GA-LNS optimization for helicopter rescue dispatch

Jieyuan Cheng, Yuan Gao, *Member, IEEE*, Yongliang Tian, Hu Liu

***Abstract*—Aviation emergency rescue has become one of the most effective means for natural disaster relief due to its flexible and timely characteristics. A reasonable emergency dispatch plan can guarantee the effective implementation of all the rescue measures. Most of previous studies in this area focused on the scheduling and routing but ignored the impact of the specific rescue process, for example the fuel consumption of various helicopters. In this paper, a multi-helicopter-multi-trip Aviation Rescue Routing Problem (ARRP) is analysed which covers the whole rescue process. In addition, a time-domain procedural simulation model is built which can consider different helicopters, refueling or not, various resource locations, multiple disaster sites and other operation factors. Based on that, a Genetic Algorithm (GA) hybridized Large Neighborhood Search (LNS) algorithm (GA-LNS) is proposed for optimization. In ARRP, single search algorithm may lead to the local optimum due to complexity. In contrast, the distance greedy strategy and the load ratio strategy are combined in GA-LNS which can fix the local optimum problem. More specifically, based on the helicopter-tagged-task-sequenced chromosome, the single-point crossover operator is used in GA and then, the worst removal strategy and the first/last insertion strategy are adopted in LNS. Finally, the numerical experiments are exercised to verify the effectiveness of the proposed GA-LNS algorithm which is compared with three traditional basic heuristic algorithms and a stateof-the-art memetic algorithm.**

***Index Terms*—Aviation rescue, Helicopter mission planning, Multi-trip rescue process, Fuel consumption, Procedural simulation, Genetic algorithm, Large neighborhood search.**

I. Introduction

NATURAL disasters, such as earthquakes, tsunamis, floods and hurricanes, have caused tremendous damage in the past and continue to threaten infrastructure and millions of people every year. When natural disasters occur, the main task of emergency rescue is to respond in the shortest time to reduce casualties and economic losses [1]. Therefore, the aviation emergency rescue has become one of the most common and effective emergency measures in response to natural disasters by virtue of its fast response speed, high efficiency and few restrictions. The efficient allocation and scheduling of rescue units is the key to emergency rescue. It is not only related to whether emergency supplies can be transported to the disaster site accurately and efficiently, but also determines the success/failure of rescue mission.

In this paper, we investigate the transfer of personnel (rescuers and disaster victims) and relief supplies. The main task of aviation emergency rescue is to put rescuers and supplies into disaster sites and to transfer the disaster victims to resettlement sites. In contrast to the traditional capacitated vehicle routing problem, the number of requests for relief in a disaster situation may strongly exceed the capacity of the available fleet. Thus, different helicopters are considered here for the rescue mission and most of transportation tasks need to rely on their cooperation.

Our problem is an extension of the classic pickup-delivery problem (PDP) [2]. In PDP, every vehicle follows a route to satisfy transportation requests. Along a given route, each vehicle picks up or drops off the load at required/assumed locations. And each load has to be transported by one vehicle.

There are three main differences between the studied problem and the classic PDP. First, the fleet consists of different types of helicopters. Therefore, the problem not only considers the route planning but also the collaboration between different helicopters. Second, the load at the origin may be split and transported by multiple helicopters and the destination is also not unique. In other words, the disaster victims may be relocated to different resettlement sites, and the rescuers/supplies may be transferred to different disaster sites. Third, owing to the limited fuel quantity, the helicopter may need to return to the airport for refueling [3], [4].

Here, our objective is to design a set of routes for helicopters such that the overall task completion time can be minimized. At the same time, all task demands should be satisfied. The problem is modeled as a mix-fleet multi-trip split-pickup-delivery Aviation Rescue Routing Problem (ARRP). In this problem, we make the following assumptions: (a) The velocity of helicopters is constant (ignoring the acceleration and deacceleration); (b) The speed of people and supplies getting on/off the helicopter is constant; (c) The demands of disaster sites are deterministic (without considering uncertainties); (d) Each rescue task has the same priority.

To address the ARRP, a procedural simulation model is established based on time series. In this way, the problem complexity can be dramatically reduced because, the allocation of helicopters can only be updated whenever they are available, rather than coupling with various constraints. For the search algorithm, a genetic algorithm (GA) hybridized large neighborhood search (LNS) is proposed. In this optimization algorithm, the GA chromosome is composed of the task sequence of each helicopter and the tasks are ordered chronologically. First, the initial population is filled by procedural simulation based on greedy algorithm; After which, the global search optimization is employed. Next, a local search mechanism (LNS) is introduced into the framework of GA to reassign poorly behaved genes in the chromosome. Therefore, the whole structure of the chromosome and the general framework of GA are modified accordingly to fit ARRP and the introduced LNS.

The remainder of this paper is organized as follows: In

Manuscript received **, 2022; revised **, 2023. (*Corresponding author: Yuan Gao*)

Jieyuan Cheng, Yongliang Tian, and Hu Liu are with the School of Aeronautical Science and Engineering, Beihang University, Beijing 100083, China. (e-mail: 15981937197@163.com; tianyongliang_buaa@163.com; liuhu@buaa.edu.cn)

Yuan Gao is with the School of Engineering, University of Leicester, Leicester LE1 7RH, UK. (e-mail: yuan.gao@leicester.ac.uk)

Section II, we describe the relevant literature, including studies on pickup-delivery problem (PDP), split-delivery vehicle routing problem (SDVRP), hybrid GAs, and allocation and scheduling of rescue units in disaster response. We then provide the formulation of the ARRP to determine the constraints and optimization objective in Section III. In Section IV, the proposed algorithm is detailed with three parts: procedural simulation, GA operation, and LNS operation. In Section V, the experimental results of a case are reported to evaluate the performance of the proposed algorithm. Finally, Section VI summarizes the study.

II. Literature review

In this paper, we discuss an aviation rescue routing problem involving multi-trip, split-pickup-delivery, and fuel quantity limit. To the best of our knowledge, the ARRP has not been previously studied, mainly due to the specific area and the problem's complexity. However, the problem is related to two variants of the vehicle routing problem: PDP and SDVRP. Recently, the application of vehicle routing problem in post-disaster response attracts increasing interest from the researchers [1], [5]–[7]. To solve the problems, many methods including the hybrid genetic algorithm have been proposed. Next, we will review the literature from three aspects: related vehicle routing problems, disaster relief operational problem, and evolutionary optimization algorithms.

A. Related vehicle routing problems

Pickup-delivery problem (PDP) is an important variant of the vehicle routing problem (VRP) in which goods or passengers have to be transported from origins to destinations. According to the number of origins and destinations, the PDP is classified into three main categories: many-to-many (M-M) problem, one-to-many-to-one (1-M-1) problem, and one-to-one (1–1) problem [8]. In M-M problem, any vertex can serve as a source or as a destination for any commodity. In 1-M-1 problem, deliveries and pickups concern two distinct sets of commodities: some are shipped from the depot to the customers, and others are picked up at the customers and delivered to the depot. And in 1–1 problem, each commodity has only one origin and one destination.

The PDP arises in many aspects of real-life, such as urban courier operations, municipal waste collection, and the repositioning of inventory between retail stores. Various researches have been done on this problem. Lu and Dessouky [9] formulated the multiple vehicle pickup and delivery problem (MVPDP) as a 0-1 integer programming and developed a branch-and-cut algorithm to optimally solve the problem. Ropke and Pisinger [10] presented an adaptive large neighborhood search algorithm for solving the pickup and delivery problem with time windows (PDPTW) which relies on multiple subheuristics both to remove and to reinsert customers in the solution. Zhu et al. [11] modeled the one-to-many-to-one dynamic pickup-and-delivery problem (DPDP) as a multi-objective optimization problem and proposed a multi-objective memetic algorithm based on locality-sensitive hashing to address the problem. Ancele et al. [12] investigated a rich vehicle routing problem with pickup and delivery including numerous attributes and developed a multi-threaded meta-heuristic based on Simulated Annealing.

Split-delivery vehicle routing problem (SDVRP) is also a mainstream study in the field of vehicle routing problem. In contrast to the traditional VRP, each customer can be visited multi-times, and the demand may be greater than the vehicle capacity. In contrast to the traditional VRP, each customer can be visited multi-times, and the demand may be greater than the vehicle capacity. The SDVRP were introduced by Dror and Trudeau [13], [14], their study showed that there can be potential savings of total distances travelled and total vehicles used compared with capacitated vehicle routing problem (CVRP). Archetti et al. [15] analyzed the maximum possible savings obtained by allowing split deliveries. Some exact approaches for the SDVRP were suggested in the literature, such as cutting-plane algorithm [16], [17], column generation [18], and dynamic program [19]. However, more researches have focused on using heuristic algorithms to deal with medium-large SDVRP instances. Archetti et al. [20] proposed a three-phase Tabu Search (TS) heuristic for the problem. A scatter search algorithm was developed by Campos et al. [21] which generates initial populations with two different procedures. Boudia et al. [22] applied a Memetic Algorithm with population management (MAPM) and designed four new components especially devoted to the SDVRP. Silva et al. [23] implemented a multi-start Iterated Local Search (ILS) based heuristic to deal with the SDVRP considering both limited and unlimited fleet. Gu et al. [24] proposed a heuristic based on an Adaptive Large Neighborhood Search (ALNS) to solve the commodity constrained SDVRP and used several local search moves and a mathematical programming based operator to improve the solutions.

B. Disaster relief operational problem

Disaster management activities can be usually divided into four phases: mitigation, preparedness, response, and recovery [25]. For the response phase, most studies investigate the dispatching and routing of rescue units [26]–[29]. Generally, there are two most important intervention activities: the logistics of materials and the evacuation of people. Rath and Gutjahr [30] considered the warehouse location-routing problem (WLRP) to establish a supply system after a disaster and developed a math-heuristic algorithm based on the adaptive epsilon constraint to address the problem. Zhang et al. [28] modeled a multi-resource emergency response problem considering possible secondary disasters by an integer mathematical programming. Tlili et al. [31] discussed enhancing the response-time of emergency medical services providers by improving the ambulance routing problem and developed a genetic based algorithm to solve the problem.

In the entire emergency rescue system-of-systems, the aviation emergency rescue has been a crucial even essential measure in response to natural disasters by virtue of its fast response speed, high efficiency and few restrictions. However, compared with the studies of ground emergency transportation planning (e.g. vehicles [32], trains [33], etc.), research on aviation emergency rescue is still rarely reported. Barbarosoğlu et al. [34] developed a mathematical model and a hierarchical multi-criteria methodology for helicopter mission planning. Ozdamar [35] proposed an efficient planning system for coordinating helicopter operations, which can accommodate the special aviation constraints of helicopters and handle large scale helicopter missions.

C. Evolutionary optimization algorithms

Disaster relief operational problem is an NP-hard (nondeterministic polynomial-time hardness) problem, for which the traditional exact methods have difficulty in developing a solution in a limited time. Evolutionary algorithms, due to their simplicity, sufficient flexibility and general applicability, can obtain satisfying solutions in an acceptable computational time [36]. Hence, the evolutionary algorithms, including genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO), hybrid algorithms, etc., have been increasingly used in disaster relief operational problem. Zheng and Ling [37] developed a GA-based cooperative optimization method for the emergency transportation planning problem which divides the integrated problem into a set of subcomponents, evolves the sub-solutions concurrently, and brings the sub-solutions together to build the complete solution. Yuan and Wang [38] presented a multi-objective path selection model based on a single-objective model that further considers chaos, panic and congestion in time of disaster to minimize the total travel time along a path and to minimize the path complexity, and proposed an ant colony optimization algorithm to solve the model. Bozorgi-Amiri [39] developed a modified particle swarm optimization algorithm for disaster relief logistics under uncertain environment to minimize the sum of the expected total cost (which includes costs of location, procurement, transportation, holding, and shortage) and the variance of the total cost.

Hybrid GAs are optimization algorithms based on the synergistic combination of GAs and other search and optimization techniques [40]. These algorithms have been effectively applied to solve the vehicle routing problem (VRP) and its variants. Vidal et al. [41] proposed a framework with advanced diversity management for the GA and demonstrated its effectiveness on a series of VRP variants [42], [43]. Liu et al. [44] developed an effective hybrid GA to solve the multi-depot open vehicle routing problem (MDOVRP). Zhang et al. [45] provided a hybrid GA with the advanced framework and a customized recombination operator to generate high-quality solutions for multi-trip dial-a-ride problem (MTDARP). Willey and Salmon [46] studied the electric vehicle route-planning decisions in the presence of multiple charging options and used the GA to optimize the location of roadways with dynamic charging capabilities.

III. Aviation rescue routing problem (ARRP)

The ARRP can be formulated by a directed graph

$$G = (V, E) \tag{1}$$

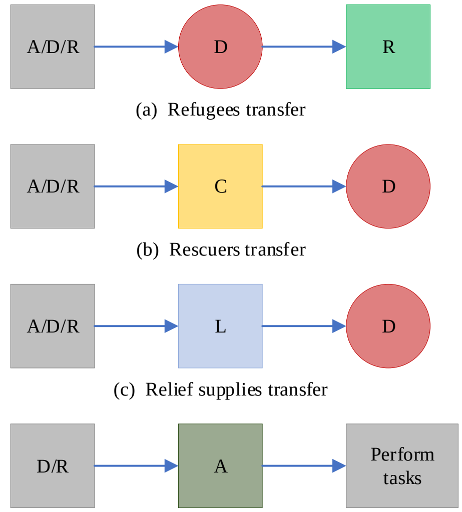


Fig. 1. Four possible task execution routes. In case (a), the helicopter may depart from the airport (A), resettlement location (R), or other disaster site (D) to the disaster site and transport refugees to the resettlement location. Ditto for cases (b) and (c). In case (d), if the remaining fuel is not enough for next task, the helicopter needs to fly to the airport for refueling first.

where V denotes the node set, and E represents the directed edge set

$$E = \{(i, j) \mid i, j \in V\} \tag{2}$$

There are five subsets in V: A, D, R, L and C which denote set of airports, disaster sites, resettlement locations, depots and rescue team bases, respectively, i.e.

$$V = A \cup D \cup R \cup L \cup C \tag{3}$$

Each disaster site has at least one rescue task. The tasks will be executed by heterogeneous helicopters H which are parked at different airports in A. There are three types of rescue tasks P: (a) Refugees transfer: the helicopter flies to the disaster site and transports refugees to the resettlement location; (b) Rescuers transfer: the helicopter flies to the rescue team base and transports rescuers to the disaster site; (c) Relief supplies transfer: the helicopter flies to the depot and transports relief supplies to the disaster site. Thus, let d_{ip} denote the demand of disaster site $i \in D$ for rescue task $p \in P$. Note that, before each task, the helicopters will judge whether the remaining fuel is enough for its next task. If not, the helicopter will first fly to the airport for refueling, and then perform the following task. The execution procedures of different types of rescue tasks are shown in Fig. 1.

One solution to ARRP consists of a set of helicopter task sequences, one sequence for each helicopter. Let Φ_h represent the helicopter task sequence which is performed by the helicopter $h \in H$. All helicopter task sequences are arranged chronologically to form the entire fleet task sequence (denoted by Φ). Each helicopter task sequence corresponds to a helicopter flight plan, which represents the locations passed by the helicopter during the tasks. Let Ψ_h be the flight plan of helicopter h. Each task $\varphi \in \Phi_h$ is then uniquely associated with one directed transportation arc

$$\psi \in \Psi_h = \{\langle i,j,k\rangle \; or \; \langle i,a,j,k\rangle \mid i,j,k \in V, a \in A\} \quad (4)$$

where $\langle i,j,k\rangle$ denotes two consecutive directed edges $(i,j),(j,k) \in E$. Note that, each helicopter $h \in H$ has a maximum load capacity Q^h. When the demand of a disaster site exceeds the boarding capacity of a single helicopter, this helicopter may re-visit the site to continue the transportation task but, other helicopters could come to finish the same task as well if they are available. We use the $w_{ih}^{\varphi p}$ to represent the amount of people/supplies transferred when helicopter h performs p-type task φ at disaster site i and the d'_{ip} to indicate the residual demand for task p in disaster site i.

The ARRP model focuses on the overall length of the rescue time. Thus, the optimization work performed in this paper is to find the optimal solution which minimizes the overall fleet task completion time τ, while at the same time, satisfies all the task demands of disaster sites. The task completion time τ is determined by the time when the last helicopter completes its tasks. We use τ_h to denote the task completion time of helicopter h. The overall time includes not only the helicopter flight time but the duration for people to get on/off the helicopter and loading/unloading the relief supplies. Let t_{ij}^h denote the travel time of helicopter h from i to j, while u_r^h and u_s^h indicate the speed for people to get on/off the helicopter and the rate for relief supplies to load/unload, respectively. In addition, the fuel quantity can also affect the helicopter's task completion time. When the fuel quantity is not enough for next task, the helicopter needs to fly to the airport for refueling first. In this paper, the O_M^h and O_R^h denote the maximum fuel capacity and the remaining fuel for the helicopter h, while u_U^h and u_A^h denote the fuel consumption rate and refueling speed. The single task time is denoted as t_φ^h which comprises helicopter travel time, ground working time and fueling time (if necessary). A summary of notations used is provided in Table I.

TABLE I
NOTATION USED TO MODEL THE ARRP.

Sets:	
V	Set of locations, $V = A \cup D \cup R \cup L \cup C$
E	Set of directed edges
A	Set of airports
D	Set of disaster sites
R	Set of resettlement locations
L	Set of relief supplies depots
C	Set of rescue team bases
P	Set of task types
H	Set of helicopters
Φ	Fleet task sequence
Φ_h	Helicopter task sequence performed by helicopter h
Ψ_h	Flight plan of helicopter $h \in H$
Parameters:	
d_{ip}	Demand of disaster site i for rescue task p
t_{ij}^h	Travel time from i to j with (i,j) by helicopter h
Q^h	Load capacity of helicopter h
v^h	Airspeed of helicopter h
u_r^h	Speed for people to get on/off the helicopter h
u_s^h	Loading/unloading speed of relief supplies for helicopter h
O_M^h	Maximum fuel capacity of the helicopter h
u_U^h	Fuel consumption rate of the helicopter h
u_A^h	Refueling speed of the helicopter h
Intermediate variables:	
d'_{ip}	Residual demand for task $p \in P$ in disaster site $i \in D$ after helicopter performed the task
$\langle i,j,k\rangle, \langle i,a,j,k\rangle$	Directed transportation arc, $i,j,k \in V$, $a \in A$
O_R^h	Remaining fuel of the helicopter h
t_φ^h	Execution time of a single task $\varphi \in \Phi_h$
τ_h	Task completion time of helicopter h
Decision variables:	
$x_{ij}^{h\varphi}$	Binary, indicate whether helicopter h traverse edge (i,j) during task $\varphi \in \Phi_h$
$w_{ih}^{\varphi p}$	The amount of people/supplies transferred when helicopter h performs p-type task $\varphi \in \Phi_h$ at disaster site $i \in D$
Objective function:	
τ	Task completion time among all helicopters, $\tau = \max\{\tau_1, \tau_2, \dots, \tau_h, \dots\}$, $h \in H$

The ARRP model can be formulated as follows:

$$\text{minimize } \tau, \text{ i.e., } \min\{\max\{\tau_1, \tau_2, \dots, \tau_h, \dots\}\} \quad (5)$$

subject to:

$$0 \le w_{mh}^{\varphi p} \le \min\{Q^h, d'_{mp}\}, m \in D, \varphi \in \Phi_h, p \in P \quad (6)$$

$$\sum_{h \in H} \sum_{\varphi \in \Phi_h} w_{mh}^{\varphi p} = q_{mp}, \forall m \in D, p \in P \quad (7)$$

$$t_{ij}^h u_U^h \le O_M^h, \forall (i,j) \in E, \forall h \in H \quad (8)$$

$$t_{ij}^h x_{ij}^{h\varphi} u_U^h \le O_R^h, \forall (i,j) \in E, h \in H, \varphi \in \Phi_h \quad (9)$$

$$\left(\sum_{(i,j)\in E} t_{ij}^h x_{ij}^{h\varphi} + \min\{t_{ka}^h\} \right) \cdot u_U^h \le O_R^h, \quad h \in H, \varphi \in \Phi_h, k \in \psi, a \in A \quad (10)$$

$$t_\varphi^h = \begin{cases} \sum_{(i,j)\in E} t_{ij}^h x_{ij}^{h\varphi} + 2 \cdot \frac{w_{mh}^{\varphi p}}{u^h} & \psi = \langle i,j,k\rangle \\ \sum_{(i,j)\in E} t_{ij}^h x_{ij}^{h\varphi} + 2 \cdot \frac{w_{mh}^{\varphi p}}{u^h} + \frac{Q_M^h - Q_R^h}{u_A^h} & \psi = \langle i,a,j,k\rangle \end{cases},$$

$$\forall h \in H, \varphi \in \Phi_h, u^h \in \{u_r^h, u_s^h\}, m \in D, a \in A \quad (11)$$

$$\sum_{\varphi \in \Phi_h} t_\varphi^h = \tau_h, \forall h \in H \quad (12)$$

$$\tau \ge \tau_h, \forall h \in H \quad (13)$$

$$x_{ij}^{h\varphi} \in \{0,1\}, \forall (i,j) \in E, h \in H, \varphi \in \Phi_h \quad (14)$$

The objective function of ARRP is stated in (5). Constraints (6)-(7) deal with the helicopter load capacity and disaster site demands: In Constraint (6), the maximum amount of people/supplies a helicopter can deliver is bounded by the helicopter capacity and residual demand; And the demands of each disaster site must be satisfied by Constraint (7).

Constraints (8)-(10) are for the helicopter's flight time and fuel quantity. Constraint (8) means that the fuel consumption between any two points is less than the maximum fuel capacity of the helicopter. Regarding the Constraint (9), it means the fuel consumption of travel arc during single rescue task is less than the remaining fuel of the helicopter. Constraint (10) is to make sure that the helicopter is capable of flying to the nearest airport after each single task. If any of the above Constraints (9)-(10) are not met, the helicopter needs refueling before doing the next rescue task.

Constraints (11)-(13) deal with the task execution time of the helicopter. Constraint (11) indicates single task execution time. Constraint (12) guarantees that the task completion time of the single helicopter is equal to the sum of single task time. Constraint (13) tracks the task completion time for the entire fleet. Constraint (14) defines the domain of the decision variable.

Since the method of mathematical programming cannot well reflect the changes of key parameters over time in the rescue process, to better describe and document the whole rescue process, we construct a procedural simulation model in Section IV.A which integrates the above constraints to limit the helicopter performance, mission route, rescue time, etc.

IV. Heuristics to solve ARRP

The solution proposed to solve the ARRP is a heuristic method based on GA hybridized LNS (GA-LNS) and procedure simulation. To clarify the algorithm, the new notions used in this algorithm are given in Table II.

There are mainly three phases in the proposed optimization algorithm. In the first phase, a series of feasible solutions are randomly generated through a procedural simulation, which is the first generation of GA. The second phase is using the GA operation. It includes not only the traditional operators: selection, crossover and, mutation, but also the non-equal chromosome preprocessing and the procedural simulation with chromosome repair. Since the distance greedy strategy is adopted in the first two phases, there may be potential solutions wasting the helicopter capacity, which could prolong the rescue time. Therefore, in the third phase, the LNS operation based on the load ratio strategy is introduced for the local optimization after the above GA operation. The third phase uses a series of destroy and repair operators to explore the local optimal solution, which can effectively avoid the design candidates generated with capacity wasting. Noting that, even though the introduced LNS adds a few operations to the optimization process, the proposed GA-LNS can guarantee the drawbacks of distance greedy strategy would be reasonably corrected according to the load ratio, thereby speeding up the overall optimization process.

The latter two phases are exercised through a certain number of iterations, after which the final optimal solution will be output. The overall flow of the proposed algorithm is shown in Fig.2. These GA and LNS phases will be clearly explained with technical details; before that, we may have to discuss the implemented fundamental technique, procedural simulation, which generates the task complete time for each design candidate in the optimization.

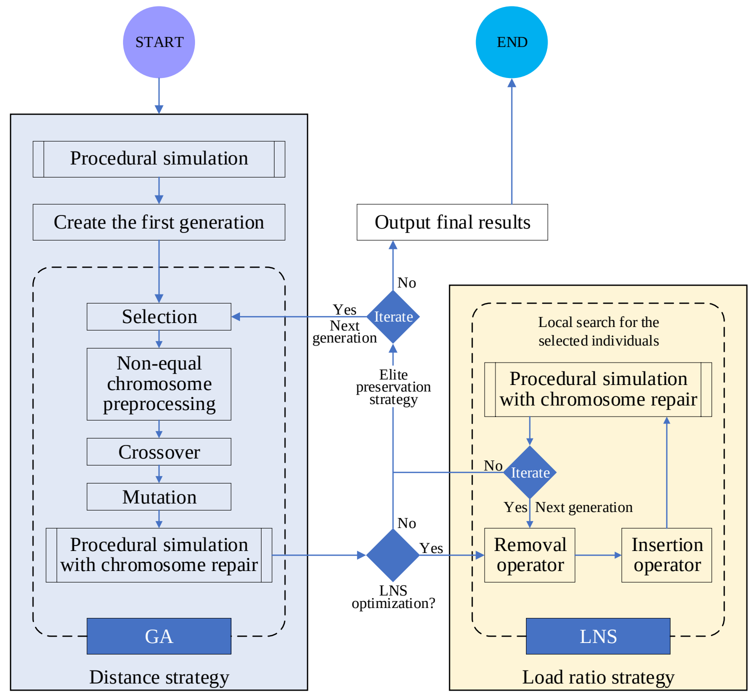


Fig. 2. The algorithm framework of proposed GA-LNS.

A. Procedural simulation

The procedural simulation (PS) is mainly to simulate the execution of tasks in the time domain, which discretizes the process in unit time (*second*) . It consists of two parts, one is task allocation, the other is state transition. In the procedural simulation, the state of the helicopter (fuel capacity, load, position, etc.) changes over the discretized time. In order to demonstrate the change of helicopter state, we introduce the state transition array π_h^φ into PS. The state transition array has two different types: three elements $[s_{ij}, s_{jk}, lr]$ (no refueling process) or four elements $[s_{ia}, s_{aj}, s_{jk}, lr]$ (with

TABLE II

Notation used in the algorithm.

Sets:	
T	Set of demands
$\mathbb{S}$	Set of population
Parameters & Variables:	
π_h^φ	State transition array of helicopter $h \in H$ when executing the task $\varphi \in T$
s_{ij}	Distance between i and j with $(i,j) \in E$
lr	Load ratio
$[s_{ij}, s_{jk}, lr]$, $[s_{ia}, s_{aj}, s_{jk}, lr]$	Two cases of state transition array,$\{i,j,k\} \in V$, $a \in A$
u^h	Speed of boarding and alighting, $u^h \in \{u_r^h, u_s^h\}$
N	Size of the fleet task sequence
$len(\Phi)$	Size of the fleet task sequence $\Phi \in \mathbb{S}$
u_A^h	Refueling speed of the helicopter h

Algorithm 1: Procedural simulation

Data: Helicopters H, locations V
Result: Fleet task sequence Φ, Task completion time τ
▷ Initialize set of demands T
$T \leftarrow \emptyset$;
for $i \in D$ **do**
  $T \leftarrow T \cup \{d_{ip}\}$;
end
▷ Time-domain procedural simulation based on the constraints of Section III
$time \leftarrow 0, \Phi \leftarrow \emptyset$;
repeat
  ▷ Task allocation part: assign tasks to the available helicopters
  for $h \in H$ **do**
    ❶ Randomly assign a task $\varphi \in T$ to helicopter h, or ❷ Assign a task $\varphi \in T$ with the shortest route to h;
    Label the task φ with the helicopter tag h;
    $\Phi \leftarrow \Phi \cup \{\varphi\}$;
    if $\left(\sum t_{ij}^{h} x_{ij}^{h\varphi} + \min\{t_{ka}\}\right) \cdot u_U^h \leq O_R^h$ **then**
      $\pi_h^{\varphi} \leftarrow [s_{ij}, s_{jk}, 0]$;
    else
      $\pi_h^{\varphi} \leftarrow [s_{ia}, s_{aj}, s_{jk}, 0]$;
    end
  end
  ▷ State transition part
  for $h \in H$ **do**
    if $\pi_h^{\varphi} \neq [0, 0, 0]$ **then**
      if $s_{ij} \neq 0$ **then**
        $s_{ij} \leftarrow s_{ij} - v^h, \quad O_R^h \leftarrow O_R^h - O_U^h$;
      else if $lr < Q^h$ *and* $\varphi \neq 0$ **then**
        $lr \leftarrow lr + u^h, \quad \varphi \leftarrow \varphi - u^h$;
      else if $s_{jk} \neq 0$ **then**
        $s_{jk} \leftarrow s_{jk} - v^h, \quad O_R^h \leftarrow O_R^h - O_U^h$;
      else
        $lr \leftarrow lr - u^h$;
      end
    end
    if $\varphi = 0$ **then**
      $T \leftarrow T \backslash \{\varphi\}$;
    end
  end
  $time \leftarrow time + 1$;
until $T = \emptyset$;

refueling process), where the last element lr represents the rescue process and the other elements represent the travel process at each instant. The algorithm of procedural simulation is presented in Algorithm 1.

1) Task allocation: The first part (i.e., task allocation) is used to assign tasks to the available helicopters and initialize the state of the helicopters, which paves the way for the following state transition part.

The task allocation adopts distance greedy strategy. In other words, for each helicopter, the initial task is randomly assigned and the subsequence tasks are assigned according to the closest distance. Compared with the random assignment strategy, the distance greedy strategy could converge to the optimal solution faster, the comparison results of the case will be given in Section V. However, the strategy may lead to two undesired situations: (a) Too many helicopters perform one task. (b) The capacity of the helicopter does not match the task volume (e.g., a large helicopter performs a small volume task). These situations may waste the capacity of the helicopter thereby prolonging the rescue time. To this end, we design the follow-up algorithm in Section IV.C using the LNS to avoid the non-desired solutions in optimization.

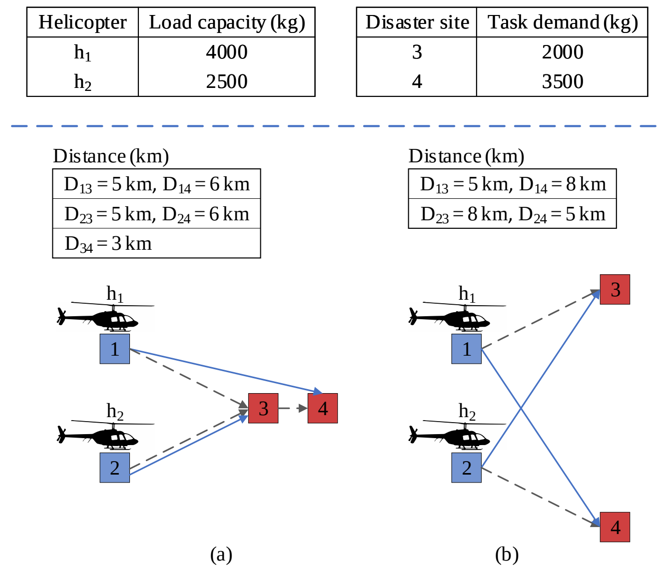


Fig. 3. Illustrative example of the two non-desired situations.

Fig. 3 shows two illustrative examples of the above non-desired scenarios. In these examples, assume that all data are the same except for the data of helicopter load capacity and disaster site task demands. In Fig. 3(a), according to the distance greedy strategy, the helicopters simultaneously go to disaster site 3 and then to disaster site 4 for the following tasks, as shown by the gray dashed path; however, this will lead to a lower load ratio and an increased distance, causing a longer rescue time. In Fig. 3(b), according to the distance greedy strategy, helicopter h_1 goes to disaster site 3 for the task and helicopter h_2 to disaster site 4. In this scenario, the load ratio of h_1 is lower, while h_2 cannot complete the task in one trip and needs to make two round trips to complete it, resulting in a super long time. By using the proposed GA-LNS algorithm, these un-desired solutions can be effectively avoided and the optimal ones for two scenarios can be found, which are marked by blue solid lines in Fig. 3.

After the helicopter is assigned a task, it will perform the task along the shortest transportation arc based on the remaining fuel (Constraint (9) in Section III). Therefore, the initialization of π_h^{φ} is $[s_{ij}, s_{jk}, 0]$ or $[s_{ia}, s_{aj}, s_{jk}, 0]$. Note that, in the following optimization part, the task allocation part of procedural simulation is using the newly generated fleet task sequence, which will be discussed in Section IV.B.

2) State transition: The second part of procedural simulation (i.e., state transition) is used to record all the information (in time domain) during the task operation (flight and ground working), namely the changes of state transition array π_h^φ, remaining fuel O_R^h and task demand φ. When a task demand is satisfied, it will be removed from the demand set T. Obviously, three-element state transition array and four-element state transition array have the same internal logic during the task operation, the only difference is going to the airport for refueling or not. Therefore, in Algorithm 1, we only take the three-element state transition as an example for the explanation.

A series of feasible solutions are generated through procedural simulation to form the first-generation population. The feasible solution is a chromosome and each of its genes has a helicopter tag (h_k). Each chromosome in the population corresponds to a fleet task sequence, and each gene on the chromosome represents a task. The helicopter tag on the gene represents the helicopter performing this task. This helicopter tag will be further used in the procedural simulation with chromosome repair of GA (Section IV.B) and the removal/insertion operators of LNS (Section IV.C).

Since one task may be executed multiple times, the size of each chromosome may be different. For this reason, in the subsequent optimization process, the chromosomes need to be preprocessed to enable them to perform GA operation while maintaining the original characteristics.

Algorithm 2: Population preprocessing

Data: Initial population $\mathbb{S}$
Result: Pretreated population
▷ Determine the longest fleet task sequence
$length \leftarrow \max_{\Phi \in \mathbb{S}} \{len(\Phi)\}$
foreach Φ *in the population* $\mathbb{S}$ **do**
  if $len(\Phi) < length$ **then**
    $\Phi \leftarrow \Phi \cup \{None_1, \cdots, None_{length-len(\Phi)}\}$;
  end
end

B. GA operation

In the GA procedure, as shown in Fig. 2, we optimize the population through a series of selection, non-equal chromosome preprocessing, crossover, mutation and procedural simulation with chromosome repair. The iterative optimization of the population is based on the fitness value (i.e., reciprocal of task completion time). In order to prevent the loss of optimal individual in evolution, the GA operation adopts elite preservation strategy.

1) Selection operator: In the selection operator, we adopt roulette wheel selection to generate population for crossover operator. The selection probability of each individual is proportional to its fitness value. The larger the fitness value, the easier it will be selected. Compared with random selection, roulette wheel selection can guide the population to the evolution direction more effectively without destroying the diversity of the population.

2) Population preprocessing: As mentioned, due to the inconsistent size of chromosome, the population needs to be preprocessed before the crossover operator. The method of non-equal chromosome preprocessing is using the "$None$" values to fill the positions to make sure each chromosome has the same size. The pretreated population has the same characteristics as the original population. Algorithm 2 shows the preprocessing method adopted in this part.

3) Crossover operator: Crossover operator is used to create new chromosomes from pretreated population. It maintains the diversity of the population. There are various crossover operators in the GA, such as single-point crossover and uniform crossover. In generally, compared with the single-point crossover operator, other crossover operators have faster crossover mixing speed and explore more solution space. However, in this paper, we adopt a single-point crossover. This is because the single-point crossover characterizes simple operation with less damage thus can maintain the original intention of the distance greedy strategy as much as possible. If the uniform crossover operator is used, the role of the distance greedy strategy would not exist when the genes at the same position cross over probabilistically.

In single-point crossover, a single cross point on both parents' chromosomes is selected randomly. All genes beyond that point in either chromosome are swapped with each other. The resulting chromosomes are new individuals [47].

4) Mutation operator: In order to maintain the diversity of the population and enhance the local search ability, the mutation operator is applied. In mutation operator, two random numbers i,j are selected from [1,2,...,N] as the mutation points. For the chromosome $\Phi \in \mathbb{S}$, the genes at these two mutation points swap their positions.

5) Procedural simulation with chromosome repair: After applying the above four operations to create the new chromosomes, procedural simulation with chromosome repair (PSCR) is exercised. The main reason of using PSCR is that the newly generated chromosomes, after the above GA operation, may have the meaningless tasks (genes) or miss some tasks, which should be repaired to be a feasible and practical fleet task sequence.

Three operations are included in PSCR: (a) removing the "$None$" values and the tasks that have been completed but still stay in the fleet task sequence, (b) adding the tasks that need to be executed but are not in the fleet task sequence, and (c) determining the fitness value of the individual. The PSCR has the same state transition process as the PS in Section IV.A but, the task allocation process is different. The task allocation of PSCR proceeds in Algorithm 3.

In the task allocation of PSCR, each helicopter selects the task from its corresponding task sequence Φ_h which is a part of the newly generated fleet sequence Φ. If the task (e.g., the completed task or "$None$" value) is not in the set of demands, it will be skipped and the next task with the same label will be selected. If the fleet task sequence is swept and the set of demands is not empty (i.e. the tasks are not finished), the helicopter will adopt distance greedy strategy to select the next

Algorithm 3: Task allocation part of procedural simulation with chromosome repair

Data: Helicopters H, locations V, task sequence Φ

$\triangleright$ Task allocation: assign tasks to availiable helicopters
for $h \in H$ **do**
  if *h is available* **then**
    $\varphi \leftarrow None$;
    while *True* **do**
      if *h-tagged tasks in Φ* **then**
        $\varphi \leftarrow 1st$ h-tagged task, $\Phi \leftarrow \Phi \backslash \{\varphi\}$;
        if *φ in T* **then**
          break;
        end
      else
        $\varphi \leftarrow None$, break;
      end
    end
    if *φ is None* **then**
      Assign a task $\varphi' \in T$ with the shortest route to $h, i.e., \varphi \leftarrow \varphi'$;
    end
  end
  $\Phi_h \leftarrow \Phi_h \cup \{\varphi\}$;
  if $\left(\sum_{(i,j)\in E} t_{ij}^h x_{ij}^{h\varphi} + \min\{t_{ka}\}\right) \cdot u_U^h \leq O_R^h$ **then**
    $\pi_h^\varphi \leftarrow [s_{ij}, s_{jk}, 0]$;
  else
    $\pi_h^\varphi \leftarrow [t_{ia}, t_{aj}, t_{jk}, 0]$;
  end
end

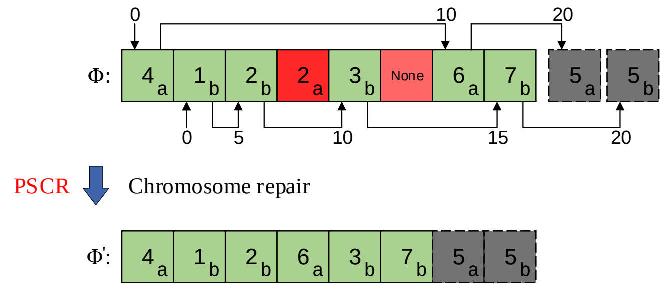


Fig. 4. Illustrative example of PSCR. a, b are two helicopters. the task demand set is $T = \{1, 2, 3, 4, 5, 6, 7\}$.

task.

Fig. 4 shows an illustrative example of the above process. In the example, suppose two helicopters a,b with infinite load capacity (i.e., each helicopter can complete one task at a time), the task demand set $T = \{1, 2, 3, 4, 5, 6, 7\}$, and a newly generated chromosome example $\Phi = 4_a 1_b 2_b 2_a 3_b None 6_a 7_b$. In addition, assume that it takes 10 minutes for helicopter a to complete each task, and 5 minutes for helicopter b. The helicopters select tasks sequentially from Φ. At 0 instant, helicopter a selects task “4” and helicopter b selects task “1”. 5 minutes later, helicopter b completes task “1”, and then executes the next task: task “2”. At 10 minutes, helicopter a completes task “4”, and then finds that task “2” has been completed ($T = \{3, 5, 6, 7\}$). Therefore, helicopter a will skip task “2” (also “$None$”) and finally select task “6”. Afterwards, two helicopters perform follow-up tasks in sequence. At 20 minutes, two helicopters would complete all the tasks in Φ. However, at this time $T = \{5\}$, which means the task “5” is left. The helicopters need to select the nearest task in T until T is empty. Therefore, at 20 minutes, the two helicopters will perform task “5” together until its completion.

Note that, the newly generated chromosomes using the above operations may cause the helicopter running out of fuel. In that case, the chromosome will be regarded as an infeasible solution and set the fitness value of the fleet task sequence to zero. Since the GA operation uses roulette wheel selection, this solution will be eliminated in the next iteration.

6) Elite preservation strategy: In order to prevent the optimal individual of the current population from being lost in the next generation, resulting in the algorithm not converging to the global optimal solution, the elite preservation strategy [48] is used before the next iteration. In this part, we first sort the old and new individuals according to their fitness values, and then select a certain number of individuals with high fitness as the next generation.

C. LNS operation

The GA operation usually has a strong global search capability for optimization [49]. In order to enhance the local search ability, LNS operation [50] is introduced to further improve the current population. The main motivation of using LNS is that, as mentioned in Section IV.A, there are two drawbacks of the distance greedy strategy which can make the algorithm converge to a local optimum. To address this problem, the LNS operation is proposed here which is in series but independent with the GA operation. As shown in Fig. 2, in a certain generation, the LNS operation optimizes the current population and returns the optimized population to GA as the next generation for subsequent operations.

There are two points to note about LNS. First, it is not used for every generation of the optimization which can be time consuming. Only several generations are randomly selected. Second, instead of performing the LNS operation on each individual of this generation population, we randomly select a few outstanding individuals (e.g., those ranked in the top 30% of fitness values). Based on that, the efficiency of the algorithm can be effectively improved. In the case study, for every potential LNS loop, only five individuals are selected, which includes the optimal individual in the current generation and four randomly chosen individuals in the top 30%.

Fig. 5 shows the general scheme of LNS operation. The LNS algorithm improves the current feasible solutions by using a series of removal and insertion operators based on the load ratio strategy. In LNS operation, a feasible solution is destroyed by a removal operator and then reconstructed by an insertion operator. The removal and insertion operators will be discussed separately, as below.

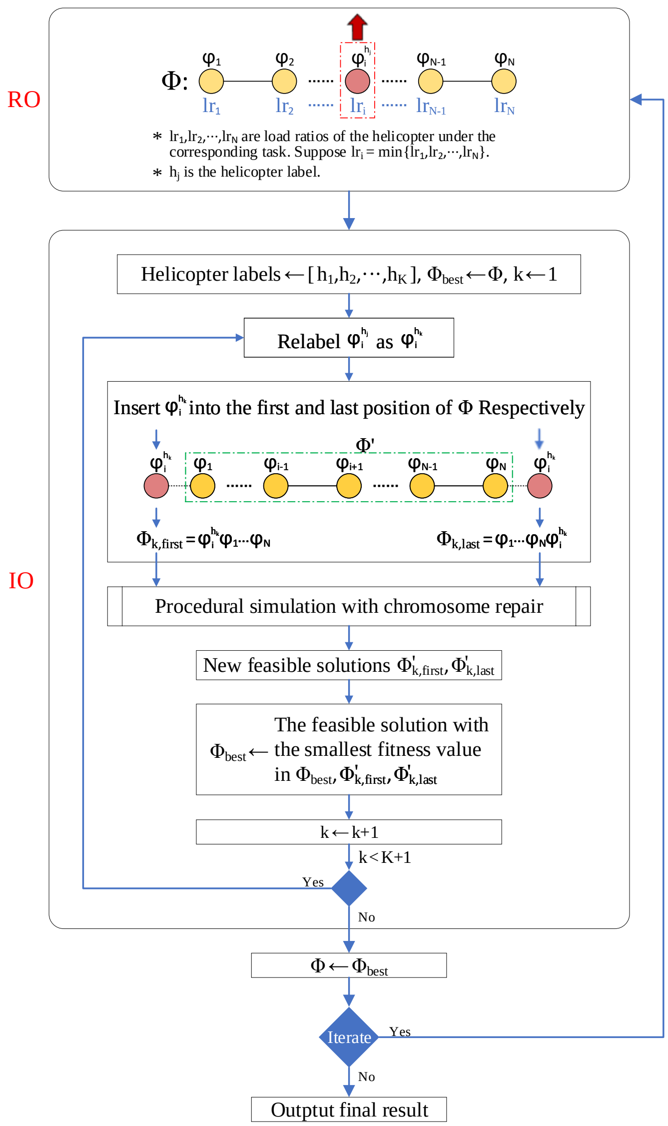


Fig. 5. LNS operation.

1) Removal operator (RO): The removal operator adopts a worst removal strategy. As shown in Fig. 5, in each iteration of LNS, the task $\varphi_i^{h_j}$ performed by the helicopter h_j is removed from the current solution because it has the lowest load ratio.

2) Insertion operator (IO): When the task $\varphi_i^{h_j}$ is removed from the solution Φ, the algorithm proceeds to the repair phase. As shown, the insertion operator first relabels the removed task $\varphi_i^{h_j}$ as $\varphi_i^{h_k}$ and then re-inserts it into the destroyed solution Φ'. To maintain the order of the original sequence as much as possible, we only insert the relabeled removed task $\varphi_i^{h_k}$ into the first and the last position of Φ' thus, two fleet task sequences $\Phi_{k,first}$,$\Phi_{k,last}$ will be generated. After that, the PSCR is performed to refine the two fleet task sequences and determine their fitness values. They are compared with the current local optimum and the best of three solutions are taken as the new local optimum. Next, the removed task's label is replaced with a new helicopter label and the above operations are repeated. When the removed task has been tagged by all helicopter labels, the LNS starts a new iteration. In summary, the insertion operator has two functions: (a) to repair the destroyed solution, and (b) to play a role similar to the mutation operator in genetic algorithm.

It is worth noting that the LNS does not consume much time. The reasons are as follows. First, the LNS optimization is performed only for several individuals in certain generations of the GA. Second, the LNS does not need to iterate many times, because there are usually not many helicopters with very low load ratios in the fleet task sequence. In the test case of this paper (Section V), the LNS is iterated five rounds at a time. Third, the insertion operator internally only loops $2 \times K$ times, where K is the number of helicopter tags. Practically, the value is not large, and here $K = 5$. We will give a complete comparison of the time consumed by GA-LNS and other algorithms in Section V.C.

V. Case Study and Numerical Experiments

The numerical experiments were conducted to evaluate the proposed algorithm. In order to test the algorithm, it was compared with three traditional basic heuristic algorithms (genetic algorithm (GA), ant colony optimization (ACO), and discrete particle swarm optimization (DPSO) [51]) and a state-of-the-art memetic algorithm (MA) [45]. Each algorithm was executed 50 times independently for the test instance to obtain overall statistical results. Here, we refer to an independent execution of the algorithm as a round of testing. They were coded in Python and were implemented on a PC equipped with an Intel i7-8700 CPU, 16 GB RAM, and Windows 10 Professional operation system.

In this section, we first introduce the test case, then set the parameters of each algorithm, and finally evaluate the performance of our algorithm. It is noted that, since many different factors are considered in this case including helicopter types, disaster locations, people and relief supplies transportation, refueling or not, etc., the proposed optimization algorithm can be generalized to other aviation rescue scenarios/cases.

A. Test case

In this rescue instance, there are five heterogeneous helicopters: M-26, AC313, S76, H155 and H225. Generally, all the helicopters can transport both people and relief supplies but, the specific max capabilities vary. In addition, their airspeed, maximum fuel capacity, fuel consumption rate and other key parameters are also different. All the parameters of five helicopters are shown in Table III.

The numbers of airports, disaster sites, resettlement locations, depots and rescue team bases are 2, 5, 2, 1, and 1, respectively. As shown in Table IV, the longitude and latitude of each location are randomly generated in the interval $[120, 122]$ and $[28, 31]$ separately. The distance s_{ij} between locations i and j are then derived by the Great-circle distance [52]. The different types of demands for each location are randomly drawn from the corresponding intervals. The value range of each task demand is summarized in Table V. Based on the demands of each location, a task set will be generated accordingly, as summarized in Table VI.

B. Parameter settings

In order to evaluate the performance of the proposed GA-LNS, it is compared with the other four algorithms. Note that, to solve the ARRP, all algorithms are integrated with the procedural simulation in this study. The procedural simulation in all algorithms is the same except for the task allocation part. Among them, GA and DPSO adopt a random allocation strategy, ACO utilizes the pheromone and distance to assign tasks, and MA is based on the reg-k regret insertion strategy.

The parameter configurations of these algorithms are according to references, which are given in Table VII. For the GA-LNS and GA, the parameters p_c and p_m represent the probability of chromosome crossover and mutation. In addition, the uniform crossover operator is used in GA, and the probability of gene swapping at the corresponding positions of paired chromosomes is denoted by p_u. For the ACO, the parameters α and β control the relative importance of the pheromone versus the heuristic information; ρ is the pheromone evaporation factor. For the DPSO, the parameters ω, c_1, and c_2 represent inertia weight, cognitive coefficient and social coefficient, respectively. For the MA, the parameter λ decides when to invoke survivor selection, n_c controls calculation of the diversity contribution, and n_e determines the number of elite solutions reserved for the next iteration. For all the algorithms, the population size and the maximum number of iterations are set as 80 and 120.

C. Evaluation of the GA-LNS

In the numerical experiments, the algorithms are executed multiple times to find the relevant data statistics of the optimal value, so as to evaluate the performance of the algorithms. In this paper, the statistical data indexes used for algorithm performance comparison include: mean, standard deviation (S.D.), maximum, minimum, median, etc. Subsequently, we evaluate the algorithms in terms of two benchmarks: convergence and robustness of the algorithm.

1) Convergence of the algorithm: In order to evaluate the convergence of all the algorithms, we record the task completion times of the same generation in each round of testing and calculate their mean values. The average convergence curves of the four algorithms are plotted in Fig. 6. As shown, compared with other algorithms, the GA-LNS shows a much stronger search ability and faster convergence velocity. Namely, it can obtain better solutions in a certain number of iterations.

As seen from Fig. 6, we can also find that the ACO has the smallest initial value, followed by the MA, and the other

TABLE III
HELICOPTER PERFORMANCE PARAMETERS.

Helicopter performance		M-26	AC313	S76	H155	H225
Airspeed (km/h)		255	251	269	280	276
Max. fuel capacity (kg)		12000	3500	900	993	2044
Fuel consumption rate (kg/h)		3000	1065	273	330	450
Refueling speed (kg/h)		100	60	30	30	50
Load capacity	People No.	82	27	13	13	19
	Supplies (kg)	20000	4000	2300	1820	5400
Boarding speed	People $(/s)$	30	45	60	60	45
	Supplies (kg/s)	6	4.5	3	3	4.5

TABLE IV
COORDINATES AND DEMANDS OF LOCATIONS.

Loc.	Geographical coordinates	Number of refugees	Demand for rescuers	Demand for supplies(kg)
A_1	121.43, 28.66	–	–	–
A_2	120.04, 29.38	–	–	–
C_1	121.10, 28.37	–	–	–
L_1	120.20, 29.65	–	–	–
R_1	120.04, 28.89	–	–	–
R_2	121.44, 29.29	–	–	–
D_1	120.45, 29.05	90	0	4000
D_2	120.69, 28.15	110	30	0
D_3	120.84, 30.12	0	40	4000
D_4	121.01, 29.15	60	60	8000
D_5	120.75, 29.50	70	50	0

TABLE V
VALUE RANGE OF RESCUE DEMAND TYPES.

Task demand	Value range
Number of refugees	$\{0, 10, 20, \cdots, 150\}$
Demand for rescuers	$\{0, 10, 20, \cdots, 100\}$
Demand for relief supplies (kg)	$\{0, 4000, 8000, 12000\}$

TABLE VI
THE TASK SET.

Task No.	Disaster site	Number of refugees	Demand for rescuers	Demand for relief supplies(kg)
1	D_1	0	0	4000
2	D_1	90	0	0
3	D_2	0	30	0
4	D_2	110	0	0
5	D_3	0	0	4000
6	D_3	0	40	0
7	D_4	0	0	8000
8	D_4	0	60	0
9	D_4	60	0	0
10	D_5	0	50	0
11	D_5	70	0	0

TABLE VII
PARAMETER SETTING OF FIVE ALGORITHMS.

Algorithm	Parameter Configurations
GA-LNS [53]	$p_c = 0.7$, $p_m = 0.02$
GA [53], [54]	$p_c = 0.7$, $p_m = 0.02$, $p_u = 0.5$
ACO [55]	$\alpha = 1.14$, $\beta = 2.02$, $\rho = 0.11$
DPSO [56], [57]	$\omega = 0.729$, $c_1 = c_2 = 1.4962$
MA [45]	$\lambda = 100$, $n_c = 48$, $n_e = 48$

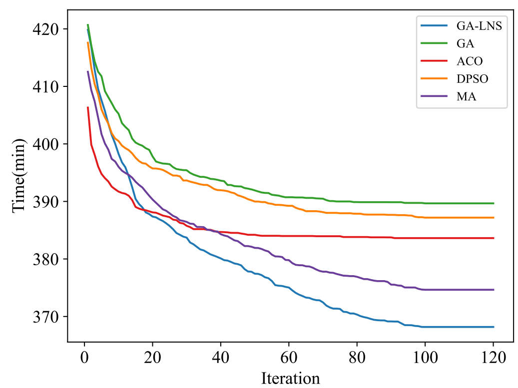


Fig. 6. Evolutionary trajectories of the five algorithms.

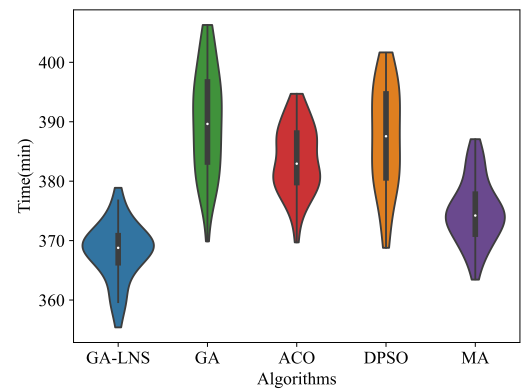


Fig. 7. 50 rounds of optimization results using five algorithms.

algorithms have larger initial values. For the GA-LNS, the distance greedy strategy can lead to the waste of helicopter load capacity and prolong the rescue time, and its initial solution is not necessarily better than the results obtained by the random allocation strategy. For the MA, the regret insertion method is able to obtain a better initial solution compared to GA-LNS because it improves the basic greedy insertion method via a look-ahead scheme. For the ACO, when generating the first-generation population, the pheromone does not work, which is equivalent to a random allocation strategy. Namely, the first generation of the ACO combines the random allocation strategy and distance greedy strategy, thus making its initial value better than other algorithms. Even though the initial fitness of GA-LNS is relatively low among five algorithm candidates, based on the proposed combination of LNS with GA operation, the GA-LNS algorithm performs much better than the others during the following search process. In the optimization-seeking phase of all algorithms, the GA-LNS, MA and ACO are able to converge to the optimum faster because they have distance information as a guide. Among them, GA-LNS and MA are able to further converge to better solutions due to their local search mechanism. Moreover, Since the GA-LNS is able to handle the drawbacks caused by the distance greedy strategy (see Section IV), it can get a better solution compared to the MA.

2) Robustness of the algorithm: To further evaluate the convergence effect of the algorithm, we record the optimal values obtained in each round of testing for each algorithm (totally 50 rounds). In every round, all of them are running with 80 individuals and 120 iterations. Fig. 7 shows the statistics results of each algorithm in a violin plot. As shown, the solutions of the GA-LNS are generally much better than other algorithms, no matter for the expectation or for the general distribution. The heuristic is a stochastic optimization algorithm, and the results of each execution of the algorithm may be different. For the algorithm with stable search performance, the difference between each execution result is relatively small. In contrast, for the algorithm with large search performance fluctuation, the difference between execution results can be dramatically large. In order to evaluate the robustness of the algorithm, we calculate the standard deviation of each algorithm. Table VIII summarizes the performance of five algorithm candidates. In the 50 rounds, the best, worst, mean, and standard deviation (S.D.) of the mission operation time are given in this table, together with their computation burden. It is shown that GA-LNS has the smallest value for all the indicators of mission operation time. And the small S.D. value indicates that the performance of GA-LNS is robust for solving the ARRP.

TABLE VIII
PERFORMANCE OF EACH ALGORITHM.

Algorithm	Best ($mins$)	Worst ($mins$)	Mean ($mins$)	S.D. ($mins$)	Computation time (s)
GA-LNS	**355.40**	**378.88**	**368.18**	**4.78**	412.2
GA	369.87	406.28	389.94	8.67	**316.7**
ACO	369.70	394.70	383.63	5.62	395.3
DPSO	368.78	401.67	387.19	8.56	329.8
MA	363.43	387.07	374.85	5.28	409.7

To check any significant difference exists in the performance of algorithms, we conducted the non-parametric Friedman test and the post-hoc Bonferroni test. The Friedman test only reveals the difference among the results of different algorithms. The Bonferroni test is performed after the Friedman test to show which particular pair of algorithms is different from each other in comparison. The two tests used $\alpha = 0.05$ as the level of significance. In the Friedman test, a significant statistical difference is found in comparing the performance of GA-LNS with all algorithms (p values = 9.96×10^{-28}). The p values of the Bonferroni test for pairwise comparisons between GA-LNS and GA, ACO, DPSO, and MA, respectively, are 4.68×10^{-42}, 1.09×10^{-28}, 2.90×10^{-36}, and 4.06×10^{-7}. These p values are less than the Bonferroni adjustment significant level of 0.01 and thus the null hypotheses are rejected. The rejection of the null hypotheses indicates that the results of the GA-LNS are statistically different from those of the other algorithms.

To further compare the performance of each heuristic algorithm fairly, we gave each algorithm the same computational resource (i.e., CPU time) and count the results of running $300s$, $350s$, $400s$ and $450s$ respectively. Each algorithm was

TABLE IX
AVERAGE TASK COMPLETION TIME ($mins$) FOR EACH ALGORITHM WITH THE SAME CPU TIME.

Algorithm	300s	350s	400s	450s
GA-LNS	**371.42**	**369.53**	**368.47**	**367.92**
GA	390.82	389.24	389.24	389.24
ACO	383.95	383.82	383.54	383.54
DPSO	387.70	387.19	387.19	387.19
MA	377.60	376.34	374.93	374.78

run 50 times individually and the average value was taken as the statistical result, which is given in Table IX. As can be seen, the GA-LNS has better results with the same computational resource. The traditional heuristic algorithms basically converge to a local optimum after 300s, and GA-LNS and MA are able to further converge to better values due to the local search mechanism.

D. Illustration of optimal results

In the above part, we evaluated the performance of the algorithms based on the task completion time. In this subsection, we select the generated optimal solution of GA-LNS as an example to show the task sequence, flight plan and task execution process of each helicopter.

In the best solution, five heterogeneous helicopters need to complete all the given tasks, with task completion time of 355.4 *mins*. Table X shows the task sequence and flight plan of each helicopter. Fig. 8 shows the task execution process of each helicopter, which illustrates the change in position and task of the helicopters over time. Here we take AC313 as an example for a brief illustration. As shown, AC313 performs task 1 (relief supplies transfer) in four phases: (a) taking off from the airport A_2 to depot L_1, (b) loading relief supplies, (c) delivering relief supplies to the disaster site D_1, and (d) unloading relief supplies. When task 1 is over, the helicopter starts to perform task 2 (refugees transfer). Since the helicopter is currently at the disaster site, it directly performs the task 2 without an extra flight. Fig. 9 shows the variations of load and fuel quantity with time, where these two variables are normalized. As shown, since relief supplies are continuous variables and personnel are discrete variables, the load curve is a straight line when the helicopter transfers relief supplies and a stepped line for personnel transport. Note that, the helicopter only consumes fuel in flight, not considered on the ground. In addition, since the refueling time is very short relative to the overall task time, the slope of the refueling curve is very large, as shown in Fig. 9(d).

VI. CONCLUSION AND FUTURE STUDIES

In this paper, a multi-heterogeneous-helicopter multi-trip aviation rescue routing problem is studied considering rescue process and fuel consumption. To solve ARRP, a procedural simulation model and a GA hybridized LNS algorithm are proposed, and the routing problem is converted into a task sequence planning problem. At first, the distance greedy strategy is used to generate the initial fleet task sequence population, and each task in the sequence is labeled for subsequent optimization operations. Then, the population is globally optimized with GA operation, where the single-point crossover is used to generate new individuals. However, the distance greedy strategy may result in a waste of helicopter load capacity thus prolonging the rescue time. To this end, the LNS is used for optimization. To remain the characteristics of the fleet task sequence, the removal and insertion operators are adopted with the worst removal strategy and the first/last insertion strategy, respectively.

To validate the performance of the proposed GA-LNS algorithm, numerical experiments were conducted and compared with other three baseline algorithms. The experiments indicate that the convergence speed, convergence effect and robustness of the GA-LNS are better than the others.

In summary, this study contributes to further research on the emergency rescue and the application of helicopters in this field. However, a few limitations are found here and regarded as the potential future studies. First, the model assumes that the demands of disaster sites are deterministic, which is not always realistic and may affect routing decisions. Second, in the emergency response, the urgency of task demands varies but is not considered. In addition, the speed of helicopters in the model in constant, while in real-world it is restricted by airflow, terrain, air traffic control, etc. In such cases, the fuzzy theory is a useful approach to model uncertainty. Future researches could also investigate how to model the uncertainty factor, how fuzzy membership functions can be formulated, and how resulting models can be evaluated. Last but not the least, multiple objectives and their different weights can be considered in the proposed GA-LNS algorithm.

REFERENCES

[1] C. T. Born, S. M. Briggs, D. L. Ciraulo, E. R. Frykberg, J. S. Hammond, A. Hirshberg, D. W. Lhowe, and P. A. O'Neill, "Disasters and mass

TABLE X
TASK SEQUENCE AND FLIGHT PLAN OF EACH HELICOPTER.

Helicopter	Helicopter task sequence	Flight plan
M-26	8-4-10-11	A_1-C_1-D_4-D_2-R_1-C_1-D_5-R_2
AC313	1-2-7-6-11	A_2-L_1-D_1-R_1-L_1-D_4-C_1-D_3-D_5-R_2
S76	3-4-2-2-5-11-9	A_1-C_1-D_2-R_1-D_1-R_1-D_1-R_1-L_1-D_3-D_5-R_2-D_4-R_2
H155	2-2-2-5-6-11	A_2-D_1-R_1-D_1-R_1-D_1-R_1-L_1-D_3-A_1-C_1-D_3-D_5-R_2
H225	3-4-7-9-11-9-9	A_1-C_1-D_2-R_1-L_1-D_4-R_2-D_5-R_2-D_4-R_2-D_4-R_2

casualties: I. general principles of response and management," *JAAOS-Journal of the American Academy of Orthopaedic Surgeons*, vol. 15, no. 7, pp. 388–396, 2007.
[2] M. W. Savelsbergh and M. Sol, "The general pickup and delivery problem," *Transportation science*, vol. 29, no. 1, pp. 17–29, 1995.
[3] M. E. Kabir, I. Sorkhoh, B. Moussa, and C. Assi, "Joint routing and scheduling of mobile charging infrastructure for v2v energy transfer," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 736–746, 2021.
[4] S. Schoenberg and F. Dressler, "Reducing waiting times at charging stations with adaptive electric vehicle route planning," *IEEE Transactions on Intelligent Vehicles*, 2022.
[5] D. Berkoune, J. Renaud, M. Rekik, and A. Ruiz, "Transportation in disaster response operations," *Socio-Economic Planning Sciences*, vol. 46, no. 1, pp. 23–32, 2012.
[6] F. Wex, G. Schryen, S. Feuerriegel, and D. Neumann, "Emergency response in natural disaster management: Allocation and scheduling of rescue units," *European Journal of Operational Research*, vol. 235, no. 3, pp. 697–708, 2014.
[7] A. Baxter, H. W. Lagerman, and P. Keskinocak, "Quantitative modeling in disaster management: A literature review," *IBM Journal of Research and Development*, vol. 64, no. 1/2, pp. 3–1, 2019.
[8] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte, "Static pickup and delivery problems: a classification scheme and survey," *Top*, vol. 15, no. 1, pp. 1–31, 2007.
[9] Q. Lu and M. Dessouky, "An exact algorithm for the multiple vehicle pickup and delivery problem," *Transportation Science*, vol. 38, no. 4, pp. 503–514, 2004.
[10] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transportation science*, vol. 40, no. 4, pp. 455–472, 2006.
[11] Z. Zhu, J. Xiao, S. He, Z. Ji, and Y. Sun, "A multi-objective memetic algorithm based on locality-sensitive hashing for one-to-many-to-one dynamic pickup-and-delivery problem," *Information Sciences*, vol. 329, pp. 73–89, 2016.
[12] Y. Ancele, M. H. Hà, C. Lersteau, D. B. Matellini, and T. T. Nguyen, "Toward a more flexible vrp with pickup and delivery allowing consolidations," *Transportation Research Part C: Emerging Technologies*, vol. 128, p. 103077, 2021.
[13] M. Dror and P. Trudeau, "Savings by split delivery routing," *Transportation Science*, vol. 23, no. 2, pp. 141–145, 1989.
[14] M. Dror and P. Trudeau, "Split delivery routing," *Naval Research Logistics (NRL)*, vol. 37, no. 3, pp. 383–402, 1990.
[15] C. Archetti, M. W. Savelsbergh, and M. G. Speranza, "Worst-case analysis for split delivery vehicle routing problems," *Transportation science*, vol. 40, no. 2, pp. 226–234, 2006.
[16] M. Dror, G. Laporte, and P. Trudeau, "Vehicle routing with split deliveries," *Discrete Applied Mathematics*, vol. 50, no. 3, pp. 239–254, 1994.
[17] J.-M. Belenguer, M. Martinez, and E. Mota, "A lower bound for the split delivery vehicle routing problem," *Operations research*, vol. 48, no. 5, pp. 801–810, 2000.
[18] M. Jin, K. Liu, and B. Eksioglu, "A column generation approach for the split delivery vehicle routing problem," *Operations Research Letters*, vol. 36, no. 2, pp. 265–270, 2008.
[19] C.-G. Lee, M. A. Epelman, C. C. White III, and Y. A. Bozer, "A shortest path approach to the multiple-vehicle routing problem with split pickups," *Transportation research part B: Methodological*, vol. 40, no. 4, pp. 265–284, 2006.
[20] C. Archetti, M. G. Speranza, and A. Hertz, "A tabu search algorithm for the split delivery vehicle routing problem," *Transportation science*, vol. 40, no. 1, pp. 64–73, 2006.

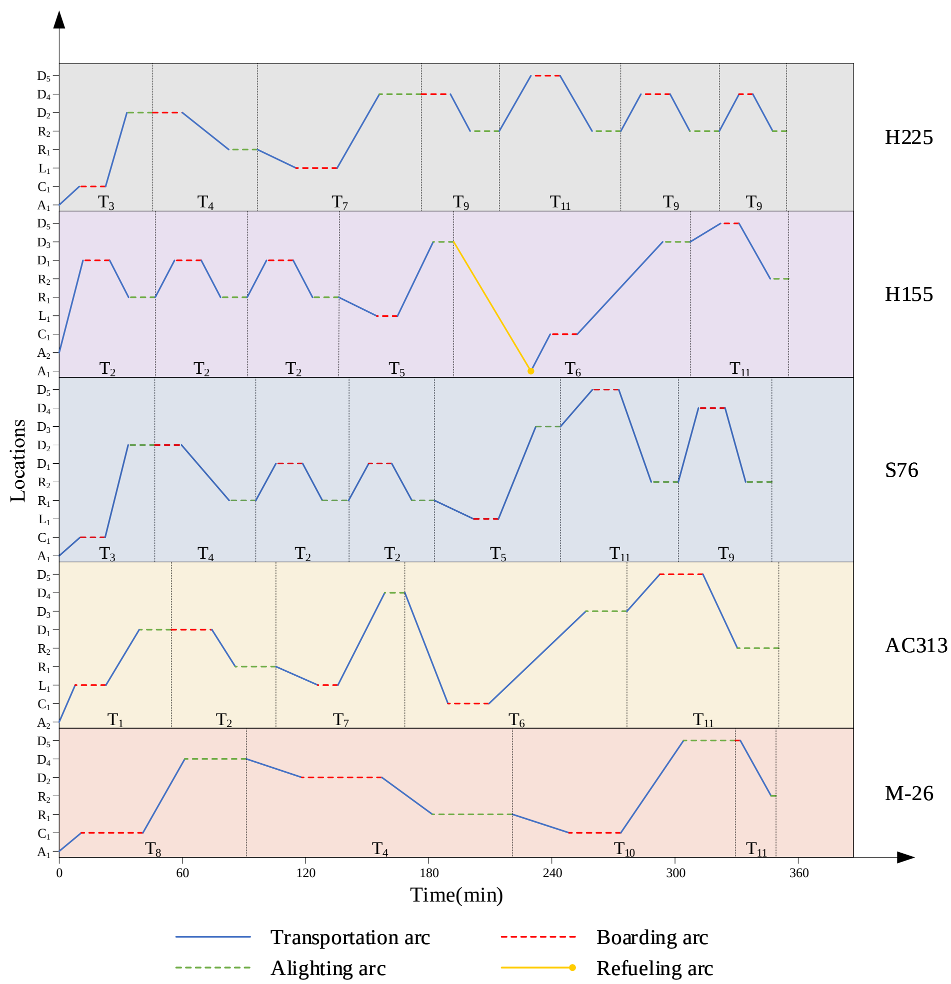
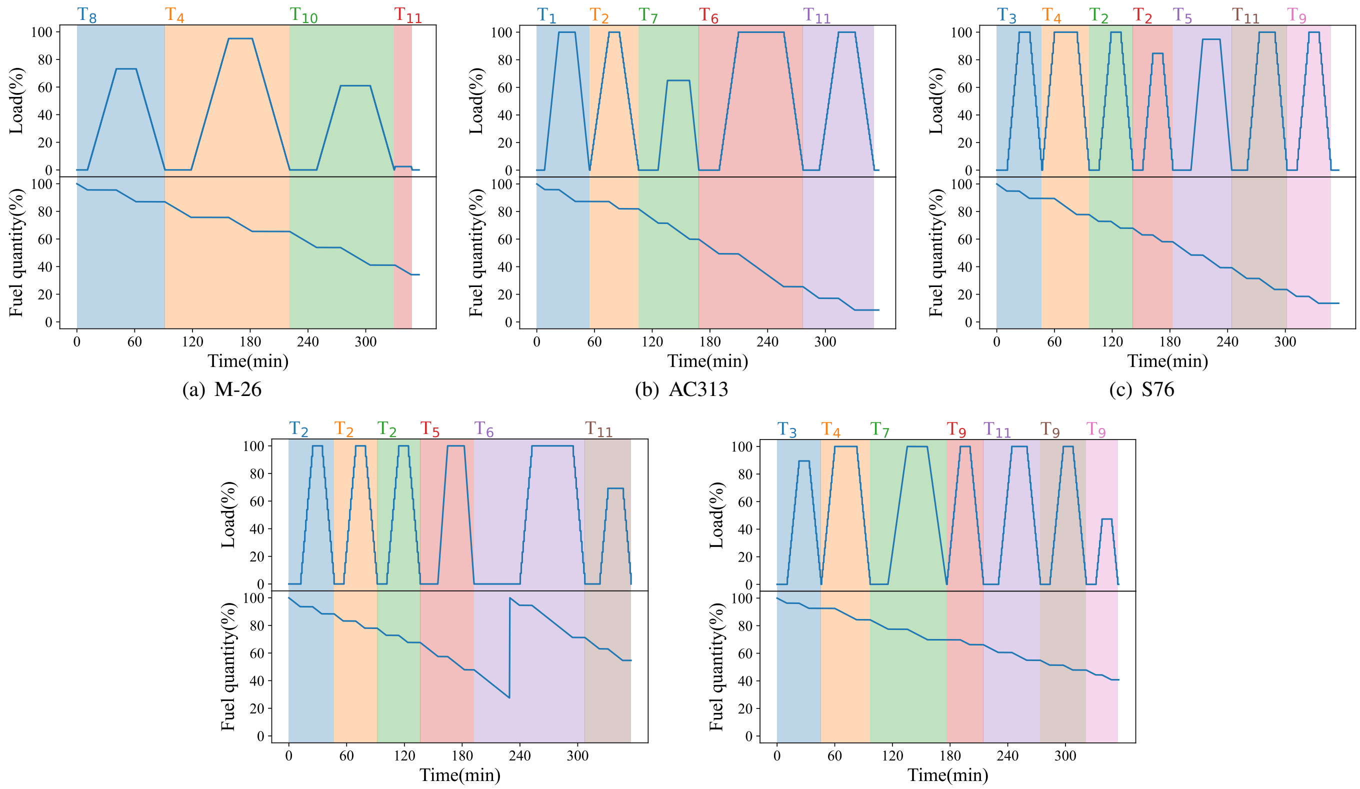


Fig. 8. Task execution process of each helicopter.



(a) M-26 (b) AC313 (c) S76

(d) H155 (e) H225

Fig. 9. The variations of load and fuel quantity of each helicopter.

[21] V. Campos, A. Corberán, and E. Mota, "A scatter search algorithm for the split delivery vehicle routing problem," in *Advances in computational intelligence in transport, logistics, and supply chain management*. Springer, 2008, pp. 137–152.

[22] M. Boudia, C. Prins, and M. Reghioui, "An effective memetic algorithm with population management for the split delivery vehicle routing problem," in *International workshop on hybrid metaheuristics*. Springer, 2007, pp. 16–30.

[23] M. M. Silva, A. Subramanian, and L. S. Ochi, "An iterated local search heuristic for the split delivery vehicle routing problem," *Computers & Operations Research*, vol. 53, pp. 234–249, 2015.

[24] W. Gu, D. Cattaruzza, M. Ogier, and F. Semet, "Adaptive large neighborhood search for the commodity constrained split delivery vrp," *Computers & Operations Research*, vol. 112, p. 104761, 2019.

[25] O. Ergun, G. Karakus, P. Keskinocak, J. Swann, and M. Villarreal, "Operations research to improve disaster supply chain management," *Operations research and management science*, 2010.

[26] A. Haghani and S.-C. Oh, "Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations," *Transportation Research Part A: Policy and Practice*, vol. 30, no. 3, pp. 231–250, 1996.

[27] Y.-H. Lin, R. Batta, P. A. Rogerson, A. Blatt, and M. Flanigan, "A logistics model for emergency supply of critical items in the aftermath of a disaster," *Socio-Economic Planning Sciences*, vol. 45, no. 4, pp. 132–145, 2011.

[28] J.-H. Zhang, J. Li, and Z.-P. Liu, "Multiple-resource and multiple-depot emergency response problem considering secondary disasters," *Expert Systems with Applications*, vol. 39, no. 12, pp. 11 066–11 071, 2012.

[29] P. Wu, F. Chu, A. Che, and M. Zhou, "Bi-objective scheduling of fire engines for fighting forest fires: New optimization approaches," *IEEE transactions on intelligent transportation systems*, vol. 19, no. 4, pp. 1140–1151, 2017.

[30] S. Rath and W. J. Gutjahr, "A math-heuristic for the warehouse location–routing problem in disaster relief," *Computers & Operations Research*, vol. 42, pp. 25–39, 2014.

[31] T. Tlili, S. Abidi, and S. Krichen, "A mathematical model for efficient emergency transportation in a disaster situation," *The American journal of emergency medicine*, vol. 36, no. 9, pp. 1585–1590, 2018.

[32] P. H. V. Penna, A. C. Santos, and C. Prins, "Vehicle routing problems for last mile distribution after major disaster," *Journal of the Operational Research Society*, vol. 69, no. 8, pp. 1254–1268, 2018.

[33] Y.-J. Zheng, M.-X. Zhang, H.-F. Ling, and S.-Y. Chen, "Emergency railway transportation planning using a hyper-heuristic approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 321–329, 2014.

[34] G. Barbarosoğlu, L. Özdamar, and A. Cevik, "An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations," *European Journal of Operational Research*, vol. 140, no. 1, pp. 118–133, 2002.

[35] L. Ozdamar, "Planning helicopter logistics in disaster relief," *OR spectrum*, vol. 33, no. 3, pp. 655–672, 2011.

[36] Y.-J. Zheng, S.-Y. Chen, and H.-F. Ling, "Evolutionary optimization for disaster relief operations: A survey," *Applied Soft Computing*, vol. 27, pp. 553–566, 2015.

[37] Y.-J. Zheng and H.-F. Ling, "Emergency transportation planning in disaster relief supply chain management: a cooperative fuzzy optimization approach," *Soft Computing*, vol. 17, pp. 1301–1314, 2013.

[38] Y. Yuan and D. Wang, "Path selection model and algorithm for emergency logistics management," *Computers & industrial engineering*, vol. 56, no. 3, pp. 1081–1094, 2009.

[39] A. Bozorgi-Amiri, M. Jabalameli, M. Alinaghian, and M. Heydari, "A modified particle swarm optimization for disaster relief logistics under uncertain environment." *International Journal of Advanced Manufacturing Technology*, vol. 60, 2012.

[40] T. A. El-Mihoub, A. A. Hopgood, L. Nolle, and A. Battersby, "Hybrid genetic algorithms: A review." *Eng. Lett.*, vol. 13, no. 2, pp. 124–137, 2006.

[41] T. Vidal, T. G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei, "A hybrid genetic algorithm for multidepot and periodic vehicle routing problems," *Operations Research*, vol. 60, no. 3, pp. 611–624, 2012.

[42] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows," *Computers & operations research*, vol. 40, no. 1, pp. 475–489, 2013.

[43] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A unified solution framework for multi-attribute vehicle routing problems," *European Journal of Operational Research*, vol. 234, no. 3, pp. 658–673, 2014.

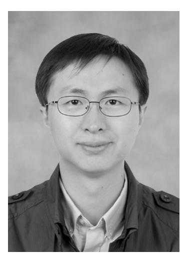
[44] R. Liu, Z. Jiang, and N. Geng, "A hybrid genetic algorithm for the multi-depot open vehicle routing problem," *OR spectrum*, vol. 36, no. 2, pp. 401–421, 2014.
[45] Z. Zhang, M. Liu, and A. Lim, "A memetic algorithm for the patient transportation problem," *Omega*, vol. 54, pp. 60–71, 2015.
[46] L. C. Willey and J. L. Salmon, "Infrastructure optimization of in-motion charging networks for electric vehicles using agent-based modeling," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 760–771, 2021.
[47] Z. Bao and T. Watanabe, "A novel genetic algorithm with cell crossover for circuit design optimization," in *2009 IEEE International Symposium on Circuits and Systems*. IEEE, 2009, pp. 2982–2985.
[48] K. A. De Jong, *An analysis of the behavior of a class of genetic adaptive systems.* University of Michigan, 1975.
[49] P. Bajpai and M. Kumar, "Genetic algorithm–an approach to solve global optimization problems," *Indian Journal of computer science and engineering*, vol. 1, no. 3, pp. 199–206, 2010.
[50] D. Pisinger and S. Ropke, "Large neighborhood search," in *Handbook of metaheuristics*. Springer, 2010, pp. 399–419.
[51] K.-P. Wang, L. Huang, C.-G. Zhou, and W. Pang, "Particle swarm optimization for traveling salesman problem," in *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE cat. no. 03ex693)*, vol. 3. IEEE, 2003, pp. 1583–1585.
[52] C. Carter, "Great circle distances," *SiRF White Paper*, 2002.
[53] C. Chudasama, S. Shah, and M. Panchal, "Comparison of parents selection methods of genetic algorithm for tsp," in *International Conference on Computer Communication and Networks CSI-COMNET-2011, Proceedings*. Citeseer, 2011, pp. 85–87.
[54] W. M. Spears and K. D. De Jong, "On the virtues of parameterized uniform crossover," Naval Research Lab Washington DC, Tech. Rep., 1995.
[55] L. Melo, F. Pereira, and E. Costa, "Mc-ant: a multi-colony ant algorithm," in *International conference on artificial evolution (evolution artificielle)*. Springer, 2009, pp. 25–36.
[56] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, vol. 3. IEEE, 1999, pp. 1951–1957.
[57] Y. Dai, L. Liu, and Y. Li, "An intelligent parameter selection method for particle swarm optimization algorithm," in *2011 Fourth international joint conference on computational sciences and optimization*. IEEE, 2011, pp. 960–964.

Jieyuan Cheng currently works as a combat employment researcher at an unmanned aircraft company, working on UAV combat mission planning. He received his master degree in aeronautical engineering from Beihang University in 2023. His research interests include combat simulation technology, multi-UAV cooperative combat technology, intelligent air warfare, and route planning.

Yuan Gao (Member, IEEE) received his PhD degree in Electrical and Electronic Engineering (EEE) from the University of Nottingham, Nottingham, UK, in 2021. He studied his masters in Aeronautical Engineering at the Beihang University, Beijing, China, 2014-2017. He finished his Bachelor of Engineering degree in 2013 at the Dalian Maritime University, Dalian, China. Since Feb 2023, He has been with the School of Engineering at the University of Leicester, Leicester, UK, as a lecturer in EEE. Prior to that, he was a postdoc research associate in hybrid autonomous system engineering, Department of Aerospace Engineering, at the University of Bristol, Bristol, UK. His research interests include motor drive, model predictive control, multi-agent modelling and simulation, autonomous system engineering, machine learning-based design, control, and maintenance.

Dr. Gao is a Reviewer for the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and IEEE TRANSACTIONS ON POWER ELECTRONICS. He also serves as the Session Co-Chair of the AIAA/IEEE Electric Aircraft Technologies Symposium (EATS) 2020.

Yongliang Tian received the B.S. degree in aircraft design and engineering from Beihang University, Beijing, in 2009, and the Ph.D. degree in aircraft design from Beihang University, Beijing, in 2015. From 2013 to 2014, He studied in school of aeronautics and aerospace engineering of Purdue University in USA as a visiting student. His main research areas are aircraft conceptual design and system engineering, including conceptual software system, virtual simulation technology, intelligent design technology.



Hu LIU , Professor, Youth Top Talent of the National Ten Thousand People Plan. He is also dean of the Department of Aircraft in School of Aeronautical Science and Engineering of Beihang University, head of the national top-quality curriculum "Aircraft Conceptual Design" and national virtual simulation experiment teaching project, deputy director of the Key Laboratory of Aircraft Advanced Design Technology (Beihang University) Ministry of Industry and Information Technology, direction leader of "VR + aviation" research area in State Key Laboratory of Virtual reality Technology and Systems.