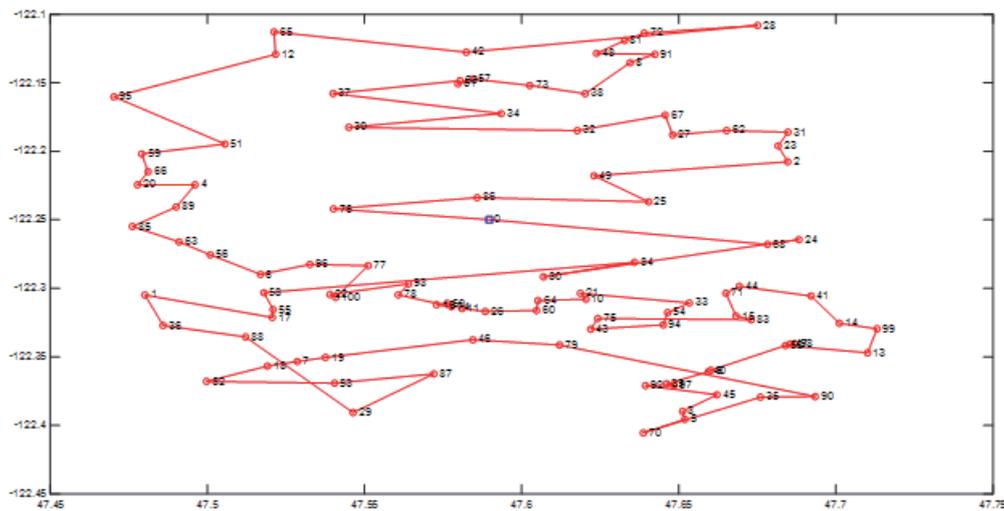


Fig. 9. Optimal route for traditional truck transfer.

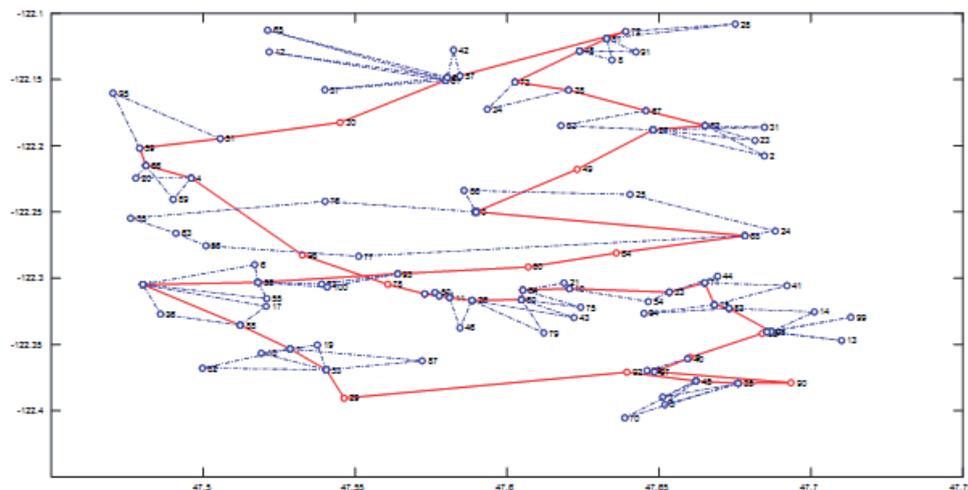


Fig. 10. Optimal route for truck and drone cooperative transfer.

particularly in cost reduction for conveying nucleic acid testing samples. Across the five computation instances, the collaborative transfer truck and drone model yielded a noteworthy 50% reduction in transportation costs on average. For medical testing facilities, this collaborative transport model presents a pioneering and effective resolution for the conveyance of nucleic acid testing samples, delivering substantial cost savings, heightened efficiency, and expedited test result acquisition.

Conclusions

This study proposes an innovative solution to address the challenges faced in COVID-19 sample transport and climate change. Our approach is based on a co-transport working model of multiple multi-visit drones on a truck, with the primary objectives of cost minimization and swift return of samples to the inspection center. To achieve this objective, we established a model of the MV-MFSTSP in which two vehicles work together, and adopted a solving algorithm called GALNS. This algorithm comprises three pivotal components: firstly, an initial solution construction method founded on genetic algorithms, which formulates the initial solution through the “sort first, then group” methodology; secondly, the ALNS algorithm, which uses customized neighborhood operations to continuously improve the solution; Finally, we introduce a simulated annealing mechanism to further augment the algorithm’s performance. Through testing across various scales, our research findings evince that the proposed MV-MFSTSP model and the GALNS algorithm exhibit wide-ranging applicability and effectiveness. Compared with the traditional truck transfer model, our solution is more competitive in reducing transmission risks, saving costs and reducing greenhouse gas emissions.

Future research endeavors shall encompass the exploration of integrating multiple trucks to enhance the efficiency of sample transfers. Moreover, we shall take into account real-world delivery time window constraints to better align with the exigency of timely sample delivery. Our ongoing efforts shall also involve the development of even more streamlined algorithms dedicated to resolving this problem model.

Acknowledgments

This research was funded by the National Natural Science Foundation of China, grant no. 70431003.

Portions of this work were presented at the 35th Chinese Control and Decision Conference in 2023, Research of the path optimization for virus detection sample transfer based on the cooperation between truck and multiple drones.

Conflict of Interest

The authors declare no conflict of interest.

References

- GUSMANOV R., SEMIN A., STOVBA E., AVARSKII N., ZALILOVA Z., FAIZOV N. Developing a strategy for sustainable rural development in the COVID-19 pandemic. *Polish Journal of Environmental Studies*, **32** (2), 125, **2023**.
- VIDYA R., KUMAR S., THOMAS A.C., SHARMILA P. Repercussions of the COVID-19 pandemic on the air quality of Chennai, India. *Polish Journal of Environmental Studies*, **32** (4), 3739, **2023**.
- JIANGUO W., JIASHENG L., SHIJUN L., ZHIYANG P., ZHE X., XUFENG W., RUICHENG Y., JIANFEI L. Detection and analysis of nucleic acid in various biological samples of COVID-19 patients. *Travel Medicine and Infectious Disease*, **37**, 101673, **2020**.
- SF Drone Collaborates with Luohu People’s Hospital to Carry Out Normalized Medical Delivery. Available online: <http://cpl.org.cn/articles/78553> (accessed on 22.10.2021).
- MURRAY C.C., CHU A.G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, **54** (54), 86, **2015**.
- AGATZ N., BOUMAN P., SCHMIDT M. Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, **52** (4), 965, **2018**.
- RAJ R., MURRAY C. The multiple flying sidekicks traveling salesman problem with variable drone speeds. *Transportation Research Part C: Emerging Technologies*, **120**, 102813, **2020**.
- LUO Z., POON M., ZHANG Z. Z., LIU Z., LIM A. The multi-visit traveling salesman problem with multi-drones. *Transportation Research Part C: Emerging Technologies*, **128**, 103172, **2021**.
- WANG X., POIKONEN S., GOLDEN B. The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, **11**, 679, **2017**.
- HAM A. M. 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*, **91**, 1, **2018**.
- LIU Y.Q., HAN J., ZHANG Y., LI Y., JIANG T. Multivisit drone-vehicle routing problem with simultaneous pickup and delivery considering no-fly zones. *Discrete Dynamics in Nature and Society*, **2023**, 21, **2023**.
- BOCCIA M., MASONE A., SFORZA A., STERLE C. A column-and-row generation approach for the flying sidekick travelling salesman problem. *Transportation Research Part C: Emerging Technologies*, **124**, 102913, **2021**.
- PENG K., DU J., LU F., SUN Q.G., DONG Y., ZHOU P., HU M.L. A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery. *IEEE Access*, **7**, 19, **2019**.
- FREITAS J.C.D., PENNA P.H.V. A variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research*, **27**, 267, **2020**.
- HA Q. M., DEVILLE Y., PHAM Q. D., HÀ M.H. A hybrid genetic algorithm for the traveling

- salesman problem with drone. *Journal of Heuristics*, **26**, 219, **2020**.
16. MATHEW N., SMITH S.L., WASLANDER S.L. Planning paths for package delivery in heterogeneous multirobot teams. *IEEE Transactions on Automation Science and Engineering*, **12** (4), 1298, **2015**.
 17. DAYARIAN I., SAVELSBERGH M., CLARKE J-P. Same-day delivery with drone resupply. *Transportation Science*, **54** (1), 229, **2018**.
 18. KIM S., MOON I. Traveling salesman problem with a drone station. *IEEE Transactions on Systems Man Cybernetics-System*, **49** (1), 42, **2019**.
 19. WANG C., LAN H. An expressway based TSP model for vehicle delivery service coordinated with truck + uav. Proc of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 307, **2019**.
 20. BARNAWI A., CHHIKARA P., TEKCHANDANI R., KUMAR N., BOULARES M. A CNN-based scheme for COVID-19 detection with emergency services provisions using an optimal path planning. *Multimedia Systems*, **29**, 1683, **2023**.
 21. YANG S.P., GUO X.P., GAO J.J. Research on the joint delivery problem of “contactless” trucks and unmanned aerial vehicles. *Industrial Engineering and Management*, **1**, **2021**.
 22. PENG Y., LI Y.J. Optimization of collaborative delivery paths for truck drones considering the impact of the epidemic. *China Journal of Highway and Transport*, **33** (11), 73, **2020**.
 23. WU G., MAO N., LUO Q., XU B., SHI J., SUGANTHAN P.N. Collaborative truck-drone routing for contactless parcel delivery during the epidemic. *IEEE Transactions on Intelligent Transportation Systems*, **23** (12), 77, **2022**.
 24. LIU C.S., WU Z., ZHOU Y.F., LUO L., ZHOU X.C., XIE X.L. Optimization of truck-uav dynamic collaborative delivery path for emergency material supply in epidemic areas. *System Science and Mathematics*, **11**, 1, **2019**.
 25. JI J. H., LIU Y.J., BIE Y.M., WANG L.H. A method for distribution of living materials in blocked communities based on the collaboration of drones and trucks. *Transportation System Engineering and Information*, **22** (5), 264, **2022**.
 26. LI H. L., XIONG K., XIE X. M. Multiobjective contactless delivery on medical supplies under open-loop distribution. *Mathematical Problems in Engineering*, **2021**, 7, **2021**.
 27. JIANG L., LIANG C. Y., ZANG X. N. A double layer heuristic algorithm for contactless delivery problem of truck and uav collaboration. *Chinese Journal of Management Science*, **22** (5), 1, **2022**.
 28. MACRINA G., PUGLIESE L., GUERRIERO F., LAPORTE G. Drone-aided routing: a literature review. *Transportation Research Part C: Emerging Technologies*, **120**, 102762, **2020**.
 29. MADANI B., NDIAYE M. Hybrid truck-drone delivery systems: a systematic literature review. *IEEE Access*, **10**, 92854, **2022**.
 30. TONG B., WANG J.W., WANG X., ZHOU F.H., MAO X.H., ZHENG W.L. Optimal route planning for truck-drone delivery using variable neighborhood tabu search algorithm. *Applied Sciences*, **12** (1), 529, **2022**.
 31. YÜREK E.E., ZMUTLU H.C. Traveling salesman problem with drone under recharging policy. *Computer Communications*, **179** (1), 35, **2021**.
 32. PHAN A.T., NGUYEN T.D., PHAM Q.D. Traveling salesman problem with multiple drones. *Proceedings of the Ninth International Symposium on Information and Communication Technology*, 46, **2018**.
 33. CAMPBELL J.F., SWEENEY D., ZHANG J. Strategic design for delivery with trucks and drones. *Supply Chain Analytics Report SCMA* (04 2017), **2017**.
 34. SEIFRIED K. The traveling salesman problem with One truck and multiple drones. *Social Science Electronic Publishing*, **5**, 1, **2019**.
 35. GONZALEZ-R P.L., CANCA D., ANDRADE-PINEDA J.L., CALLE M., LEON-BLANCO J.M. Truck-drone team logistics: a heuristic approach to multi-drop route planning. *Transportation Research Part C: Emerging Technologies*, **114**, 657, **2020**.
 36. MARA S.T.W., RIFAI A.P., SOPHA B.M. An adaptive large neighborhood search heuristic for the flying sidekick traveling salesman problem with multiple drops. *Expert Systems with Applications*, **205**, 117647, **2022**.
 37. WANG Z., SHEU J.B. Vehicle routing problem with drones. *Transportation Research Part B: Methodological*, **122**, 350, **2019**.
 38. TAMKE F., BUSCHER U. A branch-and-cut algorithm for the vehicle routing problem with drones. *Transportation Research Part B: Methodological*, **144**, 174, **2021**.
 39. EUCHI J., SADOK A. Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones. *Physical Communication*, **44**, 101236, **2021**.
 40. LEI D., CUI Z., LI M. A dynamical artificial bee colony for vehicle routing problem with drones. *Engineering Applications of Artificial Intelligence*, **107**, 104510, **2022**.
 41. MENG S., GUO X., LI D., LIU G.Q. The multi-visit drone routing problem for pickup and delivery services. *Transportation Research Part E: Logistics and Transportation Review*, **169**, 102990, **2023**.
 42. KUO R., LU S.H., LAI P.Y. SETYO T.W.M. Vehicle routing problem with drones considering time windows. *Expert Systems with Applications*, **191**, 116264, **2022**.
 43. JEONG H., SONG B., LEE S. Truck-drone hybrid delivery routing: payload-energy dependency and no-fly zones. *International Journal of Production Economics*, **214**, 220, **2019**.
 44. HAN J., LIU Y.Q. Research of the path optimization for virus detection sample transfer based on the cooperation between truck and multiple drones. *2023 35th Chinese Control and Decision Conference (CCDC)*, **2023**.
 45. ZHAN S.H., LIN J., ZHANG Z.J., ZHONG Y.W. List-based simulated annealing algorithm for traveling salesman problem. *Computational Intelligence and Neuroscience*, **2016**, 12, **2016**.
 46. CAMPBELL J.F., SWEENEY II D.C., ZHANG J., PAN D. Strategic design for delivery with linked transportation assets: Trucks and drones. *Technical Report Midwest Transportation Center*, **2018**.