Two-Layer Path Planning for Multi-Area Coverage by a Cooperated Ground Vehicle and Drone System

Yangsheng Xia, Chao Chen, Jianmai Shi, Yao Liu, Guohui Li*

***Abstract*—A novel two-layer path planning method for a cooperated ground vehicle (GV) and drone system is investigated, where the GV acts as the mobile platform of the drone and is used to conduct multiple area covering tasks collaboratively. The GV takes the drone to visit a set of discrete areas, while the drone takes off from the GV at potential nodes around each area and scans each area for collecting information. The drone can be recharged in the GV during the time when it travels between different areas. The objective is to optimize the drone's scanning path for all areas' coverage and the GV's travel path for visiting all areas. A 0-1 integer programming model is developed to formulate the problem. A two-stage heuristic based on cost saving strategy is designed to quickly construct a feasible solution, then the Adaptive Large Neighborhood Search (ALNS) algorithm is employed to improve the quality of the solution. A simulation experiment based on the parks in Changsha, China, is presented to illustrate the application of the method. Random instances are designed to further test the performance of the proposed algorithm.**

***Index Terms*—adaptive large neighborhood search, drone, heuristic, multi-area coverage, routing**

I. Introduction

In many situations, the detail information of a whole area is required, like agricultural monitoring, terrain mapping, mineral exploration, etc. Remote sensing is one of traditional methods to cover an area for collection information, e.g. monitor crop growth [1]. However, this method not only takes high cost, high risk and low time efficiency, but also is severely limited by the poor revisiting time and limited spatial variability [2]. Moreover, manned airborne platforms would also be applied, while not widely used due to high cost, complex operations and lengthy delivery of products [3]. These problems drive many researchers to look for more efficient ways and thus show great interests in area coverage with drones, also known as unmanned aerial vehicles (UAVs).

With the development of automation and artificial intelligence technologies, drones have shown their great superiority in area coverage. It is reported that drones have been applied in battlefield information collection [4] and agricultural information collection [5]. Compared with traditional area coverage ways, e.g. ground robots, drones have the advantages of short time-consuming, low cost and high precision. Besides, unlike satellite, the drone can fly at low altitudes and is less susceptible to weather. However, when they are used to cover multiple discrete areas or large areas, the efficiency may decrease due to its short endurance. It is also inappropriate to waste too much energy of the drone on transform between different areas.

A potential way to overcome the obstacle caused by the drone's limited endurance is employing the ground vehicle (GV) to work together with the drone. Liu et al. [6] established a mathematical model for cooperated GV and drone to perform Intelligence, Surveillance and Reconnaissance (ISR) missions on nodes, and showed that the efficiency of ISR tasks can be significantly improved. To enlarge the task range of the drone for covering multiple areas, the GV can be employed as the moving platform of the drone, which takes the drone from one area to another. The drone can be recharged in the GV when it travels between areas. The drone takes off from the GV when it arrives around the covered area, and returns to the GV when the drone completes the coverage of the area. The combination of GV and the drone can significantly increase coverage efficiency and reduce the operational cost. At the International Consumer Electronics Show (CES) in 2016, DJI and Ford Motor Company launched a new field search and rescue model [7], where the drone carried by the Ford vehicle is suitable for large-scale search and rescue operations in uninhabited areas due to their low cost and rapid deployment. The DJI drone can take off from a Ford F-150 truck, and land at it after completing the search of an area. The cooperation of GV and drone has shown significant advantages in the execution of such tasks.

Although the GV-drone mode has great advantages, it also brings new challenges to the path planning problem. First, there are two-layer paths where the drone's paths in the upper layer are connected with the GV's path in the lower layer in different nodes. The GV starts from and returns to the depot after visiting all areas, forming the lower layer path, which is close form. When the GV arrives around each area, the drone takes off from

Manuscript received XXXX. This work was supported in part by the National Natural Science Foundation of China under Grant 71771215, and in part by the Hunan Provincial Natural Science Fund for Distinguished Young Scholars under Grant 2018JJ1035. (Corresponding author: Jianmai Shi.)

Y. Xia, C. Chen, J. Shi, Y. Liu and G Li are with the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China (e-mail: xia_yang_sheng@163.com; changshacc@163.com; jianmaishi@gmail.com; liuyao13@nudt.edu.cn; gli2010a@163.com).

the GV and flies to scan the area, then return to the GV after the whole area is covered, which forms the upper layer paths. The flying time may quite different when different take-off and landing nodes of the drone are selected, which also impact the travel path of the GV. These two-layer paths are connected together and impact each other, which makes the problem more complex than traditional routing problems.

Motivated both by the adaption of new technologies in practical industry, and the theoretical gap existing in the current literature, a novel problem named as Two-Layer Path Planning Problem for Multi-Area Coverage by cooperated GV and Drone (2L3P-MAC-G&D) is investigated. The problem can be viewed as a path selection problem on two connected networks. The upper network of the potential paths for the drone and the lower network of the potential roads for the GV are connected, as the drone has to take off and land at the GV frequently. We build a 0-1 integer programming model based on the characteristics of two-lager path planning problem. Then a two-stage heuristic based on saving strategies is proposed to construct the travel path of GV and the flying path of the drone. The adaptive large neighborhood search (ALNS) algorithm is designed to further improve the quality of the solution. Both practical case and random instances are used to test the proposed algorithms, and computational results show that our approach can efficiently solve the problem.

The paper is structured as follows. Section 2 presents the literature review, and Section 3 illustrates the problem and the model. Section 4 proposes the design of the two-stage heuristic, while Section 5 presents the ALNS algorithm. The experiments and results are reported in Section 6. At last, Section 7 concludes the work.

II. LITERATURE REVIEW

The area coverage issues by drones has great application prospects in real life, such as assessment of damage in disaster areas [8], observation of crop growth in agricultural production [9], and locate the position of all sensors in a certain area [10]. The problem usually consists of three types. The first type is path planning for full coverage, where all points in the area are visited at least once. For example, Carvalho [11] designed several template routes for the cleaning robot to completely clean the factory floor. When the robot encounters different situations, the appropriate template is used for path planning. The second type is path planning for partial coverage, where the drone collects as much information as possible in a limited time. The objective function is usually set as maximizing the coverage area within a limited time [12]. Another common objective function is to maximize the value of collected information [13]. The third type is to cover from stationary positions, which is continuous observation of certain areas or objects. When the observation object is a certain area, the fixed position is usually unchanged, like emergency response, environmental monitoring, traffic network monitoring and so on [14]. When the observation object is a moving object in the area, the fixed position needs to change in real time with the moving object to ensure the relative invariability. Pugliese et. al [15] investigated the drone's covering problem for tracking and monitoring of all moving or stationary objects in a given area.

The area coverage problem studied in this paper belongs to the first type. Considering the disadvantages of traditional area coverage methods, companies are currently using drone and sensor technologies to acquire information of ground regions and to reduce the time and costs of operations. The drone's path planning problem for area coverage is to plan the flight path to cover all points in an area at the lowest possible cost [16], therefore, drone's flight path planning methods for fixed target points have reference significance for drone's area coverage path planning. He and Zhao [17] compares the performance on online real-time path planning abilities for four algorithms in different applying situations of drones, and the computational results indicate that the Dijkstra algorithm performs the best, Guan et al [18] proposed a drone's path planning algorithm based on double ant colony, through using genetic algorithm to generate pheromone in the early stage, the convergence of ant colony algorithm was improved. Avellar et al [19] reviewed the civil applications of the drone and presented a comprehensive illustration on the covering problems by drones. In order to simulate a more realistic environment, the impact of the wind factor is taken into consideration in the path planning for area coverage by drones [20]. In the coverage mission, the drone has a deceleration adjustment process during the turn, which consumes extra time and power [21, 22]. Therefore, the time spent in the turn process should also be considered for estimating the flying time. Ideally, the shortest path has the lowest number of turns [23]. In current area covering problems by the drone, most of them investigated a single area covered by only drones, which did not involve GVs. However, in many practical applications, it is often required to cover multiple areas, where the GV has to be introduced for improving the drone's operational range.

The cooperated GV and drone has a wide range of applications, such as intelligence collection, detection and surveillance of specific targets [24], delivery of goods [25], and power line inspections [26]. The area coverage issues by drone belong to the field of information collection, Tokekar et al [27] combined the advantages of ground vehicles and drones to design a system for collecting nitrogen content at specific points on the farm, which can help farmers effectively reduce fertilizer use. Ropero et al [28] presents an approach based on the coordination of a hybrid UGV-UAV system to explore a planetary surface. Liu et al [29] proposes a hierarchical UAV/UGV platform to detect wildfire. A set of useful strategies are obtained for the routing problem of cooperated GV and drone, but all these studies investigated how to efficiently access specific points instead of covering the entire area.

Until now, there is no literature studies how to use cooperated GV and drone to cover multiple areas. In this paper, the drone's scanning paths for multiple areas and the travel path of the GV are planned at the same time. Due to the differences in speed, endurance time and driving mode between GVs and drones, there is a complex reciprocal effect between the two kinds of paths in the planning process. Focusing on this difficulty, we propose a heuristic method to optimize the GV and drone's paths efficiently.

III. Problem Description and Modeling

A. Problem Description

In the two-layer path planning problem for multi-area coverage by cooperated GV and drone (2E3P-MAC-G&D), one GV takes one drone to corporately complete the multi-area scanning mission. The road network and the regions of all areas for covering are known. The endurance of the GV is sufficient for the whole mission, while the drone's endurance is limited and not enough to scan all areas. The GV serves as the moving station of the drone, and can recharge and change battery for the drone. The drone can only take off and land on the GV when it stops at some predetermined nodes, noted as stopping nodes. The GV travels on the road network, and launches out the drone when it arrives around an area. The drone conducts a scanning task to fully cover the area, and return to the GV. Fig. 1 shows an illustration of feasible solution for an example of 2E3P-MAC-G&D. It can be seen that there are paths in two layers, which are the lower-layer path on the ground traveled by the GV and the upper-layer paths in the air traveled by the drone.

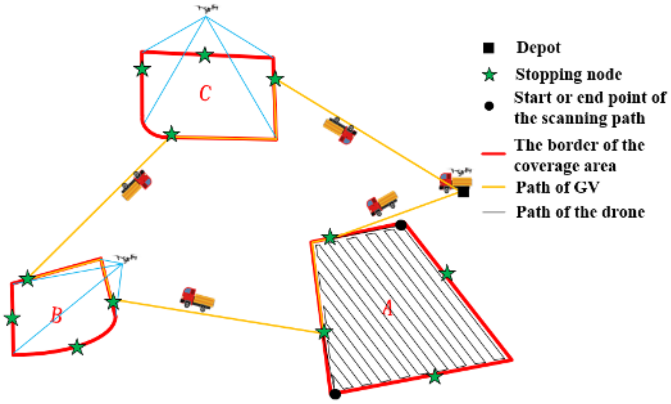

■ Depot
★ Stopping node
● Start or end point of the scanning path
The border of the coverage area
Path of GV
Path of the drone
C
B
A

Fig. 1. An illustration of the solution for an example of 2E3P-AC-G&D

In 2E3P-MAC-G&D, for completing the covering task, the GV starts from the depot, taking the drone, e.g. to scan the three areas, A, B, and C in Fig.1. After reaching around the vicinity of the area, the GV selects an appropriate stopping node according to the area scanning path of the drone and release it. Then the drone flies to the area and conducts the coverage task along the scanning path. At the same time, the GV can move to the next stopping node to pick up the drone when it completes the coverage task. After completing the whole mission, the GV and drone must return to the depot. The problem is to optimize the paths of the GV and the drone to complete the covering of all areas in a minimal time, while the drone's capacity constraints on battery are not violated.

B. Model Development

For modelling 2E3P-MAC-G&D, the notations used are summarized in TABLE 1.

TABLE I
NOTATION AND ITS IMPLICATIONS

Notation	Implications
Sets:	
G	the undirected graph where the problem is defined;
E	the set of all arcs;
S	= {1,2,…,m},the set of all areas that should be covered;
V	the set of vertices of all areas;
V_S	the set of vertices around areas *s*, where $s \in S$;
N_0	= {0}, the depot;
N	={1, 2, …, n},the set of all stopping nodes around every areas;
N_S	the set of stopping nodes around areas *s*, where $s \in S$;
C_S	the set of stopping nodes selected for use in areas *s*, where $s \in S$;
P_S	the set of start and end points of scanning paths for areas *s*, where $s \in S$;
P_s^k	the set of start and end points of the *k*-th scanning path for area *s*, where $s \in S$;
R	the set of arcs that can be traveled by GV;
Us	the set of scanning paths in area s, where $s \in S$;
Parameters:	
L_{ij}^F	the Floyd distance from point *i* to point *j*, where $i, j \in N_0 \cup N$;
l_s^k	the length of the *k*-th scanning path in covered area *s*, where $s \in S$;
T	the total time required to complete the task;
v^G	the average speed of the GV;
v^D	the average speed of the drone;
T_{max}	the maximum endurance time of the drone;
Variables:	
x_{ij}	1 if the GV travels from point *i* to point *j*, 0 otherwise, where $s \in S$;
y_{ijs}	1 if the drone flies from point *i* to point *j*, 0 otherwise, where $s \in S$.

More formally, the 2E-ACRP-G&U can be described as follows.

Minimize

$$Z = \frac{\sum_{i,j \in N \bigcup N_0} L_{ij}^F x_{ij}}{v^G} + \sum_{s \in S} \max\left(\frac{\sum_{i,j \in N_s} L_{ij}^F x_{ij}}{v^G}, \frac{\sum_{i,j \in N_s \bigcup P_s} L_{ij}^F y_{ijs} + \sum_{k \in U_s} l_s^k \sum_{i \in P_s, j \in P_s^k} y_{ijs}}{v^D}\right) \tag{1}$$

s.t.

$$\sum_{j \in N} x_{0j} = \sum_{i = N} x_{i0} = 1, \tag{2}$$

$$\sum_{i \in N, i \neq j} x_{ij} = \sum_{i \in N, i \neq j} x_{ji}, \tag{3}$$

$$\sum_{i \in N / N_s} \sum_{j \in N_s} x_{ij} = \sum_{j \in N_s} \sum_{i \in N / N_s} x_{ji} = 1, \quad \forall s \in S, \tag{4}$$

$$\sum_{i \in P_s} y_{jis} \leq \sum_{i' \in N / N_s} x_{i'j}, \quad \forall j \in N_s, s \in S, \tag{5}$$

$$\sum_{i \in P_s} y_{ijs} \leq \sum_{i' \in N / N_s} x_{ji'}, \quad \forall j \in N_s, s \in S, \tag{6}$$

$$\sum_{j \in N_s} \sum_{i \in P_s} y_{jis} = \sum_{j \in N_s} \sum_{i \in P_s} y_{ijs} = 1, \quad \forall s \in S, \tag{7}$$

$$\sum_{j \in N_s} \sum_{i \in P_s^k} y_{jis} = \sum_{j \in N_s} \sum_{i' \in P_s^k \text{ and } i \neq i'} y_{i'js}, \quad k \in U_s, s \in S, \tag{8}$$

$$\frac{\sum_{i,j \in N_s \bigcup P_s} L_{ij}^f y_{ijs} + \sum_{k \in U_s} l_s^k \sum_{i \in P_s, j \in P_s^k} y_{ijs}}{v_2} \leq T_{\max}, \forall s \in S, \tag{9}$$

$$x_{ij} \in \{0,1\}, \forall i, j \in N \bigcup \{0\}, \tag{10}$$

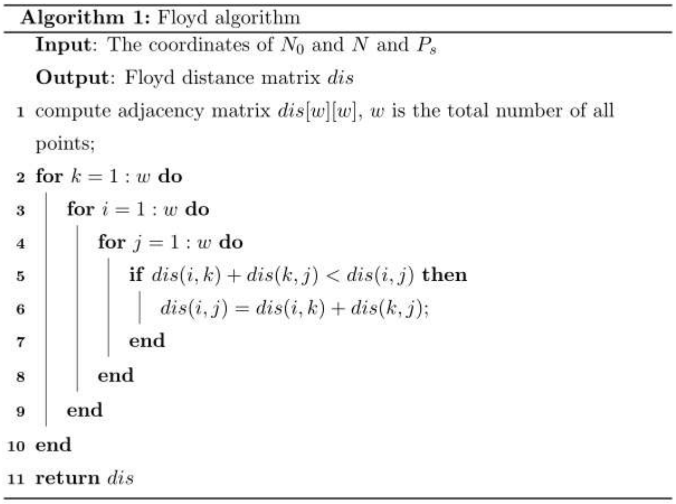
$$y_{ijs} \in \{0,1\}, \forall i, j \in N \bigcup P_s, \forall s \in S, \tag{11}$$

The objective function (1) minimizes the total time given by two main components, where the first one is the total time for the cooperated GV and drone to transfer among all areas and the depot, and the other is the total time for the drone to complete the coverage of all areas.

In order to reduce the time of the first part, one of the criteria for selecting the stopping nodes around different covered areas is to choose the ones with shortest Floyd distances from each other. To reduce the time of the second part, on the one hand, we should choose the stopping nodes with shortest Floyd distance from each other around the same area, and on the other hand, we should choose suitable scanning pattern to make the drone's total flight distance shorter.

Constraints (2) ensure that the cooperated GV and drone start from and return to the same depot. Constraints (3) guarantee that the number of outgoing arcs equals to the number of incoming arcs at each stopping node, which ensure the connectivity of the GV's route. Constraints (4) restrict that the GV enters each area once and leave the area once, that is each area can only be visited once. Constraints (5) ensure that the drone can only take off at the node visited by the GV, while constraints (6) guarantee that the drone can only land at the node visited by the GV. Constraints (7) ensure that the drone take off once and land once at each area. Constraints (8) ensure the connectivity of the drone's flying path at each area, and together with (7) guarantee that only one scanning path be selected at each area. Constraints (9) ensure the flight time of the drone cannot exceed its endurance when it flies for scanning each area. Constraints (10) and (11) define the 0-1 variables.

L_{ij}^{F} represents the Floyd distance from point i to point j, which is calculated by Floyd algorithm. Floyd algorithm, also called interpolation method, is an algorithm that uses the idea of dynamic programming to find the shortest path between multiple source points in a given weighted graph, which is similar to Dijkstra's algorithm. The main procedure of Floyd algorithm is as follows:



```
Algorithm 1: Floyd algorithm
  Input: The coordinates of $N_0$ and $N$ and $P_s$
  Output: Floyd distance matrix $dis$
1 compute adjacency matrix $dis[w][w]$, $w$ is the total number of all
  points;
2 for $k = 1 : w$ do
3     for $i = 1 : w$ do
4         for $j = 1 : w$ do
5             if $dis(i,k) + dis(k,j) < dis(i,j)$ then
6                 $dis(i,j) = dis(i,k) + dis(k,j)$;
7             end
8         end
9     end
10 end
11 return $dis$
```

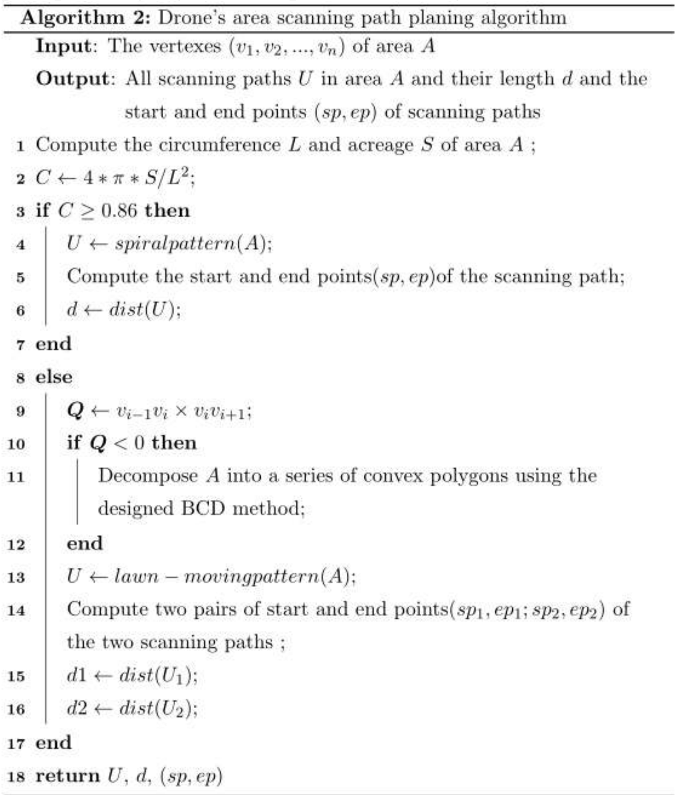
First the position information of all points is provided. Then the adjacency matrix of all points is computed in Line 1, and let $dis(i\ ,j)$ represent the straight line distance from point i to point j. If $i = j$, the distance is 0. If point i cannot be reached from point j directly, the distance is set to be infinity. Finally, we obtain the Floyd distance matrix through a triple loop (Line 2-10).

IV. TWO-STAGE HEURISTIC BASED ON SAVING STRATEGIES

Based on saving strategies, a two-stage constructive heuristic algorithm (TSH) is designed to obtain a better feasible solution for the problem. The first step is to plan the drone's scanning path in the targeted area. The second step is to plan the route of GV after the scanning path is determined.

A. Stage 1: Path Planning Algorithm for The Drone

When an area is given, the optimal scanning path of the drone can be optimized. First, we should check the shape of the areas, and the areas with concave shape be decomposed. Then the appropriate scanning pattern for the drone is selected. Finally, the scanning path of the area is calculated based on the performance of the drone. The main procedure of the drone's path planning algorithm is presented in Algorithm 2.



```
Algorithm 2: Drone's area scanning path planing algorithm
  Input: The vertexes $(v_1, v_2, ..., v_n)$ of area $A$
  Output: All scanning paths $U$ in area $A$ and their length $d$ and the
          start and end points $(sp, ep)$ of scanning paths
1 Compute the circumference $L$ and acreage $S$ of area $A$ ;
2 $C \leftarrow 4 * \pi * S/L^2$;
3 if $C \geq 0.86$ then
4     $U \leftarrow spiralpattern(A)$;
5     Compute the start and end points$(sp, ep)$of the scanning path;
6     $d \leftarrow dist(U)$;
7 end
8 else
9     $Q \leftarrow v_{i-1}v_i \times v_iv_{i+1}$;
10    if $Q < 0$ then
11        Decompose $A$ into a series of convex polygons using the
          designed BCD method;
12    end
13    $U \leftarrow lawn - movingpattern(A)$;
14    Compute two pairs of start and end points$(sp_1, ep_1; sp_2, ep_2)$ of
      the two scanning paths ;
15    $d1 \leftarrow dist(U_1)$;
16    $d2 \leftarrow dist(U_2)$;
17 end
18 return $U$, $d$, $(sp, ep)$
```

The geometric information of the covered area, the potential stopping nodes, and the flying speed of the drone are provided. Then the roundness of the area is calculated (Line 2). If the roundness of the area is greater than 0.86, the spiral pattern is adopted, and we obtain the scanning path (Line 4), the start and end points of the path (Line 5). The length of the scanning path is calculated in Line 6. If the roundness is lower, e.g. less than 0.86, the lawn mowing pattern is adopted. In this case, the concavity of the area is first checked (Line 9). If the covered area is a concave polygon (Line 10), we decompose it into a series of convex polygons by using the BCD method designed below (Line 11), and then plan the Boustrophedon path of the covered area. Starting along the direction parallel to the long axis, there are two scanning paths (Line 13), corresponding to two pairs of start and end points (Line 14). The lengths of the two paths are calculated (Lines 15-16).

1) Selection of Scanning Pattern

There are two main scanning patterns for the drone to scan an area. One is lawn mowing pattern [30] and the other is spiral pattern [31]. As shown in the Fig. 2, for the same area, the scanning paths with two scanning patterns are quite different.

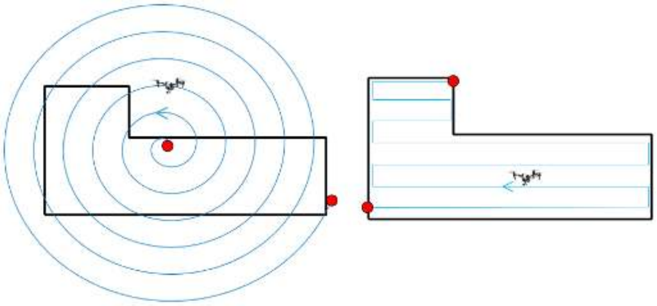


Fig. 2. Spiral pattern and lawn mowing pattern

In Fig. 2, the polygon surrounded by black border indicates the covered area, the red points indicate the start and end points of the scanning paths, the blue tracks represent the flying paths, and the arrow represents the flying direction of the drone. The two scanning patterns are specifically defined as follows.

Spiral pattern: The drone starts from the center of the covered area and then spirally scans the entire area at a constant spiral pitch until the entire area is covered.

Lawn mowing pattern: The drone scans the area at equal intervals parallel to the long axis direction of the area until it covers the entire area.

For the spiral pattern, there is no need to consider the turning process of the drone, but there are many turning operations in the lawn mowing pattern where braking and acceleration have to be considered. Dubins path is the shortest curve that connects two points in a 2D plane with a constraint on the curvature [32], and thus we employ the Dubins path to plan the turning paths of the drone and estimated the flying time in the turning process. In the straight paths, the average speed is used to estimate the flying time.

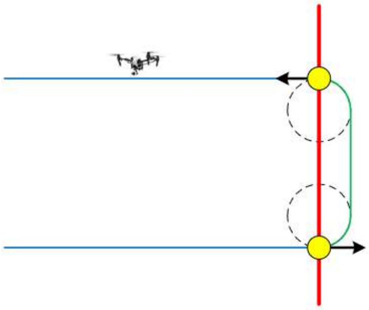


Fig. 3. Dubins turning path of the drone

Fig. 3 presents a detail illustration for a turning path in the lawn mowing pattern. In the research of this paper, the scanning width of the drone is larger than its turning radius, and thus all the turning trajectories are similar. Therefore, the scanning path of the drone is composed of the scanning line segments parallel to the long axis of the polygon and the turning trajectories with the similar shape. The minimum radius of the circle tangent to the drone's trajectory can be calculated as follows：

$$Rad = \frac{v^D}{\psi}, \tag{12}$$

where v^D is the flying speed of the drone and ψ is maximum turn rate of the drone.

For simplification, in other figures the detail turning processes of all paths in lawn mowing pattern are not shown.

Most of the literature uses lawn mowing pattern [33, 34], and very few uses spiral pattern. So far, few works investigated how to choose different scanning patterns, while different scanning patterns have a great impact on the overall length of the scanning path. For example, in Fig. 2, in order to cover the area entirely, a large part of the flying path with the spiral pattern is outside the area, which makes this path significantly longer than that with the lawn mowing pattern. The more irregular the area is, the more time the lawn mowing scanning pattern can save. However, it is conceivable that the closer the area is to the circle, the better the spiral pattern is. Here we choose different scanning patterns depending on how close the area is to the circle. We usually describe how close a polygon is to a circle based on the degree of roundness. The definition and calculation method of a polygon's roundness have been studied in literatures [35-37]. Bai [37] proposed a method for calculating the roundness of traffic signs, which is suitable for the polygonal roundness calculation in this paper. Therefore, we use this method to calculate the roundness of the polygon, and the formula is as follows:

$$C = \frac{4 * \pi * S}{L^2}, \quad (0 < C \le 1) \tag{13}$$

Where S is the acreage of the area, and L is the circumference. If the area is closer to a circle, its roundness, C, is closer to 1.

The two typical scanning patterns are utilized to cover polygons with different roundness, and the scanning time of the two patterns is compared to analyze the relationship between polygon roundness and scanning pattern selection. Let the scanning width of the drone to 2 units, and the speed is 0.1 unit per second, set the maximum turn rate of the drone be 0.7. Set the acreage of all the polygons be 84 units. The scanning time of different polygons under the two scanning patterns is presented in TABLE 2.

TABLE 2
SCANNING TIME FOR TYPICAL POLYGONS BY THE TWO SCANNING PATTERNS

Polygon	Roundness	Time of spiral pattern (s)	Time of lawn mowing pattern (s)
	0.60	1281.4	732.22
	0.79	1252.6	1126.41
	0.86	1210.0	1224.10
	0.91	1195.9	1291.36
	0.96	1140.6	1336.42

In TABLE 2, it can be seen that when the roundness of a polygon is small, the scanning time of lawn mowing pattern is significantly less than that of spiral pattern. As the roundness of the polygon becomes larger, the gap between them is getting smaller. The scanning time of the two patterns is basically the same when it is a regular pentagon. When the roundness of the polygon continues to increase, the time of spiral pattern becomes less than that of the lawn mowing pattern. When the roundness is close to that of the regular pentagon (C=0.86), both scanning patterns have similar effects. When the roundness of the area is smaller than 0.86, we choose the lawn mowing pattern. When the roundness is larger than the roundness of the regular pentagon, it better to choose the spiral pattern.

2) Judgment and Decomposition of Concave Polygon

When the spiral pattern is adopted, the concavity and convexity of the polygon has little effect on it. However, when the lawn mowing pattern is adopted, the influence is large. The previous lawn mowing pattern is based on the fact that the covered area is a convex polygon. In some situations, the area is a concave polygon, and then the concave polygon has to be decomposed into a set of convex subregions.

(1) Judgment of the concavity

There are many ways to judge the concavity of a polygon. Here we use cross product of vectors to judge the concavity of an area. The principle is that every vertex of a convex polygon should have the same turning, that is, every vertex of a convex polygon should be a convex point. The one with different turning should be a concave point, that is, the polygon with concave points should be a concave polygon. Assume the polygon *P* have *n* vertices $(v_1, v_2, ..., v_n)$, and the concavity of *P* is determined by the concavity of each vertex belonging to *P*. *P* is shown in Fig. 4.

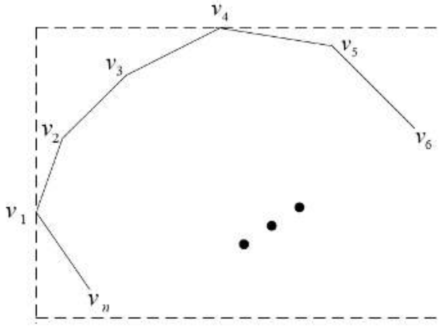


Fig. 4. Polygon *P*

Let v_i be a random vertex of *P*, v_{i-1} and v_{i+1} be two adjacent vertices of v_i. The concavity of vertex v_i is determined by the cross product of vectors *Q*.

$$Q = v_{i-1}v_i \times v_i v_{i+1} = \begin{cases} >0, & v_i \text{ is a convex vertex} \\ =0, & v_i \text{ is a collinear vertex (convex vertex)} \\ <0, & v_i \text{ is a concave vertex} \end{cases} \quad (14)$$

If all vertexes of the polygon *P* are convex vertex, then *P* is a convex polygon, otherwise *P* is a concave polygon.

(2) Decomposition of concave polygon

There are many methods for decomposing concave polygons into convex polygons, such as trapezoidal decomposition [38], triangular decomposition [39], and approximate decomposition [40]. The path generated by lawn mowing pattern is usually called Boustrophedon path. Decomposing a concave polygon into a series of convex polygons, and then plan the Boustrophedon path for each convex polygon is called Boustrophedon Cellular Decomposition (BCD) [41]. In most literatures on BCD related methods, the trapezoidal decomposition is usually employed. Li et al [42] proposed an accurate BCD method based on trapezoidal decomposition with an algorithm complexity of O(n). In [18], the concave polygon's convex hull is used to expand BCD into external cell decomposition, which effectively reduces the number of decomposed convex polygons. The BCD method designed in this paper is also based on trapezoidal decomposition. Since the effect of decomposing concave polygons along different angles is different, in order to reduce the number of decomposing convex polygons and the number of turns of the drone, we decompose the concave polygons along the direction parallel to the long axis of the concave polygons.

The long axis of a convex polygon is defined as follows. For each side of a convex polygon, the distance from the other vertices that do not belong to the side is solved separately. Note the maximum of these distances as the span of this side. The spans of all sides are compared, and the corresponding side with the minimum span is the long axis of the convex polygon. In order to find the long axis of the concave polygon, we use the convex hull generation algorithm proposed in [43] to first generate a concave polygon's convex hull. A convex hull is the simplest convex polygon that contains all vertices of the concave polygon. The algorithm is summarized below.

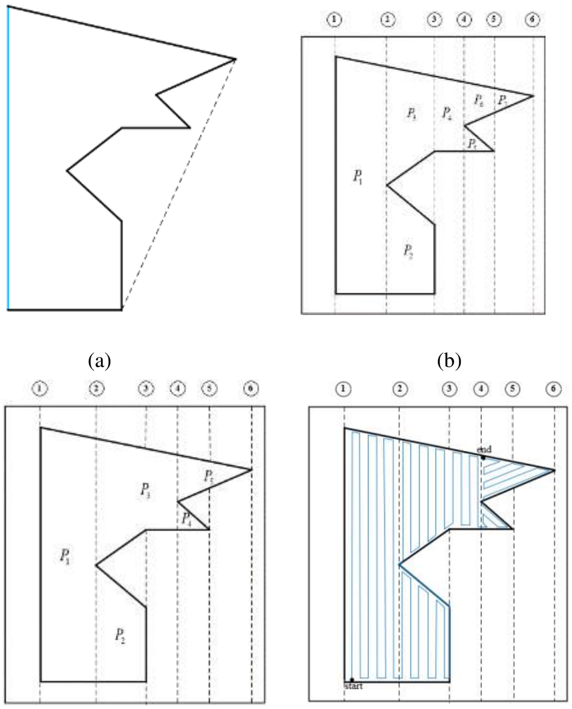
1) Find the extremal points and delete all other points falling inside the polygon they form, break the remaining set of points into 4 regions.

2) Sort the remaining points on their x-coordinate in ascending order for regions 1 and 2, and in descending order for regions 3 and 4.

3) For each region, find the convex paths from one extremal point to the other.

Then the long axis of the convex hull is viewed as the long axis of the concave polygon. After the concave polygon is trapezoidal decomposed along the parallel with the long axis, the Boustrophedon path is generated by using lawn mowing pattern.

The figure enclosed by the solid line in (a) of Fig. 5 is an example concave polygon to be decomposed, and after adding the dotted line, it becomes a convex polygon, where the blue side represents the long axis of the convex polygon and is also the long axis of the concave polygon. Fig. 5(b) indicates that the concave polygon is trapezoidal decomposed in a direction parallel to the long axis of the concave polygon, while Fig. 5(c) presents the process that a series of convex polygons after trapezoidal decomposition are merged. When two adjacent convex polygons have one edge that coincide with each other and the long axes of the two are parallel to each other, they can be merged. This process can effectively reduce the number of convex polygons and avoid unnecessary transfer of the drone between different convex polygons. Fig. 5(d) shows the Boustrophedon path generated using the lawn mowing pattern.



(a) (b)

(c) (d)

Fig. 5. Concave polygon's BCD decomposition and Boustrophedon path generation process

B. Stage 2: Route Planning of GV

The scanning path of an area by the drone and its start and end points can be obtained by the above path planning algorithm. Based on these points, we further reduce drone's flight time by choosing the appropriate stopping nodes around the area. At the same time, the travelling route of GV is also optimized through saving strategies. Combining these two aspects we develop the path planning algorithm of GV as shown in Algorithm 3.

Compute Floyd distance matrix dis based on the position information of all the points (Line 1), then randomly select a scanning path for each area (Line 2). We obtain the start and end points of the path (Line 3). Two random stopping nodes are selected in each area, and then the distance for the drone from the stopping node for taking off to the start point of the scanning path and the distance from the end points of the scanning path to the stopping node for landing is calculated, and Dis is the sum of the two distance (Line 4-10). In every area we find the two stopping nodes corresponding to the shortest distance Dis as the selected stopping nodes (Line 12) to form the goal (Line 14), then calculate the saving value matrix (Line 15-19), and arrange it in descending order (Line 20). Starting from the maximum value, the corresponding two areas are connected until all areas are included, and the initial feasible solution is obtained (Line 21).

```
Algorithm 3: Heuristic algorithm based on saving strategy
  Input: The position information of the depot $N_0$ and all stopping
         nodes $N$ and $N_S$ and the start and end points of all scanning
         paths
  Output: Initial feasible solution $s$
 1 Compute the Floyd distance matrix $dis$ according to the Floyd
   algorithm;
 2 $Route_i \leftarrow randomselect(area_i), i = 1, 2, 3, ..., m$;
 3 $p_i^1, p_i^2 \in Route_i$;
 4 for $i = 1 : m$ do
 5     for $k \in N_i$ do
 6         for $t \in N_i$ do
 7             $Dis(i, k, t) = dis(k, p_i^1) + dis(p_i^2, t)$;
 8         end
 9     end
10 end
11 for $i = 1 : m$ do
12     Find the $k$,$t$ corresponding to the smallest $Dis(i, k, t)$ to form
       $Choice_i$;
13 end
14 $goal \leftarrow (Choice_1, Choice_2, ..., Choice_m)$;
15 for $i = 1 : (m - 1)$ do
16     for $j = i + 1 : m$ do
17         $jieyue(i, j) \leftarrow dis(N_0, goal(i, 2)) + dis(N_0, goal(j, 1)) -$
           $dis(goal(i, 2), goal(j, 1))$-;
18     end
19 end
20 $W \leftarrow descendsort(jieyue)$;
21 Starting from the $W(1)$, the corresponding two area numbers are
   connected until all areas are included,and $s$ is obtained;
22 return $s$
```

V. ADAPTIVE LARGE NEIGHBORHOOD SEARCH ALGORITHM

The two-stage heuristic can present a better feasible solution for the problem, which can be further improved. In this section, an ALNS algorithm is employed to optimize the problem based on the initial solution.

The ALNS algorithm has a good effect on solving the problem that the search field grows exponentially with the size of the searched data, which allows multiple neighborhoods to be searched and the neighborhood search operators to be utilized dynamically based on the quality of their obtained solutions. The algorithm has a good effect in solving the double-layer cooperative routing problem [44-47]. According to the special structure of the solution for 2E3P-MAC-G&D, some improvements were made to the traditional ALNS. The stopping nodes adjusted by the destroy and repair operators in each round of the calculation belongs to the same area. By choosing different paths of the drone, we can get different start and end points of the paths, which also affects the planning of the entire route. Based on these differences, we designed the ALNS algorithm as shown in Algorithm 4.

```
Algorithm 4: Adaptive large neighborhood search algorithm
  Input: feasible solution $s$;
         the neighborhood operator set $\Omega$;
         the weight of each operator $w_i$
  Output: optimal solution $s^b$
1  $s^b \leftarrow s$;
2  repeat
3      for $i = 1 : N$ do
4          select neighborhood operator $dr \in \Omega$ using $w_i$;
5          $s^t \leftarrow dr(s)$;
6          if $accept(s^t)$ then
7              $s \leftarrow s^t$;
8              if $t(s^t) < t(s^b)$ then
9                  $s^b \leftarrow s^t$;
10             end
11         end
12     end
13     update $w_i$;
14 until stop criterion is met;
15 return $s^b$
```

First, we input the initial feasible solution s which is obtained through the two-stage heuristic, the neighborhood operators, and the weight of each operator. The solution s is assigned to the current optimal solution s^b (Line 1). Set the number of operations in each round to *N*. In each round, different neighborhood operators are selected according to their weights (Line 5), s^t represents the solution obtained by the selected operator. If the solution s^t meets the acceptance criteria (Line 6), set s^t as the search starting point in the next around. If the current solution is better than s^b, the optimal solution is updated (Line 8-10). After each round of the iterations is finished, the weights of all neighborhood operators are updated according to their performance (Line 13). Based on the updated weight, the next round of calculation is performed until the termination condition is met.

A. Design of Neighborhood Operators

From Equation (1) we can see that there are two main aspects to optimizing the initial feasible solution. The first is to reduce the flying time of the drone. Here, it is mainly achieved by choosing different scanning paths and selecting the appropriate stopping nodes of GV that can reduce the distance of the drone from the GV to the start point of the scanning path and the distance from the end point of the scanning path to the landing node. The second is to reduce the travelling time of the GV, including the transfer time between different covered areas, the transfer time between covered area and the depot, and the transfer time between different stopping nodes around the same covered area. For reducing the second kind of time, it is mainly achieved by optimizing the route of the GV to visit all covered areas and the stopping nodes around each area. Based on the above observation, the neighborhood operators are designed through combining different destroy and repair operators. The destroy operator is to delete some nodes and/or covered areas in a feasible solution, while the repair operator is to reinsert these nodes and/or areas into the solution to generate a new feasible solution.

First, randomly selecting a covered area in the current feasible solution, there are two situations for the drone's path: Situation 1 where the drone takes off and lands at different nodes, and Situation 2 where the drone takes off and lands at the same node. The destroy and repair operators are designed as follows.

Destroy operators

(a) Delete one of the stopping nodes in Situation 1.

(b) Delete the two stopping nodes in Situation 1.

(c) Delete the only one stopping node in Situation 2.

(d) For Situation 2, remove the taking off arc or the landing arc while keeping the stopping node in use.

To repair the drone's path destroyed by operator a, the following repair operators can be used.

(e) Generate a feasible path as Situation 2 for the drone using the remained stopping nodes.

(f) Randomly select one of the unused stopping nodes for the drone to take off/land.

(g) Select the node with the shortest Floyd distance among the unused stopping nodes for the drone to take off/land.

(h) Add one of the unused stopping nodes that is with the shortest Floyd distance to the previous area (or depot) visited by the GV.

(i) Add one of the unused stopping nodes that is with the shortest Floyd distance to the next area (or depot) visited by the GV.

(j) Add one of the unused stopping nodes that is with the shortest total Floyd distance to the previous and next areas visited by the GV.

To repair the drone's path destroyed by operators b or c, the following repair operators can be used.

(k) Randomly select an unused stopping node to generate a path as Situation 2.

(l) Select two stopping nodes among the unused nodes one by one through randomly employing repair operators f, g, h, i, and j.

To repair the drone's path destroyed by operator d, the repair operators f, g, h, i, and j can be used.

Considering the Constraints (4) in the model, a number of potential neighborhood operators can be obtained through combining the above destroy and repair operators.

For Situation 1, part of the adjustment process of the stopping nodes in the covered area by neighborhood operators can be shown in Fig. 6. It shows that we first delete one of the stopping nodes, then generate a feasible path as Situation 2 for the drone using the remained stopping nodes or add one of the unused stopping nodes.

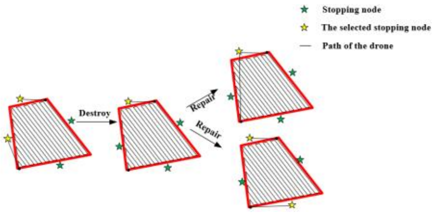


Fig. 6. The adjustment process of the stopping nodes in the covered area by neighborhood operators for Situation 1

For Situation 2, part of the adjustment process of the stopping nodes in the covered area by neighborhood operators can be shown in Fig. 7. It shows that we first delete the only one stopping node, then add one or two unused stopping nodes. We can also remove the taking off arc or the landing arc while keeping the stopping node in use, then add one unused stopping node.

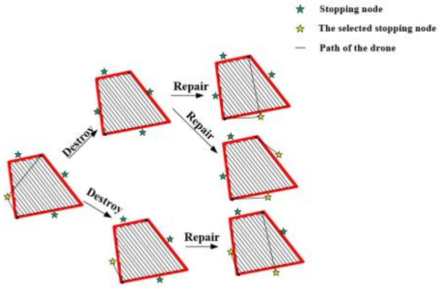


Fig. 7. The adjustment process of the stopping nodes in the covered area by neighborhood operators for Situation 2

From the two-stage heuristic, we can know that choosing different scanning paths will obtain different start and end points, and thus affect the selection of stopping nodes. If the selected area has two scanning paths, we can also adjust the initial solution by choosing different scanning paths. Fig. 8 shows the change of the selected stopping nodes after selecting different scanning paths. This operation is also a neighborhood operator and can be used in the above two cases.

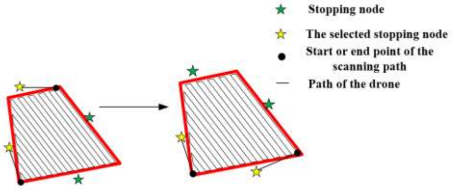

Stopping node
The selected stopping node
Start or end point of the scanning path
Path of the drone

Fig. 8. The change of the selected stopping nodes after selecting different scanning paths

To reduce the travelling time of the GV, the two and three exchange operators are utilized for changing the visiting sequence of all areas.

B. Update of Weights

The effects of the neighborhood operators may quite different in different search stages of the large neighborhood search process. The utilizing frequencies of these operators are adaptively adjusted in the iteration process. In the beginning, the same weight is set for these operators $w_i = (1/h, ...1/h)$, where h is the number of all neighborhood operators. Bring these operators into the ALNS algorithm to run them N times, and record the number of results that are better than the current optimal solution, which is set to n. Then the weights in the new round of iterations are adjusted according to the following equation.

$$w' = w(1\text{-}\lambda)+\lambda\frac{n}{N} \quad (0<\lambda<1), \tag{15}$$

Where λ is the adjustment coefficient. If we'd like to avoid excessive adjustment of the existing weights, which would reduce the diversity of the operators, a smaller value of λ should be adopted. If we prefer to gradually eliminate the poor operators and improve the efficiency of the search, we can take a larger λ value.

VI. Experiments and Results

In this section, a simulation experiment based on seven green parks and the road network in Changsha, a city in China, is first investigated, and used to illustrate the application of the truck-drone system and the proposed approach. Then, random instances are generated to further test the algorithms. All the computational experiments are conducted on a HUAWEI laptop, which uses Core i7 1.8GHz quad-core processor, 16GB of memory, Windows 10 operating system, and the algorithms are coded in Matlab R2018a.

A. Analysis of Practical Case

1) Case Description

In China, the government pay much attention to protect areas with green vegetations. In city of Changsha, there are seven main green parks as shown in Fig. 9, which are protected carefully. The planner would like to know the exact states of the vegetations in different periods. Thus, the GV-drone system is thought to be an efficient way to scanning all the parks and collect the information of the vegetations in them. The GV carries the drone starts from the depot and releases the drone to scan the area after arriving around the park. After the drone completes the coverage task, it returns to the GV, and then the GV departs for the next park. After all the parks are scanned, the GV returns to the depot. The objective is to find the optimal paths of the cooperated GV and drone to complete the scanning of all areas.

Fig. 10 is the abstraction and simplification of the main research content in Fig. 9, where the polygon surrounded by the red border represents the park to be scanned, and the star on the side of the park represents the potential stopping nodes for releasing and recycling the drone. The blue lines indicate the main roads and streets that connect the parks. The red square represents the depot.

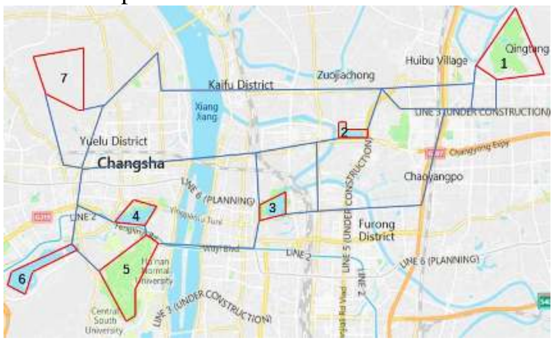


Fig. 9. Distribution of seven parks in Changsha

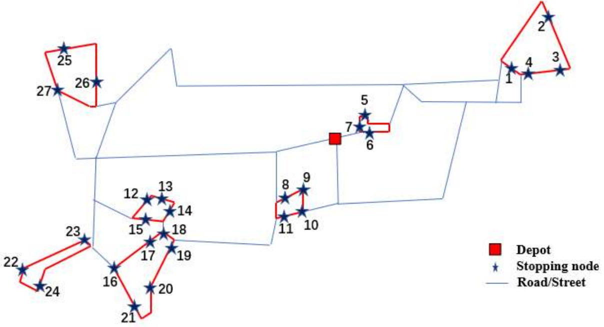


Fig. 10. Simplified diagram of the park and road network

In the experiments, the latitude and longitude coordinate information of all target points is obtained by open map systems, e.g. Baidu map. The employed drone is produced by DJI, PHANTOM 4 RTK, which is a small multi-rotor aerial survey drone with high-precision, and usually used for low-altitude photogrammetry. The average speed of the drone is 40km/h (Maximum speed is 50 km/h), the maximum turn rate of the drone is 0.7. Considering the height of the buildings and trees in the park, we set the flying height of the drone as 100m. Then the scanning width obtained under the accuracy requirements of GB/T 7930-2008 1: 500 topographic map aerial photogrammetry industry specification is 200m. Since these parks are located in the urban area, we set the average speed of the GV as 30km/h.

In the ALNS algorithm, N is set to 100, and λ is set to 0.2. After each round, the weight of each operator and the current optimal solution are updated. When there is no change in the current optimal solution for 3 consecutive rounds of operations, the iteration of the algorithm is stopped and the optimal solution is output.

First, the geometric characteristics of the seven areas are brought into formula (13) to calculate the roundness of different areas. In this case, the roundness of all the areas considered is less than 0.86, and thus the lawn mowing pattern is utilized. Areas 2, 5, and 6 are concave polygons, which are decomposed by the BCD method based on trapezoidal decomposition previously designed, and then a lawn mowing pattern was used to plan the Boustrophedon path for them.

TABLE 3
ROUNDNESS OF SEVEN PARKS

Park No.	Park name	Acreage (m^2)	Circumference (m)	Roundness
1	Songyahu	3.45e+06	8.36e+03	0.62
2	Yuehu	6.43e+05	4.37e+03	0.42
3	Lieshi	1.53e+06	5.29e+03	0.69
4	Xihu	1.37e+06	4.97e+03	0.70
5	Yuelushan	3.88e+06	8.63e+03	0.66
6	Meixihu	1.96e+06	8.50e+03	0.34
7	Gushan	4.36e+06	9.62e+03	0.59

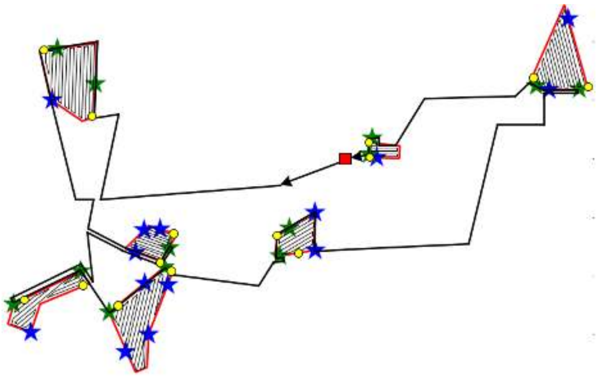


Fig. 11. Paths of the GV and drone in initial feasible solution

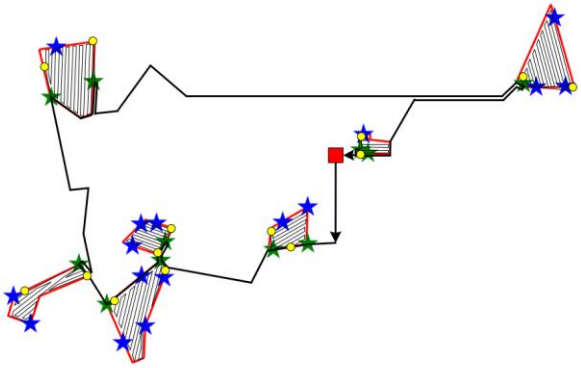


Fig. 12. Paths of the GV and drone in the solution by ALNS

2) Results and Analysis

The two-stage heuristic algorithm is used to construct an initial feasible solution. First, a set of stopping nodes are obtained, which are {3,1;5,7;11,8;14;16,18;23,22;26,25}. After the calculation we get the visiting sequence of the seven areas, which is 7-4-6-5-3-1-2. Then, the visiting order of the stopping nodes is 26-25-14-23-22-16-18-11-8-3-1-5-7, and the total time for covering all areas by the GV and the drone is 324.12 minutes.

Fig. 11 shows the stopping nodes used by the GV in different areas in the initial feasible solution. The green asterisks on the border of each area indicate the stopping nodes selected in that area. The start and end points of the scanning paths of each area are indicated by yellow circles.

Based on the initial feasible solution, the ALNS algorithm is employed to obtain better or optimal solution. In the final solution by ALNS, the visiting sequence of the seven areas is 3-4-5-6-7-1-2, and the visiting order of the stopping nodes is 10-11-14-18-16-23-27-26-1-6-7, which are illustrated in Fig. 12. The total time is 284.73 minutes, which is reduced by 12.15% compared to the time of the initial solution.

B. Experiments on random instances

In order to further verify the performance of the algorithms, random instances are designed and solved.

1) Problem Description

A 6×6 road network is generated on a plane, which divides the plane into 36 square grids. Set the side length of each square grid to 10 units. A certain number of grids are first randomly selected from the 36 grids. For each selected grid, a number of nodes are randomly generated in this grid, which are viewed as the vertices of the polygon. Then these vertices are connected in order, which forms the polygon in the grid. The lines dividing the plane into 36 grids are viewed as the main road network. For each generated polygon, three nodes are randomly selected in the boundary and connected to the main road to form the whole road network. Set the center of the plane as depot, where the GV carrying the drone starts to cover all areas and returns after completing the mission. The objective is to find the optimal routes of the cooperated GV and drone to complete the covering of all areas.

As shown in Fig. 13, eighteen grids are randomly selected and a random polygon is generated in each grid. The 18 polygons are numbered in order from left to right and bottom to top. A random stopping node is generated on each side of the polygon, which is indicated by a blue star. Finally, the road is connected with the polygon with line segments. The red square indicates the depot. The GV carrying the drone starts from the depot to cover the 18 areas, and returns to the depot after completing the mission. Find the optimal routes of the cooperated GV and drone to complete the covering of all areas.

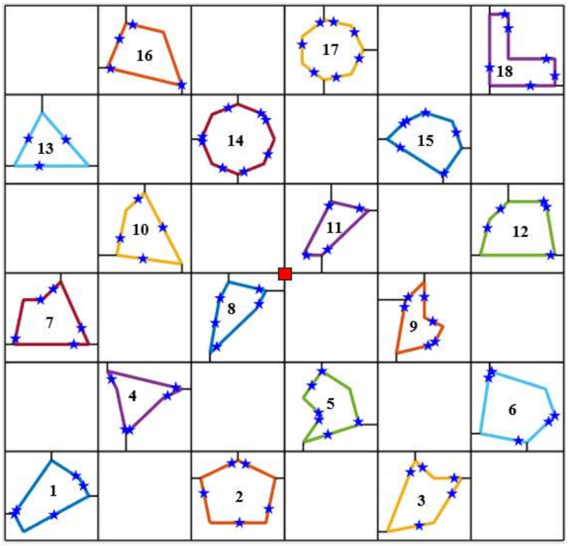


Fig. 13. Distribution of polygons and road network in random instance

TABLE 4
ROUNDNESS OF 18 AREAS

Area No.	Acreage	Circumference	Roundness
1	31.00	23.35	0.71
2	43.00	25.14	0.85
3	31.00	25.30	0.61
4	22.50	23.47	0.51
5	27.00	24.09	0.58
6	44.00	25.90	0.82
7	35.00	25.54	0.67
8	21.50	21.75	0.57
9	20.00	21.51	0.54
10	30.00	23.94	0.66
11	18.50	19.90	0.59
12	39.00	25.03	0.78
13	24.00	22.52	0.59
14	43.43	23.99	0.95
15	32.00	22.21	0.82
16	33.00	24.08	0.72
17	32.69	20.99	0.93
18	31.00	30.00	0.43

The average speed of GV is set to 0.05 units per second, and the flight speed of the drone is set to 0.1 units per second. Set the maximum turn rate of the drone be 0.7. Set the scanning width of the drone to 1 unit. N is set to 100, and λ is set to 0.2. The roundness of the 18 covered areas is calculated and presented in TABLE 4.

It can be seen from TABLE 4 that the roundness of the areas 14 and 17 is obviously greater than 0.86, so the spiral pattern is adopted, and the remaining areas adopt a lawn mowing pattern. Areas 3、4、5、7、9 and 18 are concave polygons. Decompose them using the BCD method based on trapezoidal decomposition previously designed, and then a lawn mowing pattern was used to plan the Boustrophedon path for them.

2) Results and Analysis

The initial feasible solution obtained by the Two-stage Heuristic algorithm is illustrated in Fig. 14, where the visiting sequence of the 18 areas is 11-9-5-2-3-6-1-4-12-18-13-16-17-15-7-10-14-8 and the order for visiting the stopping nodes is 54-52-43-45-23-26-7-6-12-13-30-28-5-2-17-60-59-93-90-61-81-79-86-85-77-75-32-34-51-48-69-68-38-40. The total time is 416.07 minutes.

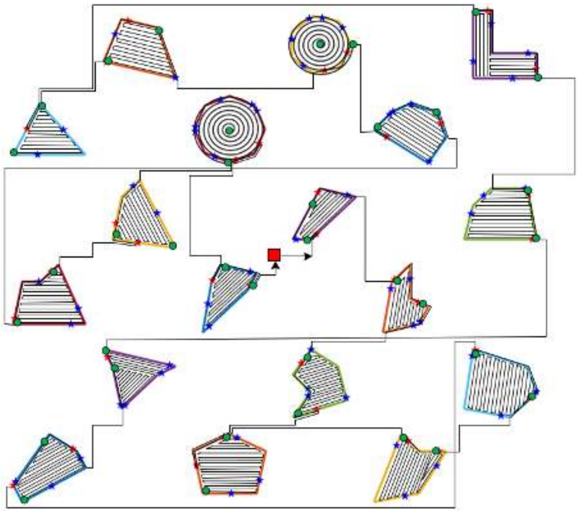


Fig. 14. Paths of the cooperated GV and drone in initial feasible solution

The initial feasible solution is further improved by the ALNS algorithm, and the final solution is illustrated in Fig. 15. The total time of the solution obtained the ALNS is 330.97 minutes, which is reduced by 19.25% compared to the initial solution.

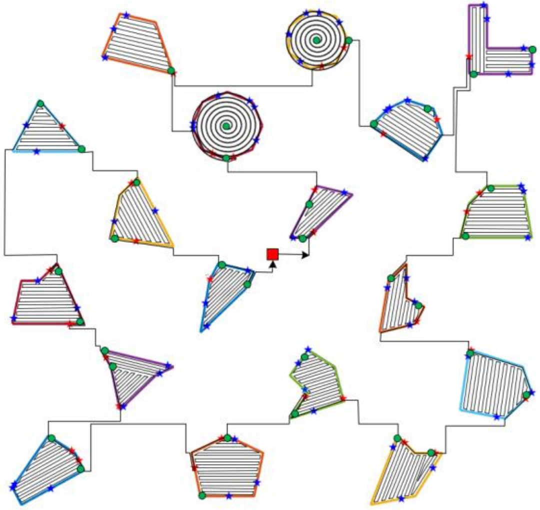


Fig. 15. Paths of the cooperated GV and drone in optimal solution

Based on the above process, ten such instances are randomly generated, and the number of areas for each instance is between 15-30. All the ten instances are solved by both the TSH algorithm and the ALNS algorithm, and the computational results are presented in TABLE 5.

It is the first time to employ the cooperated GV and drone for multi-area coverage, which generates a new variant of the routing problem. The algorithms presented are designed to solve this problem and there are no similar solution approaches in the literature. Thus, we only compared the initial solutions and final improved solutions to test the performance of the algorithm. In order to further test the performance of ALNS, an approximative path-planning algorithm, a multi-area coverage TSP path planning algorithm (MAC-TSP), designed and utilized to solve the ten instances.

The main process of MAC-TSP is as follows:

(a) Each covered area is abstracted into a point at the center of the polygon, and the routing problem of GV is transformed into a classic TSP problem, which is solved by the commercial software CPLEX 12.6.3.

(b) For each polygon, the flying and scanning path of the drone is optimized.

(c) At last, the route of GV and the flying paths of the drone are combined to form a feasible solution.

The MAC-TSP algorithm is easy to conduct, while it ignores the interaction between the GV and the drone. Thus, the solutions obtained by MAC-TSP are upper bounds of the ten instances.

All the 10 instances are solved by the MAC-TSP algorithm, and the solutions are also presented in TABLE 5. From the results in TABLE 5, it can be seen that the solutions obtained by ALNS is better than those of MAC-TSP over 12-13%.

TABLE 5
COMPUTATIONAL RESULTS OF 10 RANDOM INSTANCES

Instance No.	Number of areas	Scanning pattern S/L	TSH (min)	ALNS (min)	MAC-TSP (min)	Improvement (%)
1	15	1/14	334.67	275.40	313.71	12.21
2	17	2/15	372.13	311.19	354.80	12.29
3	18	2/16	410.13	332.28	380.55	12.68
4	20	3/17	447.55	357.52	414.71	13.79
5	21	4/17	466.60	371.54	431.97	13.99
6	23	5/18	497.67	398.79	463.16	13.90
7	24	6/18	514.88	413.76	480.33	13.86
8	26	7/19	554.82	451.77	518.48	12.87
9	28	8/20	591.37	484.44	556.40	12.93
10	30	10/20	638.30	512.67	584.21	12.24

VII. CONCLUSION

In this paper, for the first time, a study is conducted on the combination of the drone's area coverage problem and the cooperated GV and drone routing problem. we build a mixed integer programming model for the 2E3P-MAC-G&D problem. Then, a Two-stage Heuristic Based on Saving Strategies is proposed, which is used to obtain the initial feasible solution. And the ALNS algorithm was designed with destroy operators and repair operators to find better solution. Finally, the rationality and superiority of the algorithm are verified based on random instance and practical instance.

During the design of the drone's coverage path planning algorithm, a qualitative method for selecting the scanning pattern based on the roundness of the polygon is creatively proposed, where two scanning patterns are considered, and more patterns can be investigated in future research. This paper assumes that the drone's endurance can meet the coverage of each area, but this assumption should be relaxed when encountering large areas. At this time, the drone needs to fly back to the GV to recharge during the coverage process. It greatly increases the complexity of the problem, and the future work will be carried out on this aspect.

REFERENCES

[1] Warren, G., and Metternicht, G., "Agricultural applications of high-resolution digital multispectral imagery: Evaluating within-field spatial variability of canola (Brassica napus) in Western Australia," *Photogrammetric Engineering and Remote Sensing*, vol. 71, no. 5, pp. 595–602, May. 2015.

[2] Moran, M. S., Inoue, Y., and Barnes, E. M., "Opportunities and limitation for image-based remote sensing in precision crop Management," *Remote Sensing of Environment*,vol. 61, no. 3, pp. 319–346, Dec. Sept. 1997.

[3] Stafford, J. V, "Implementing precision agriculture in the 21st century," *Journal of Agricultural Engineering Research*, vol. 76, no. 3, pp. 267–275, July. 2000.

[4] Sanders R, "An israeli military innovation: Uavs," *INDUSTRIAL COLL OF THE ARMED FORCES WASHINGTON DC*, Mar. 2003.

[5] Zhang C and Kovacs J M, "The application of small unmanned aerial systems for precision agriculture: a review," *Precision agriculture*, vol. 13, no. 6, pp. 693–712, July. 2012.

[6] Luo Z, Liu Z and Shi J, "The Mathematical Modeling of the Two-Echelon Ground Vehicle and Its Mounted Unmanned Aerial Vehicle Cooperated Routing Problem," *2018 IEEE Intelligent Vehicles Symposium(IV)*, Changshu, Suzhou, China, pp. 1163–1170, June. 2018. DOI: 10.1109/IVS.2018.8500391.

[7] "DJI Kicks Off 2016 DJI Developer Challenge. DJI," Jan. 2016. [Online]. Available: http://www.dji.com/newsroom/news/dji-kicks-off-2016-dji-developer-challenge.

[8] Nedjati A, Izbirak G and Vizvari B, "Complete coverage path planning for a multi-UAV response system in post-earthquake assessment," *Robotics*, vol. 5, no. 4, pp. 26, Dec. 2016.

[9] Oksanen T and Visala A, “Coverage path planning algorithms for agricultural field machines,” *Journal of field robotics*, vol. 26, no. 8, pp. 651–668, Aug. 2009.

[10] Pinotti, Cristina M., et al. "Localization with guaranteed bound on the position error using a drone." *Proceedings of the 14th ACM International Symposium on Mobility Management and Wireless Access*. Nov, 2016, pp. 147-154 DOI: 10.1145/2989250.2998178.

[11] R. N. De Carvalho, H. A. Vidal, P. Vieira and M. I. Ribeiro, "Complete coverage path planning and guidance for cleaning robots," *ISIE '97 Proceeding of the IEEE International Symposium on Industrial Electronics*, Guimaraes, Portugal, 1997, pp. 677-682. DOI: 10.1109/ISIE.1997.649051.

[12] Çakıcı F, Ergezer H and Irmak U, “Coordinated guidance for multiple UAVs,” *Transactions of the Institute of Measurement and Control*, vol. 38, no. 5, pp. 593–601, May. 2015.

[13] Le Thi H A, Nguyen D M and Dinh T P, “Globally solving a nonlinear UAV task assignment problem by stochastic and deterministic optimization approaches,” *Optimization Letters*, vol. 6, no. 2, pp. 315–329, Feb. 2012.

[14] Pugliese L D P, Guerriero F and Zorbas D, “Modelling the mobile target covering problem using flying drones,” *Optimization Letters*, vol. 10, no. 5, pp. 1021-1052, July. 2016.

[15] Zorbas D, Pugliese L D P and Razafindralambo, “Optimal drone placement and cost-efficient target coverage,” *Journal of Network and Computer Applications*, vol. 75, no. 3, pp. 16–31, June. 2016.

[16] Otto A, Agatz N and Campbell J, “Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones,” *Networks*, vol. 72, no. 4, pp. 411-458, Mar. 2018.

[17] He, ZeFang, and Long Zhao. The comparison of four UAV path planning algorithms based on geometry search algorithm. *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. Aug. 2017, vol.2, pp. 33-36. DOI: 10.1109/IHMSC.2017.123.

[18] Guan Y, Gao M and Bai Y. Double-ant Colony Based UAV Path Planning Algorithm. *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*. Zhuhai, China, Feb, 2019, pp. 258-262 DOI: 10.1145/3318299.3318376.

[19] Avellar G S C, Pereira G A S and Pimenta L C A, “Multi-UAV routing for area coverage and remote sensing with minimum time,” *Sensors*, vol. 15, no. 11, pp. 27783-27803, Nov. 2015.

[20] Coombes M, Fletcher T and Chen W H, “Optimal polygon decomposition for UAV survey coverage path planning in wind,” *Sensors*, vol. 18, no. 7, pp. 2123-2159, July. 2018.

[21] Pěnička R, Faigl J and Váňa P, “Dubins orienteering problem,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1210-1217, Feb. 2017.

[22] Torres M, Pelta D A and Verdegay J L, “Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction,” *Expert Systems with Applications*, vol. 55, no. 15, pp. 441-451, Aug. 2016.

[23] Xu A, Viriyasuthee C and Rekleitis I, “Optimal complete terrain coverage using an unmanned aerial vehicle,” *2011 IEEE International Conference on Robotics and Automation,* Shanghai, May. 2011, pp. 2513-2519. DOI: 10.1109/ICRA.2011.5979707.

[24] Manyam S G, Casbeer D W and Sundar K, “Path planning for cooperative routing of air-ground vehicles,” *2016 American Control Conference (ACC)*, Boston, MA, July. 2016, pp. 4630-4635. DOI: 10.1109/ACC.2016.7526082.

[25] Agatz N, Bouman P and Schmidt M, “Optimization approaches for the traveling salesman problem with drone,” *Transportation Science*, vol. 52, no. 4, pp. 965-981, Apr. 2018.

[26] Deng C, Wang S and Huang Z, “Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications,” *Communication*, vol. 9, no. 9, pp. 687-692, Sept. 2014.

[27] Tokekar P, Vander Hook J and Mulla D, “Sensor planning for a symbiotic UAV and UGV system for precision agriculture,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498-1511, Dec. 2016.

[28] Ropero F, Muñoz P and R-Moreno M D, “TERRA: A path planning algorithm for cooperative UGV–UAV exploration,” *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 260-272, Feb. 2019.

[29] Phan C and Liu H H T, “A cooperative UAV/UGV platform for wildfire detection and fighting,” *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing,* Beijing, 2008, pp. 494-498. DOI: 10.1109/ASC-ICSC.2008.4675411.

[30] Huang W H. V, “Optimal line-sweep-based decompositions for coverage algorithms,” *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, Seoul, South Korea, May. 2001, vol.1, pp. 27-32. DOI:10.1109/ROBOT.2001.932525.

[31] Moon S W and Shim D H C, “Study on path planning algorithms for unmanned agricultural helicopters in complex environment,” *International Journal of Aeronautical and Space Sciences*, vol. 10, no. 2, pp. 1–11, Nov. 2009.

[32] Dubins, L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*. vol. 79, no. 3, pp. 497–516, July, 1957.

[33] Maza I and Ollero A, “Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms,” *Distributed Autonomous Robotic Systems 6. Springer,* Tokyo, Jan. 2007*,* pp. 221-230. DOI: 10.1007/978-4-431-35873-2_22.

[34] Xu A, Viriyasuthee C and Rekleitis I, “Optimal complete terrain coverage using an unmanned aerial vehicle,” *2011 IEEE International Conference on Robotics and Automation,* Shanghai, May. 2011, pp. 2513-2519. DOI: 10.1109/ICRA.2011.5979707.

[35] Mehlhorn K, Shermer T C and Yap C K. A complete roundness classification procedure. *Proceedings of the thirteenth annual symposium on Computational geometry*. vol. 21, no. 2, Dec, 1997.

[36] Smid M and Janardan R. On the width and roundness of a set of points in the plane. *International Journal of Computational Geometry & Applications*, vol. 9, no. 1, pp. 97–108, Aug. 1999.

[37] Bai B Q. “Traffic ban sign extraction based on color and circularity,” *Electronic Design Engineering*, vol. 19, no. 6, pp. 150–153, Mar. 2011.

[38] Seidel R, “A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons,” *Computational Geometry*, vol. 1, no. 1, pp. 51–64, Jan. 1991.

[39] Lee D T and Schachter B J, “Two algorithms for constructing a Delaunay triangulation,” *International Journal of Computer & Information Sciences*, vol. 9, no. 3, pp. 219–242, Feb. 1980.

[40] Lien J M and Amato N M, “Approximate convex decomposition of polygons,” *Computational Geometry*, vol. 35, no.1-2, pp. 100–123, Aug. 2006.

[41] Choset H, “Coverage of known spaces: The boustrophedon cellular decomposition,” *Autonomous Robots*, vol. 9, no. 3, pp. 247-253, Dec. 2000.

[42] Li Y, Chen H and Er M J, “Coverage path planning for UAVs based on enhanced exact cellular decomposition method,” *Mechatronics*, vol. 21, no. 5, pp. 876-885, Aug. 2011.

[43] Akl S G and Toussaint G T, “A fast convex hull algorithm. Information Processing Letters,” *Journal of Agricultural Engineering Research*, vol. 7, no. 5, pp. 219–222, Aug. 1978.

[44] Emeç U, Çatay B and Bozkaya B, “An adaptive large neighborhood search for an e-grocery delivery routing problem,” *Computers & Operations Research*, vol. 69, pp. 109–125, May. 2016.

[45] Ghilas V, Demir E and Van Woensel T, “An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines,” *Computers & Operations Research*, vol. 72, pp. 12–30, Aug. 2016.

[46] Grimault A, Bostel N and Lehuédé F, “An adaptive large neighborhood search for the full truckload pickup and delivery problem with resource synchronization,” *Computers & Operations Research*, vol. 88, pp. 1–14, Dec. 2017.

[47] V Hemmelmayr V C, Cordeau J F and Crainic T G, “An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics,” *Computers & operations research*, vol. 39, no. 12, pp. 3215-3228, Dec. 2012.

Yangsheng Xia received the B.S. degree in management science and engineering in 2018 from National University of Defense Technology, Changsha, P. R. China. He is currently working toward the M.S. degree with National University of Defense Technology. His research interests include vehicle routing, and heuristic design.

Chao Chen received the B.E. degree in industrial engineering from Beihang University, Beijing, China, in 2001, and the Ph.D. degree in management science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2007. Since 2008, he has been with the Science and Technology on Information Systems Engineering Laboratory, School of Systems Engineering, NUDT, as an Associate Professor, where he is currently a Professor with the College of Systems Engineering. He has authored over 60 journal and conference papers and has written four books. His main research interests include multi-objective optimization, target recognition, and machine learning.

Jianmai Shi received the Ph.D. degree in management science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2011. During 2008 to 2009, he was a Visiting Ph.D. Student with the University of Windsor, Canada. During 2014 to 2015, he was a Visiting Scholar at McGill University. He is now an Associate Professor at the School of System Engineering, NUDT, Changsha 410075, China. His current research interests include vehicle routing, supply chain management and heuristic design. He has authored more than 30 refereed papers including those published in IEEE TITS, COR and Applied Energy.

Yao Liu received the B.S. degree in management science and engineering in 2017 from National University of Defense Technology, Changsha, P. R. China. She is currently working toward the PhD. degree with National University of Defense Technology. Her research interests include vehicle routing, and heuristic design.

Guohui Li received PhD degree in information system engineering, National University of Defense Technology (NUDT), March 2001. He is the member of a council in Hunan Association for Artificial Intelligence. He has become full professor in NUDT since 1999. He had studied and worked in Communication Lab, Department of Electrical and Computer Engineering, University of Delaware, and Multimedia Communication and Visualization Lab, Department of Computer Science, University of Missouri-Columbia, USA as a research scholar during 1999 to 2000. His research interests include intelligent systems, information retrieval, machine learning and intelligent digital media.