IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 17, 2019, accepted September 29, 2019, date of publication October 9, 2019, date of current version October 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946448

A Self-Heuristic Ant-Based Method for Path Planning of Unmanned Aerial Vehicle in Complex 3-D Space With Dense U-Type Obstacles

CHAO ZHANG[1], CHENXI HU[1], JIANRUI FENG[1], ZHENBAO LIU[1], (Senior Member, IEEE), YONG ZHOU[1], AND ZEXU ZHANG[2]

[1]School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China
[2]Nanjing Research Institute of Electronics Technology, Nanjing 210039, China

Corresponding authors: Chao Zhang (caec_zc@nwpu.edu.cn) and Zhenbao Liu (liuzhenbao@nwpu.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61104030 and Grant 61672430, in part by the Fundamental Research Funds for the Central Universities under Grant 3102019ZX031, in part by the Shaanxi Key Research and Development Program under Grant 2019ZDLGY14-02-01, and in part by the Research Funds for Interdisciplinary Subject of NPU under Grant 19SH030401.

ABSTRACT Optimal path planning is required in autonomous navigation and intelligent control of the unmanned aerial vehicle (UAV). However, as a kind of common obstacles in complex three-dimensional (3-D) spaces, U-type obstacles may cause UAV to be confused and even lead to a collision or out of control. Although most of the Ant Colony Optimization (ACO) algorithm can generate proper path, solutions to U-type obstacles based on the specific behaviors of each ant are investigated rarely. Hence, different search strategies are studied and a novel ACO-based method called Self-Heuristic Ant (SHA) is proposed in this paper. The whole space is constructed by grid workspace model firstly, and then a new optimal function for UAV path planning is built. To avoid ACO deadlock state (i.e., ants are trapped in U-type obstacles when there is no optional successor node), two different search strategies are designed for choosing the next path node. In addition, the SHA is utilized to improve the ability of the basic ACO-based method. Specifically, besides pheromone update, a new information communion mechanism is fused to deal with the special areas which contain dense obstacles or many concave blocks. Finally, several experiments are investigated deeply. The results show that the deadlock state can be reduced effectively by the designed two different search strategies of ants. More importantly, compared with the conventional fallback strategy, the average number of retreats and the average running time of ACO can be reduced when SHA is applied.

INDEX TERMS Ant colony optimization, path planning, self-heuristic ant.

I. INTRODUCTION

As a kind of advanced autonomous robot, unmanned aerial vehicles (UAVs) have been widely utilized in science, transportation, energy, agriculture, entertainment, etc. for great advantages of superior maneuverability, low cost, easy to use and no casualties [1]–[5]. According to incomplete statistics, the sales scale of civil UAV products were about ¥1.5, ¥2.33, ¥3.7 billion in 2014, 2015, 2016 in China and is expected to reach ¥97.69 billion by 2023, the year-on-year growth rates are as high as 55%, 58%, 59%, respectively. With the urgent development of UAV towards autonomous flight and intelligent control, optimal global path planning as one of the key technologies for UAV are becoming increasingly essential.

Global path planning is to create an ordered sequence of intermediate way points under the known obstructed environment based on certain given evaluation criterion, as well as segments linking each pair of adjacent way points, that the vehicles can visit each way point along these segments to generate an optimal or secondary optimal obstacle-avoidance path from origin to destination [6].

Various conventional approaches have been developed to solve this problem, such as visibility graphs [7], potential field [8], heuristic approaches [9], etc. In visibility graphs approaches, a set of lines is defined to connect an object's

The associate editor coordinating the review of this manuscript and approving it for publication was Halil Ersin Soken.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

features to those of another. But for n features, n^2 links will be created and its complexity is $O(n^2)$. In potential field approaches, the whole space is always processed as vector fields firstly and artificial potential functions are used to describe it, then UAV is considered as a point in the space that integrates tension to the destination and expulsion from obstacles. Under the affection of vector fields, UAV would head toward the destination and avoid a collision which is convenient for the underlying real-time control. However, UAV is easily trapping to local optima when concave obstacles are contained in workspace. Furthermore, the vector fields may cancel each other out which will cause these approaches to fail easily. In heuristic approaches, the occupancy grid is utilized to divide the whole space into equally separated cells and thus the path planning problem becomes a graph search problem, so the A* algorithm [10], [11] is widely used. However, the path generated may be too close to obstacles which could lead to a collision between vehicles and obstacles. Besides, the time complexity of A* algorithm cannot be ignored when the number of grids in workspace is large.

In the last decade, inspired by natural phenomena, many bio-inspired intelligent algorithms, e.g., Ant Colony Optimization (ACO) [12]–[16], Genetic Algorithm (GA) [17], [18], Artificial Immune System (AIS) [19], [20], Cuckoo Search [21], Bacteria Foraging Optimization [22], Particle Swarm Optimization (PSO) [23]–[27] are attempted at path planning. It should be noted that each of these algorithms has advantages and disadvantages. As a kind of evolutionary algorithm, GA has strong robustness and global search capabilities. By simulating the behaviors between chromosomes, the core of the algorithm is independent of the problem to be solved. However, the local search ability and the convergence speed are not high. Besides, it may be difficult to encode and decode the problem itself. Based on the relationship between antigen and antibody, AIS is able to adjust itself and produce solutions according to the inputs. Although the convergence speed is fast, AIS may reach a balanced state and antibody population cannot be improved further. Moreover, for the evaluation which based on the concentration of antibody, the concentration of good antibody should be increased but not too high to guarantee both the convergence and the population diversity. Thus, the algorithm may be easily influenced by this conflict. Based on the cluster behavior of birds, PSO owns the advantages of high efficiency and simple structure, but it cannot process the discrete optimization problem effectively and may trap into the local optimum. In addition to the above, a detailed review can be referred to [28].

Bio-inspired algorithms are various and each has both advantages and disadvantages in different aspects. Thus, it may be hard to determine which one is absolutely advantageous. With the strong robustness, preferable global optimization performance, good distributed computing, and self-organization characteristics, ACO which proposed by the Italian scholar M. Dorigo is investigated popularly as an important path planning solving method [29]–[31]. However, some problems exist in ACO cannot be ignored and many scholars have been solving them.

In a 2-D environment. To simulate real ant colonies, the conceptions of the neighboring area and smell area were presented, then a hybrid ant colony (HAC) algorithm was designed to avoid premature convergence by combining pheromone search and random search strategies [32]. According to actual ants, a scout ant cooperation (SAC) algorithm was presented by dividing ants into two groups that adopted nearest neighbor search strategy and random search strategy respectively [33]. To increase path searching efficiency, a bidirectional searching ACO was proposed [34]. To accelerate the decision process of selecting new path nodes, a simple ACO distance-memory (SACOdm) is proposed, in which the distance of the source and destination is added to state transition function [13].

In a 3-D environment. To simplify the planning task, each path node was selected from different layers generated along the longitude or latitude direction, further a state transition probability was given and differential evolution algorithm (DEA) was used to update pheromone [35]. To improve the performance, a visibility graph as well as a series of ACO pheromone updating rules was produced [36].

As ACO is easy to blend with other algorithms, much research has been done on the combination of ACO and other algorithms. Combined with the immune network, the stimulation and suppression between antigen and antibody were utilized to find initial paths before ACO was used to search the optimal path in the antibody network [37]. A hybrid metaheuristic ACO was proposed as well as DEA was utilized to update pheromone [38]. Cellular Ants (CA) and ant algorithm were combined to find a collision-free path without priori information of configuration area, namely, they could be used in a dynamic environment [39].

However, deadlock state and stagnation problem cannot be ignored, much research has been carried out to overcome these drawbacks from different aspects. An enhanced ACO algorithm was proposed, in which the search deadlock was ruled out by modifying initial environment pheromone and state transition probability [40]. The chaos disturbance factor was employed to avoid the stagnation phenomenon and a new evaluation criterion was introduced to avoid the deadlock state [41]. To avoid stagnation and get the best result, the amount of pheromone on each path was limited to a certain range, besides elitist strategy was adopted and the best solution in each iteration was recorded [42]. To improve the ability of global searching, the parameter which determines the importance of exploitation and exploration was adjusted continuously, further two strategies of depositing pheromone were presented to prevent ants from dabbling into complex traps deeply and to avoid the premature phenomenon [43].

The ACO-based algorithms above have been improved in many ways, however in the following aspects, the considerations in this paper are different or more comprehensive.

1) Research in the 3-D space with dense U-type traps is rare because of the variety of obstacle shapes. For U-type

obstacles only, pits, corners, concave walls, etc. are included. When the type and quantity of obstacles are numerous, path-finding is difficult. In this paper, the path-finding behaviors are implemented in different complex environments with various U-type obstacles.

2) The deadlock state is easily generated in the map with U-type obstacles. However, the research about the deadlock state in 3-D environments is rare. When ants select new path nodes, this problem is caused by the interaction of complex obstacles and search strategies. In this paper, deadlock state between the complex obstacles were analyzed, then different search strategies were tried to solve the deadlock state.

3) To improve the efficiency of path-searching among the complex obstacles, a novel idea is proposed based on the fallback strategy. Specifically, ants are prevented from entering the U-type traps by modifying the specific behaviors of each ant. Based on above, inspired by the fallback strategy of ACO, a new information exchange mechanism was established to enhance the ability of ants identifying U-type traps. As a result, these traps will not be explored by ants deeply.

In this paper, 3-D environments with U-type obstacles are constructed by the grid workspace model. In the research of deadlock state, a taboo list which records the cells have been visited is modified as the result of changing search strategies when selecting new path nodes. Furthermore, the paths produced in this way contain repeated path segments and two methods are adopted to cut off redundant path segments. To improve the ability of ACO-based method, a new approach called Self-Heuristic Ant (SHA) is proposed. In this approach, the fallback strategy which allows ants to move backward is integrated with a new idea, then a novel information communion mechanism between each ant is established, as a result, the times of moving backward is reduced and the original algorithm is improved. The effectiveness of the proposed approach is demonstrated via simulation.

The remainder of this paper is organized as follows: Section 2 reviews the basic ant algorithm theory. Section 3 elaborates on the ACO algorithm in 3-D environments. Section 4 introduces the research about solving deadlock state and the methods adopted to cut off redundant path segments. Our proposed approach, SHA, is discussed in Section 5. Section 6 contains multiple simulation results and the analyses. Finally, Section 7 concludes the paper and offers the future direction of this research.

II. BASIC ANT ALGORITHM THEORY

Inspired by the foraging behavior of real ant colony, M. Dorigo *et al.* proposed the basic model of the ant colony algorithm. Specifically, ants deposit pheromone along the paths they traversed between the nest and food source [44]. To reach the food source, they exchange information and cooperate with each other through perceiving the pheromone trails [45]. It has been proved both mathematically and experimentally that the ants can find the shortest path between the nest and the food source. The mathematical proof is provided in [46] and the foraging behavior of real ant is simulated in [47]. According to double bridge experiment which depicted in Fig. 1, the experimental proof is given, more details are provided in [39].

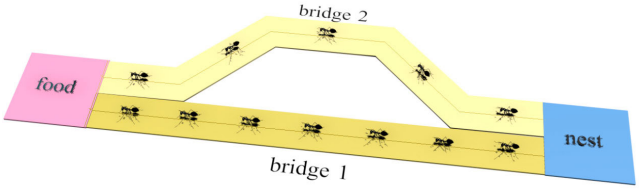


FIGURE 1. Double bridge experiment.

TABLE 1. Variables of Ant-based algorithm.

Variable	Description
α	Importance coefficient of pheromone concentration
β	Importance coefficient of heuristic information
τ_{ij}	Pheromone concentration accumulated on link (i, j)
η_{ij}	Heuristic information, i.e., cost of link (i, j)
$taboo_k$	Visited nodes table by ant k
ρ	Pheromone evaporation coefficient
m	Amount of ants
Q	Constant value related to the speed of convergence
L_k	Path length
d_{ij}	Distance between node i and node j
N_c	Number of iterations
N_c_max	Maximum number of iterations

Several main steps contained in ant algorithms are briefly introduced as follows.

1) Environment modeling and parameter initialization. The workspace should be modeled as a graph with N nodes and L links. Existing paths and costs between nodes are represented by weighted directed graph. The number of ants (N_{ants}) is defined and each ant is put on the origin. The values of N_{ants}, as well as more basic parameters of ant algorithms, can be assigned via experimental experience or trial and error approaches.

2) State transition. The probability selection formula used by ant k to select the next node is established as Eq. (1), which is used to calculate the state transition probability of j as next node from i. All variables mentioned can be found in Table 1.

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\partial}\eta_{ij}^{\beta}}{\sum\limits_{s\notin taboo_k}\tau_{is}^{\partial}\eta_{is}^{\beta}} & i,s,j \notin taboo_k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Equation 1 has a great influence on the behavior of ants. If α is larger than β, ants are more likely to search along paths according to pheromone left by previous ants. Otherwise, ants tend to select paths according to heuristic information which is similar to the greedy search algorithm.

3) Pheromone update. This process contains pheromone reinforcement and pheromone evaporation. The update process is represented by Eq. (2) where all variables mentioned

can be found in Table 1.

$$\tau_{ij}^{now} = (1-\rho)\tau_{ij}^{past} + \rho\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{2}$$

Based on the Ant-Cycle model, the amount of pheromone deposited by ant k on the path between i node and j node is computed via Eq. (3), all variables mentioned can be found in Table 1.

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} & \text{if the } k^{th} \text{ ant walks along link(i, j)} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

III. UAV PATH PLANNING BASED ON ACO

A. ENVIRONMENTAL MODELING

The actual flight area is a real physical space whereas the workspace applying ACO is an abstract space of the real environment. In this paper, the grid workspace model is used to construct the actual flight area, the grid granularity which has a great influence on the precision of environmental modeling is taken as $1\times1\times1$. More details are discussed in the following.

1) Storage of terrain information. The 2-D matrix is used to record the height of terrain. To explain this, a matrix which contains 4 rows and 4 columns is formed as Eq. (4).

$$\text{Terrain} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

where $Terrain(i, j) = 2$ indicates that there is an obstacle at the position of the i-th row and the j-th column, furthermore, the obstacle height is 2.

2) Division of workspace and the corresponding graph. In the Cartesian coordinate system, take X_{max}, Y_{max}, Z_{max} as the maximum values in the positive direction of the x axis, the y axis, and the z axis, respectively. If the horizontal step size is δ_1 and the vertical step size is δ_2, then the configuration area of size $X_{max} \times Y_{max} \times Z_{max}$ can be divided into $N_x \times N_y \times N_z$ cells, where N_x, N_y, N_z is the round of X_{max}/δ_1, Y_{max}/δ_1, Z_{max}/δ_2 respectively. The cell occupied by obstacles is obstacle grid, otherwise it is a free grid. The coordinates of each cell is represented by its center point. An example is depicted in Fig. 2 (a), where $N_x = N_y = 4$ and $N_z = 2$, the colored cells are obstacle grids and colorless cells are free grids. The 3-D structure of the graph corresponding to Fig. 2 (a) is shown in Fig. 2 (b). There are bidirectional paths between the adjacent nodes in free cells, and each blue node is a potential path node.

3) Workspace processed by ACO. In this paper, three maps of size $21\times21\times10$ were used in simulation, they are depicted in the simulation section.

B. MOTION CONSTRAINTS OF THE VEHICLE

To make the path obtained by ACO become flyable, the motion constraints of aircraft should be considered. Because the workspace is represented by the grid map, the motion constraints are simplified to easily combine with the environmental modeling method and the ACO itself. Here are some main constraints.

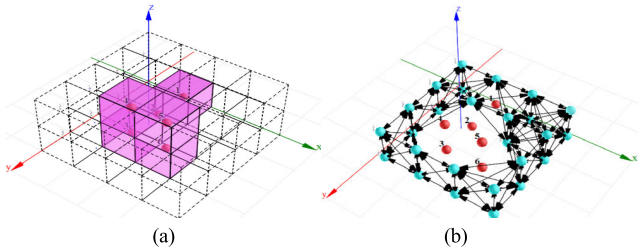


FIGURE 2. Grid workspace and the corresponding graph.

1) The minimum step size. Before the aircraft makes a manoeuvre, a distance needs to be maintained. Take this value as l_{min}, then the length between node i and node j cannot be smaller than l_{min}, which is shown in Eq. (5).

$$d_{ij} \geq l_{min} \tag{5}$$

For convenience, the distance between the neighboring grids is longer than l_{min} in our research.

2) Maximal slope of UAV. Slope is the included angle formed by the horizontal and the flying direction, which indicates the change of flying direction in the vertical direction. Limited by the maneuverability, slop s_i at node (x_i, y_i, z_i) should be less than the maximal slop and it can be represented by Eq. (6).

$$s_i = \frac{z_i - z_{i-1}}{\left\|(x_i - x_{i-1}, y_i - y_{i-1})\right\|} \tag{6}$$

where $||x||$ is the norm of vector x.

3) Maximal turning angle. In the horizontal direction, turning angle is the angle formed by the previous direction of UAV and its current direction. Limited by the maneuverability, the turning angle θ_i at node (x_i, y_i, z_i) cannot exceed the maximum. The formula is as following.

$$\theta_i = \arccos\left(\frac{(x_i - x_{i-1}, y_i - y_{i-1})\cdot(x_{i+1} - x_i, y_{i+1} - y_i)^T}{\left\|(x_i - x_{i-1}, y_i - y_{i-1})\cdot(x_{i+1} - x_i, y_{i+1} - y_i)\right\|}\right) \tag{7}$$

C. ACO BASED ON 3-D ENVIRONMENT

To generate a feasible path in 3-D environments, several improvements have been made to the basic ant algorithm and some of them are discussed as follows.

1) Selection of successive node. Based on Eq. (1), the state transition probability is calculated, then the roulette wheel selection procedure is used to select the next node.

2) The establishment of the objective function. To evaluate the paths obtained by ACO, a new objective function representing the comprehensive cost of the path is computed by Eq. (8).

$$\text{costs} = C_1 \bullet \text{cost\_length} + C_2 \bullet \text{cost\_altitude} + C_3 \bullet \text{cost\_height} \tag{8}$$

TABLE 2. Related variables.

Variable	Description
C_1	Importance coefficient of *cost_length*
C_2	Importance coefficient of *cost_altitude*
C_3	Importance coefficient of *cost_height*
H_i	The height of node *i* relative to the lowest point of workspace
h_i	The height of node *i* relative to the ground directly below it
d_{ij}	Euclidean distance between node *i* and node *j*
$x(i)$	*X*-coordinate of node *i*
$y(i)$	*Y*-coordinate of node *i*
$z(i)$	*Z*-coordinate of node *i*
L_1	The minimum height of UAV from the ground below it
L_2	The maximum height of UAV from the ground below it
d_{jF}	Euclidean distance between node *j* and destination
C_4	Importance coefficient of the distance factor
C_5	Importance coefficient of the height factor
η_j	Circumvention heuristic factor of cell *j*
$\tau_{ij}^{initial}$	Initial pheromone concentration on link (i, j)
ρ_1	Pheromone evaporation coefficient of local update
ρ_2	Pheromone evaporation coefficient of global update

where *cost_altitude* is the sum of the heights of all nodes in one path relative to the lowest point of workspace, *cost_height* is the sum of the heights of all nodes in one path relative to the ground directly below them, *cost_length* is the length of a path. These three factors are controlled by weight variables, C_1, C_2, and C_3, respectively. It should be noticed that *cost_altitude* and *cost_height* are different, because low flying altitude relative to the lowest point of workspace does not mean flying close to the ground and vice versa. When $C_2 > C_3$, its more important to fly at low altitude relative to the lowest point. When $C_2 < C_3$, its more important to fly as close as possible to the ground below UAV. If it is assumed that a path obtained contains *n* nodes, then the variables above can be calculated by Eq. (9), Eq. (10), and Eq. (11). The variables mentioned can be found in Table 2.

$$\text{cost\_altitude} = \sum_{i=2}^{n-1} H_i \tag{9}$$

$$\text{cos t\_height} = \sum_{i=2}^{n-1} h_i \tag{10}$$

$$\text{cos t\_length} = \sum_{i=1}^{n-1} d_{ij}, \quad j = i + 1 \tag{11}$$

The height between UAV and the terrain below it should always be higher than a safe distance predefined but not too large. Furthermore, flying outside the known workspace is not permitted. With the new objective function and the constraints above, the minimum value f of the *costs* is expressed via Eq. (12). Partial variables can be found in Table 2.

$$f = \min(\text{costs}) \begin{cases} 0 < x(i) \le X_{max} \\ 0 < y(i) \le Y_{max} \\ h_i + L_1 \le z(i) \le h_i + L_2 \\ 0 < h_i \le Z_{max} \end{cases} \tag{12}$$

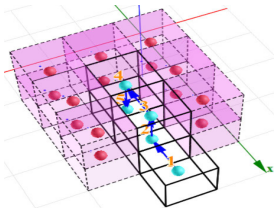


FIGURE 3. Ant trapped in an U-type obstacle.

3) Establishment of new heuristic information. For the distances between two adjacent grids in a 3-D environment, the difference between the maximum value (i.e., $\sqrt{3}$) and the minimum value (i.e., 1) is small. Thus, the effect of the heuristic information which usually represented as Eq. (13) is not obvious. In addition, flight height is not contained in Eq. (13). In response to these problems, the heuristic function is redefined as Eq. (14), where η_j is represented as Eq. (15). All variables can be found in Table 2.

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{13}$$

$$\eta_{ij} = \frac{C_4}{d_{ij} + d_{jF}} \bullet \frac{C_5}{h_j} \bullet \eta_j \tag{14}$$

$$\eta_j = \begin{cases} 0 & j \text{ is an obstacle cell} \\ 1 & j \text{ is an free cell} \end{cases} \tag{15}$$

4) Pheromone update strategy. Pheromone update is divided into two parts, namely, local update and global update. Local update defined as Eq. (16) which allows ants to explore new paths is executed every time when ants move to the next node. The variables can be found in Table 2.

$$\tau_{ij}^{now} = (1 - \rho_1)\tau_{ij}^{past} \tag{16}$$

The global update which ensures the convergence of ACO is only performed on the best path after all ants in a generation have arrived at the destination. The pheromone update function is as Eq. (17), as shown at the bottom of this page, where the variables mentioned can be found in Table 2.

5) Measures to prevent stagnation. In 3-D environments, ants are easy to fall into U-type traps. An example is shown in Fig. 3 where an ant is trapped in node 5. To avoid the deadlock state and the stagnation of algorithm, some special measures (e.g., ants move backward sequentially to get out of traps, fill the traps with obstacle grids, drop the lost ants)

$$\tau_{ij}^{now} = (1 - \rho_2) \bullet \tau_{ij}^{past} + \rho_2 \bullet \left[\frac{Q}{C_1 \bullet \text{cost\_length}_{best} + C_2 \bullet \text{cost\_altitude}_{best} + C_3 \bullet \text{cost\_height}_{best}} \right] \tag{17}$$

TABLE 3. Parameters and their ranges.

α	β	ρ_1	ρ_2	m	Q	N_c_max	$\tau_{ij}^{initial}$
1-10	1-10	0-1	0-1	1-50	1-500	1-200	1-100

have been proposed. More methods are discussed later in this paper.

6) Optimization of parameters using multiple population genetic algorithm (MPGA). The parameters involved in ACO which directly affect the performance of the algorithm are numerous and interrelated. But for the problem of the parameters selection, there is no general solution. Thus, eight parameters of ACO are optimized by GA because of its good operability and high efficiency. The range of these parameters is shown in Table 3.

The parameters above are coded as genes in the chromosomes of GA. During the execution of GA, ACO is called and the comprehensive costs of paths computed by Eq. (8) are utilized to evaluate the fitness. When GA is complete, the optimal chromosome which contains the best combination of 8 parameters can be obtained. The whole steps are shown in Fig. 4 (a).

However, under the influence of some features of GA, especially the influence of elite retention strategy, GA tends to fall into local optimum. To solve the problem, MPGA was adopted. Specifically, several independent populations with different control parameters were combined by the immigration operator. For each generation during the evolution, artificial selection operator was used to put the optimal solution into the elite population for preservation. Based on this population, the number of generations maintained by the optimal individual is taken as the basis for ending algorithm. Since the focus in the article is not MPGA, more relevant content will not be mentioned here. The structure of MPGA is as Fig. 4 (b).

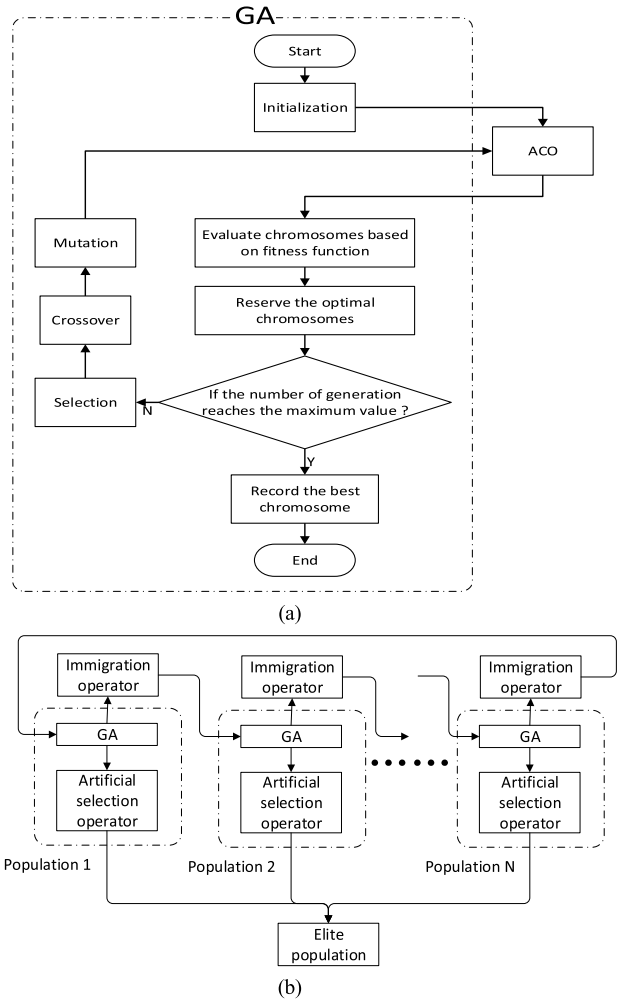


FIGURE 4. The structures of GA and MPGA.

IV. NEW IDEAS TO SOLVE DEADLOCK STATE

When encountering U-type obstacles, ants may be trapped when there is no optional successor cell. In basic ACO, the ant cannot revisit the same nodes, which is called Default Mode (DM) in this paper. In this case, when the neighboring cells of current cell are fully occupied (e.g., occupied by the obstacle cells, the cells have been visited, the cells does not satisfy the constraints), then there is no successor cell available and it is called deadlock state in this paper. To escape from the U-type obstacles and avoid deadlock state, dropping the ants trapped in traps is widely used. However, when lots of U-type obstacles are distributed in workspace, a large number of ants may be dropped as well as many incomplete paths, i.e., a lot of computing resources are wasted. To solve this, search strategies of ants are studied and two different methods of searching for new path nodes are compared which called Extension Mode 1 (EM1) and Extension Mode 2 (EM2). The paths obtained by these methods contain redundant path segments and this problem is solved by two methods which called Splicing method and Dijkstra method.

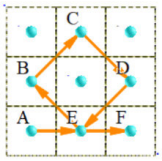


FIGURE 5. Path with redundant path segments.

A. EXTENSION MODE 1

In this approach, the taboo table only contains the previous node of the current node. Specifically, when ant k reaches a new node, the unique element in $taboo_k$ is a previous node. Except for the previous node, those nodes which have been visited can be revisited. Thus, the path nodes that ants can choose are more diverse and the problem of no successor cell is less likely to occur. By this means, however, the deadlock state cannot be completely avoided, and redundant path segments may be contained in the paths. To illustrate the formation of redundant path segments, a 2-D grid map is taken as an example.

TABLE 4. Variables in Splicing method.

Variable	Description
i	The ordinal of path node
path_1	The original path
list_1	The path that does not contain redundant segments
co_1	The *i*-th node in *path_1*
co_2	In *list_1*, the node which has the same coordinates as *co_1*

In Fig. 5, an ant visits nodes in the order of A, E, B, C, D, E, and F. After the ant reaches the point E for the first time, it passes through B, C, D and returns to point E again, generating redundant path segments.

B. EXTENSION MODE 2

In this case, all nodes which have been visited by ants can be revisited. By this means, the diversity of searching for new nodes is further increased and the deadlock state can be completely avoided. However, the redundant path segments which are contained in the paths are more complex.

C. METHODS TO CUT OFF REDUNDANT PATH SEGMENTS

The paths which contain redundant path segments produced by EM1 and EM2 are not feasible. To cut off the redundant path segments, two methods are applied.

1) Splicing method. This method is capable of deleting redundant path segments between two same nodes and connecting the remaining parts at the same nodes. The steps are as follows and all variables can be found in Table 4.

Step 1. The parameter i is initialized to 1. The original path which contains the redundant path segments is stored as *path_1* according to the order of each path node.

Step 2. The i-th node in *path_1* is stored in *list_1*.

Step 3. $i \leftarrow i + 1$. The i-th node in *path_1* is recorded as *co_1*. If *co_1* is the destination, *co_1* is stored in *list_1* and go to Step 6. Otherwise, move on to Step 4.

Step 4. If there is a node in *list_1* with the same coordinates as *co_1*, move on to Step 5. Otherwise, go back to Step 2.

Step 5. In *list_1*, the node which has the same coordinates as *co_1* is recorded as *co_2*. Delete *co_2* and all nodes after it in *list_1*. Go back to Step 2.

Step 6. Output the path in *list_1* as the result which does not contain the redundant segments.

2) Dijkstra method. The Dijkstra algorithm is a typical algorithm for solving the shortest path problem. By applying it, the establishment of the shortest path from the origin to the destination as well as the removal of the redundant segments can be achieved.

To evaluate the effects of the Splicing method and the Dijkstra method conveniently, a 2-D example is taken as Fig. 6 (a), where an ant visits each node in the order of A, D, F, D, B, C, E, F, and G. The effect of applying Splicing method and Dijkstra method are shown in Fig. 6 (b) and Fig. 6 (c), respectively.

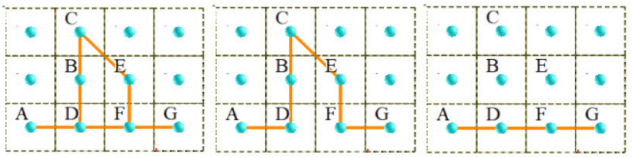


FIGURE 6. Evaluation of Splicing method and Dijkstra method.

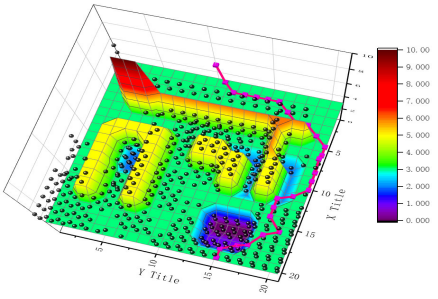


FIGURE 7. The optimal path and all taboo nodes obtained by a group of ants.

In Fig. 6 (a), the ant needs to reach the point F via B, C, and E before arriving at the point G. Therefore, Fig. 6(b) can reflect the true order of the nodes visited by the ant though the path obtained is not the shortest one. Conversely, the shortest path can be obtained by applying the Dijkstra algorithm although the result cannot reflect the true order of nodes visited by the ant, which is depicted in Fig. 6 (c).

V. SELF-HEURISTIC ANT

EM1 and EM2 can solve the problem of deadlock state. However, as the diversity of searching for new nodes increases, the efficiency of the algorithm is not high enough. Although ants can escape from traps under the influence of pheromone update and roulette wheel selection, the same nodes in U-type traps may be visited repeatedly, which consumes a lot of computing resources.

Reconsider the convex processing strategy and the fall-back strategy in ACO which is widely used, the former one processes the workspace before the algorithm starts, which fills the traps with obstacle grids and turns the concave traps into the convex shapes. However, this process is complicated in 3-D environments where many obstacles of different shapes are contained. The latter one makes the ants move backward sequentially to get out of the U-type traps. However, the process of judging whether an ant needs to move backward is complicated and time consuming. In this means, the nodes which ant k retreats from are called taboo nodes and are added to list $taboo_k$ to prevent being revisited. Once the ant k arrives at the destination, the $taboo_k$ should be emptied, the reason is illustrated as follows.

In the 3-D workspace shown in Fig. 7, the pink line is the obtained path and taboo nodes are represented by black dots. The optimal path here is not the final path, and the motion constraints of UAV have not been considered yet. The taboo

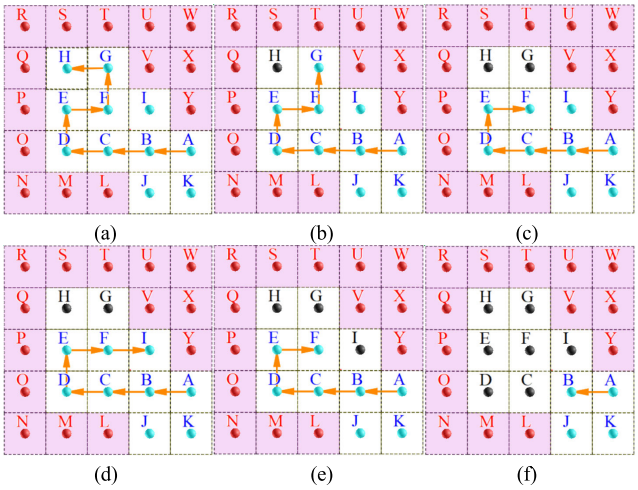


FIGURE 8. The process of SHA in 2-D workspace.

nodes generated by the fallback strategy also exist outside the U-type traps, thus, these taboo nodes cannot be used to prevent ants from entering U-type traps and $taboo_k$ should be emptied every time when the ant k arrives at the destination, i.e., these taboo nodes are not being fully utilized. Another problem with taboo nodes in the fallback strategy is that when the ant travels in the narrow area, the taboo nodes may block the area containing the optimal path, which could cause the ants to fail to reach the goal. In response to the above, the SHA is proposed.

A. THE PRINCIPLES OF BASIC SHA

In SHA, taboo nodes generated by the fallback strategy are divided into two categories which are defined as follows.

1) Valid taboo nodes. The taboo nodes inside the U-type traps are valid taboo nodes which can be used to prevent all ants from entering the traps.
2) Invalid taboo nodes. The taboo points outside the U-type traps are invalid taboo nodes which cannot be used to prevent all ants from entering the traps.

Because of the characteristics above, the invalid taboo nodes are not expected to be generated. To gain a set of valid taboo nodes, it is necessary to exclude the invalid taboo nodes when generating taboo nodes. The set of taboo nodes generated in this way can be used to prevent all ants from entering the same U-type traps. By studying the reasons for the formation of valid taboo nodes, the following conditions which are indispensable to generate a valid taboo node k are proposed.

(1) Condition one. k is a taboo node which the ant moves back from.

(2) Condition two. In the horizontal plane where the taboo node k is located, other taboo nodes or the obstacle nodes existing in the eight neighboring grids around k can form corners in the shape of the letter "L."

For the convenience of understanding, an example is taken as Fig. 8, where the red dots in pink squares are obstacle nodes whereas the blue dots in white squares are free nodes. In addition, the taboo nodes are represented by black dots, the trajectory and the direction of the path crawled by ant are represented by a series of orange vectors. These representations are also applied to other examples which are mentioned later.

First of all, an ant reaches node H from node A in the order of B, C, D, E, F, and G, as shown in Fig. 8 (a). When H is reached, there is no successor node to select and the ant needs to move back from H to G. Among the eight nodes around node H, the obstacle nodes Q, R, S may form a corner in the shape of the letter "L." According to the two conditions proposed above, H is a valid taboo node. The result of which the ant moves back to node G is shown in Fig. 8 (b). The steps above will be repeated before the node F is reached, as shown in Fig. 8 (c). Now the node I is the only node which can be selected. Thus, the ant moves to node I, as shown in Fig. 8(d). Then ant will move back to F again, which is shown in Fig. 8(e). At this time, the obstacle node V and the valid taboo nodes G, I form a corner in the shape of the letter "L." Condition one and Condition two are ensured, ant continues to move back. If things went on like this, the ant will go back to node B as shown in Fig. 8 (f). Finally, ant may select the node J or K as the next node and get out of the trap. The U-type trap is filled with valid taboo nodes C, D, E, F, G, H, and I which are utilized to prevent all ants from entering the trap.

B. SHA COMBINED WITH THE MOTION CONSTRAINTS

Because the workspace is divided by cubes and the path is formed by the connections of adjacent grid nodes, to simplify the problem, the maximal turning angle and slop are taken as 45 degrees in this paper. Besides, if the maximal turning angle is less than 45, the successor nodes of ants can be selected from a wider range than the adjacent nodes. When the motion constraints of UAV are integrated into the searching process, the generation of valid taboo nodes is influenced. Several examples in 2-D maps are as following.

In Fig. 9 (a), the black nodes are ideal valid taboo nodes which generated above the green line. These taboo nodes can prevent ants from travelling deep inside the L-shaped corner. However, when the ants travel along the orange vectors to visit node B and D respectively, there is no available node to be selected because the constraints of UAV are considered. Based on the principles of the basic SHA, node B and D are marked as the valid taboo nodes. If more retreats happen in this area, the whole space would be filled by more valid taboo nodes and the result is as Fig 9 (b). Obviously, those nodes which under the green line should not be marked as the valid taboo nodes because they block the area which may contain the optimal path.

Another problem is depicted as Fig. 9 (c). When the constraints of UAV are considered, node A and node B will be viewed as the valid taboo nodes after the ant retreated to node C. As a result, the best path which is marked by the green line cannot be obtained. In addition, when the obstacles are

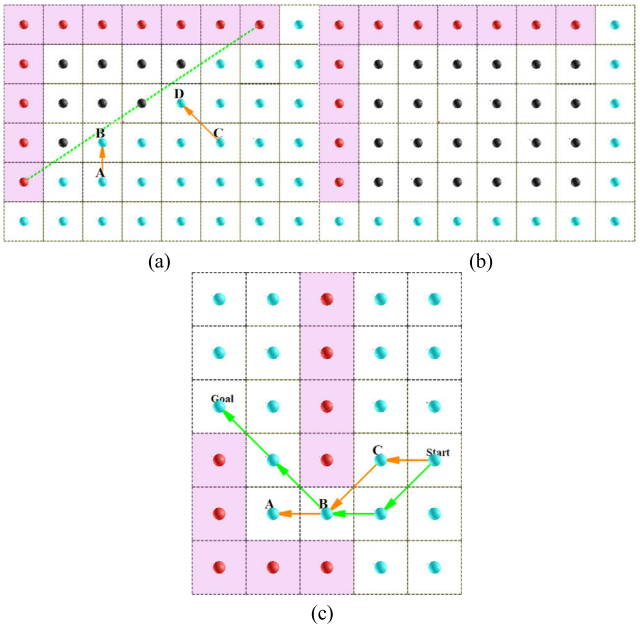


FIGURE 9. Problems caused by the motion constraints.

contained in the U-type obstacles, the valid taboo nodes could narrow the passage. More constraints would cause fewer optional nodes for ants, so deadlock state could easily occur in those cramped passages. Finally, more valid taboo nodes will be generated until the passage is completely filled, which is not expected. More seriously, the algorithm will not converge when the only path is blocked.

According to the above, the principles of basic SHA are not enough to generate valid taboo nodes when the motion constraints are considered. Therefore, more conditions are added to the basic SHA.

3) Condition three. The taboo node k must exist in the area which surrounded by the obstacles and the boarder. The border is the connection at both ends of the L-shaped corner. e.g., the green line in the Fig. 9 (a).

4) Condition four. Valid taboo nodes are produced layer by layer from the deepest node of the L-shaped corner. Besides, there must be enough space between the outermost layer and the internal obstacles, this layer could be formed by the taboo nodes and free nodes. When the origin or target is inside the trap, it can be viewed as the internal obstacle temporarily.

To understand the condition three and four, an example is shown below.

In Fig. 10, dotted lines of different colors represent the boarders. The nodes with the same color as a boarder are the valid taboo nodes in the region surrounded by this boarder and the corresponding L-shaped corner. e.g., the black nodes above are in the area surrounded by the black boarder and the upper left corner, which satisfies the condition three. These black nodes are generated layer by layer from node O. The third layer is formed by taboo node A, B, C and a free node, it is not adjacent to the internal obstacle, which is consistent with the condition four.

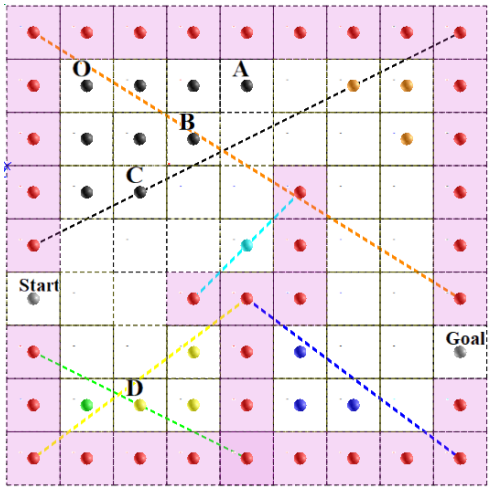


FIGURE 10. The SHA combined with new conditions.

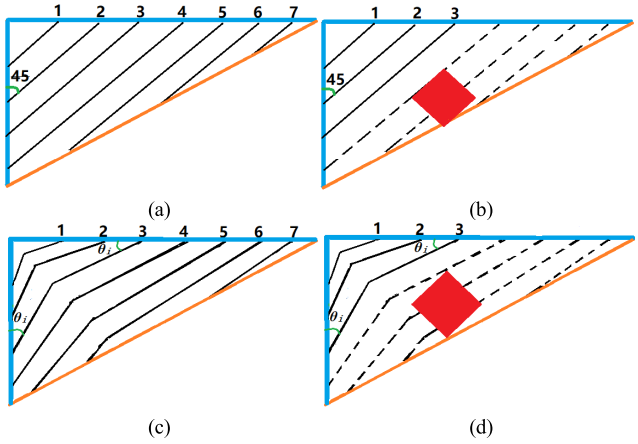
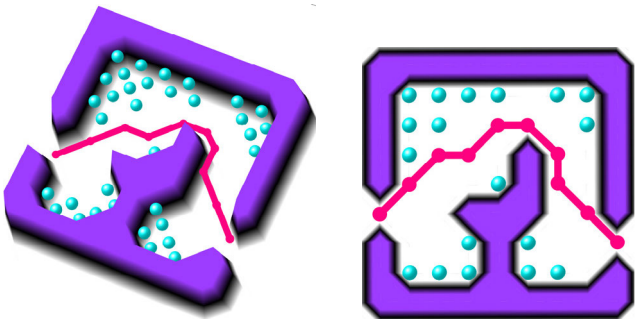


FIGURE 11. The difference of condition four when θ_i changes.

It is noteworthy that when the maximal turning angle θ_i is less than 45 degrees, the condition four would be slightly different. An example is as Fig. 11, where the orange lines are boarders, blue lines are corners, the red squares are obstacles, the layers of valid taboo nodes are represented by black solid lines.

If $\theta_i = 45$, the layers of valid taboo nodes are generated one by one at a 45 degree angle. When there is no obstacle in the U-type trap, the result is as Fig. 11 (a), if there is an obstacle, the result is as Fig. 11 (b). When $\theta_i < 45$, the layers of valid taboo nodes are generated in the shape of angle bracket as Fig. 11 (c). When an obstacle is in the U-type trap, the result is as Fig. 11 (d). If more space is needed between the obstacles, the third layer can be obviated too.

Under the influence of condition three, the problem depicted in the Fig. 9 (b) will not occur. Besides, condition four can be utilized to prevent the passage from becoming too narrow, i.e., the area containing the optimal path is retained and the convergence of ACO will not be affected by the SHA. With all the four conditions above, the valid taboo nodes are



(a) (b)

FIGURE 12. The result of applying SHA to ACO.

generated in right space and SHA can works effectively when the motion constraints of UAV are integrated.

C. THE SHA IN 3-D WORKSPACE

The result of applying SHA to ACO in a 3-D map is shown as Fig. 12 (a) where the pink line is the optimal path and it satisfies the motion constraints. Valid taboo nodes represented by blue dots are spread around from the innermost and deepest part of U-type obstacles, the sufficient space between complex obstacles is maintained to produce the best path. The corresponding top view is as Fig. 12 (b). Compared with Fig. 10, node B and C are free nodes. This is because the places where the ants retreat from are random. In this experiment, the ants did not move back from node B and C. i.e., the valid taboo nodes in Fig. 10 are generated ideally, and the number of them may be less in reality. In addition, the valid taboo node cannot be equivalent to the obstacle node.

By applying SHA, a new communion mechanism between each ant is established besides the pheromone matrix. In this mechanism, ants who complete the path planning leave messages (namely, the valid taboo nodes) inside the traps to warn the younger generations against entering the same traps. Furthermore, the more the ants retreat in U-type traps, the more complete the traps would be filled by the valid taboo nodes, and the less likely the new generations are to enter the same traps.

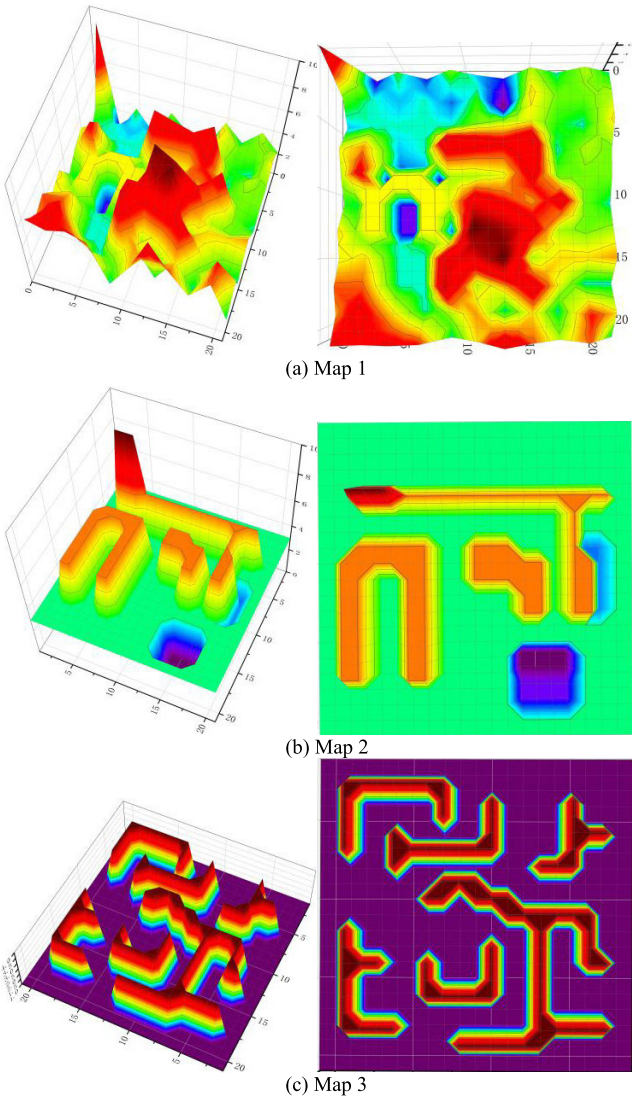
VI. PERFORMANCE EVALUATION

In this section, the simulation contains two parts. In part one, DM, EM1, and EM2 are applied to ACO in one map respectively. In part two, the fallback strategy and SHA are applied to ACO in different maps respectively. In order to see the U-type obstacles clearly, all maps and their top views are as following.

Before the simulation, parameters listed in Table 3 are optimized by MPGA and the values obtained are adopted as the initial values of eight corresponding parameters in ACO. Technically, this process should be executed whenever the experimental conditions change, which is consuming and troublesome. Here we only provide two groups of results,

TABLE 5. The values of parameters optimized by MPGA.

α	β	ρ_1	ρ_2	m	Q	N_c_max	$\tau_{ij}^{initial}$
1	6	0.1	0.5	47	216	158	32
1	10	0.1	0.4	42	468	120	32



(a) Map 1

(b) Map 2

(c) Map 3

FIGURE 13. Maps and the obstacles.

which belongs to map 2 and map 3 respectively, as shown in Table 5.

A. PERFORMANCES OF EXTENSION MODE 1 AND EXTENSION MODE 2

In Fig. 13 (a), DM, EM1, and EM2 are applied to ACO respectively. When the motion constraints are not considered, two groups of experiments are implemented. In group one, set the origin to (21, 16, 6) and the destination to (1, 10, 5). In group two, set the origin to (12, 6, 2) and the destination

to (12, 19, 5). In each group, basic ACO is executed 50 times and the average value of deadlock state is recorded in Table 6.

B. PERFORMANCES OF SHA

In the last two maps of Fig. 13, several groups of the origins and goals are set, then the SHA and fallback strategy are applied to ACO respectively according to Table 7.

Based on the order in Table 7, simulation results are as following. Where the pink line in maps is the optimal path and the valid taboo nodes are represented by black dots. The number of retreats and the running time when the algorithm is executed 50 times are provided too. The blue dots represent SHA, the pink dots signify fallback strategy, the pink line indicates the average value of fallback strategy, the blue line denotes the average value of SHA.

1) MAP 2

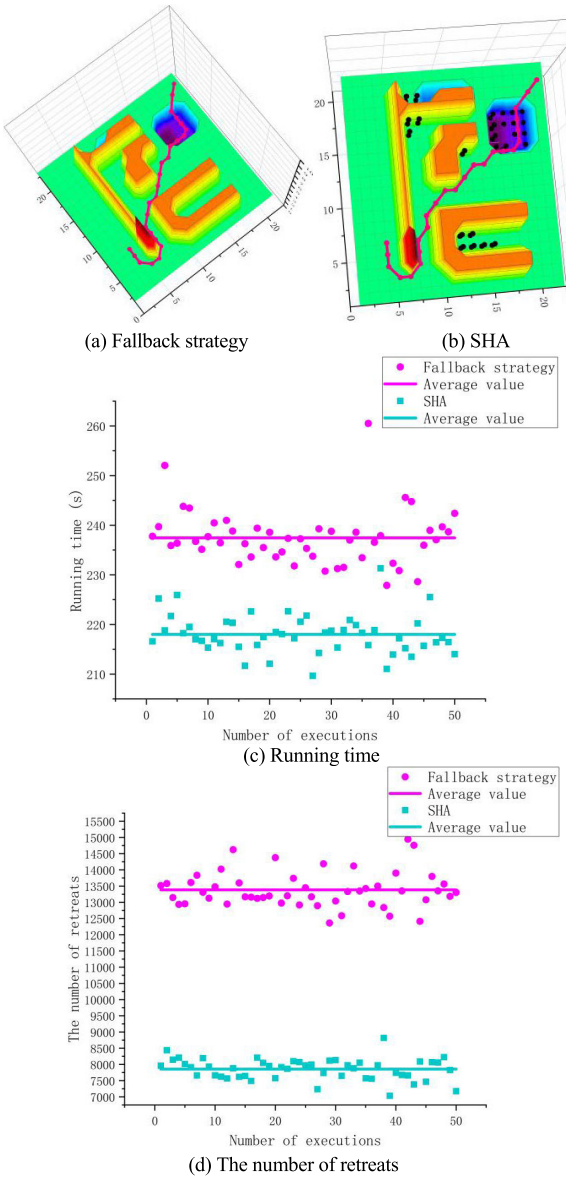


FIGURE 14. Map 2, Group 1, Line 1.

2) MAP 3

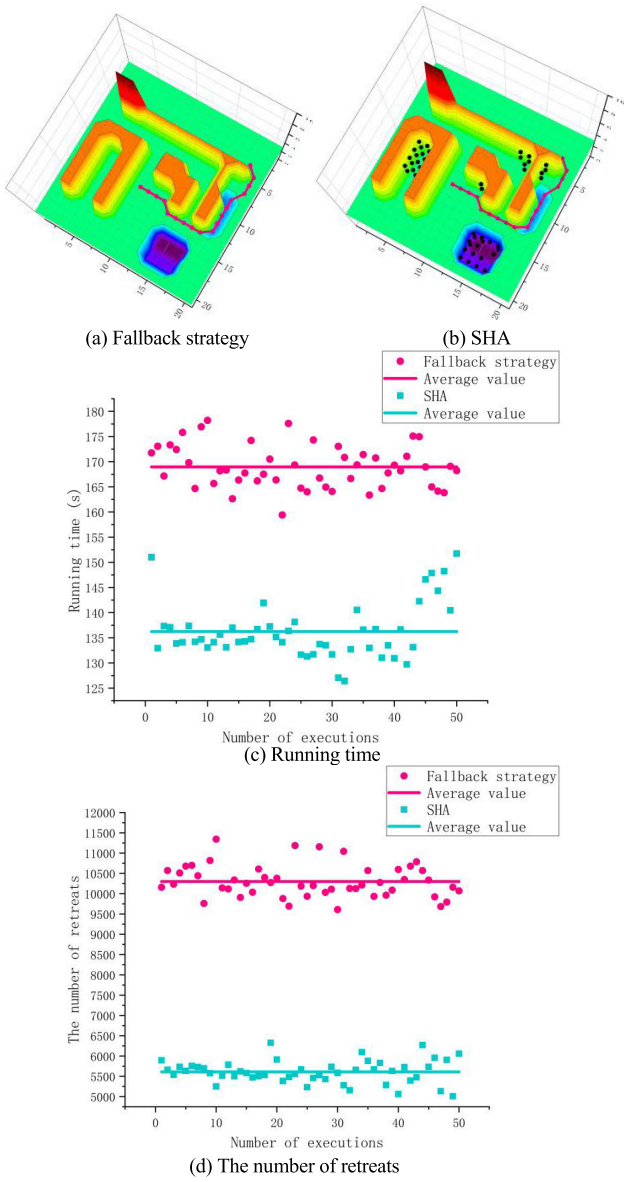


FIGURE 15. Map 2, Group 1, Line 2.

TABLE 6. The results of deadlock state using three different methods.

Methods	Deadlock State	
	Group 1	Group 2
Default Mode	616	883
Extension Mode 1	281	185
Extension Mode 2	0	0

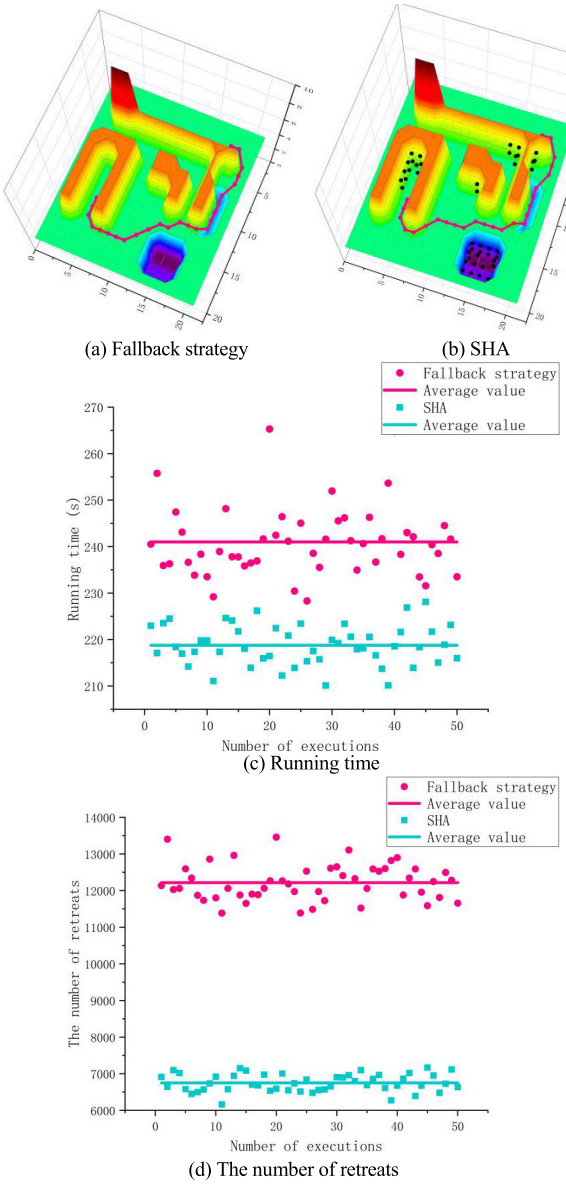
C. ANALYSIS

1) DM, EM1, AND EM2

In Table 6, the times of deadlock state occurs is reduced when the EM1 is applied and it can be completely eliminated by applying EM2. Therefore, the greater the degree of freedom when ants select the next node, the less likely the deadlock state will occur.

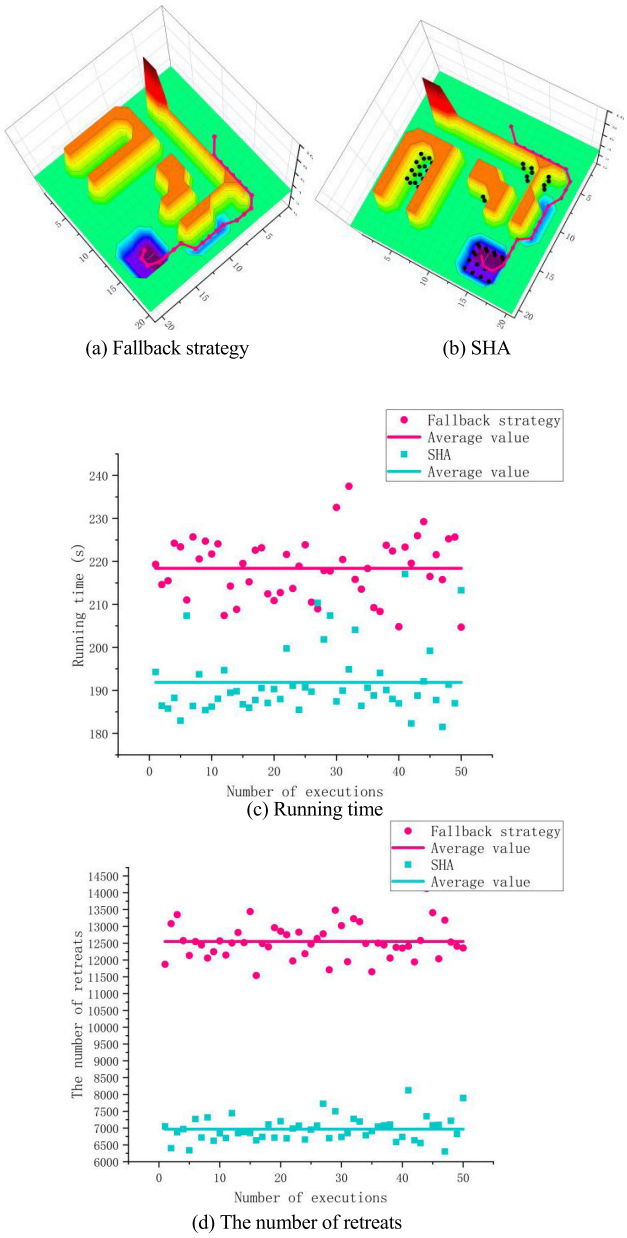
TABLE 7. Settings of each simulation.

Map	Group	COORDINATE	
		Origin	goal
2	1	(20, 19, 4)	(4, 6, 4)
		(13, 10, 4)	(5, 19, 4)
	2	(15, 6, 4)	(5, 18, 5)
		(17, 15, 2)	(4, 12, 5)
3	1	(20, 20, 3)	(2, 2, 2)
		(9, 15, 3)	(2, 2, 2)
	2	(20, 2, 3)	(6, 13, 2)
		(15, 5, 3)	(5, 20, 2)



(a) Fallback strategy (b) SHA
(c) Running time
(d) The number of retreats

FIGURE 16. Map 2, Group 2, Line 1.



(a) Fallback strategy (b) SHA
(c) Running time
(d) The number of retreats

FIGURE 17. Map 2, Group 2, Line 2.

2) THE FALLBACK STRATEGY AND SHA

In the second chart of each figure from Fig. 14 to Fig. 21, although the origins and the goals are different, all the paths obtained can fulfil the motion constraints and the valid taboo nodes are only present inside the U-type obstacles. Since the height of most U-type obstacles in the map is greater than the maximum flying height predefined, the UAV usually bypasses from the sides of obstacles in the horizontal plane to jump out of U-type obstacles. However, when the space above the obstacles is reachable, UAV can directly pass over obstacles from the area above, e.g., the drone flies through certain areas above the pit-shaped obstacle in map 2. Most importantly, when other obstacles are located inside the U-type traps, SHA can reserve enough space between obstacles for the drone to pass through, which prevents the ACO from not converging or even failing.

In the last two charts of each figure from 14 to 21, the average number of retreats (namely, the number of times that the ants move backward) and the average running time of

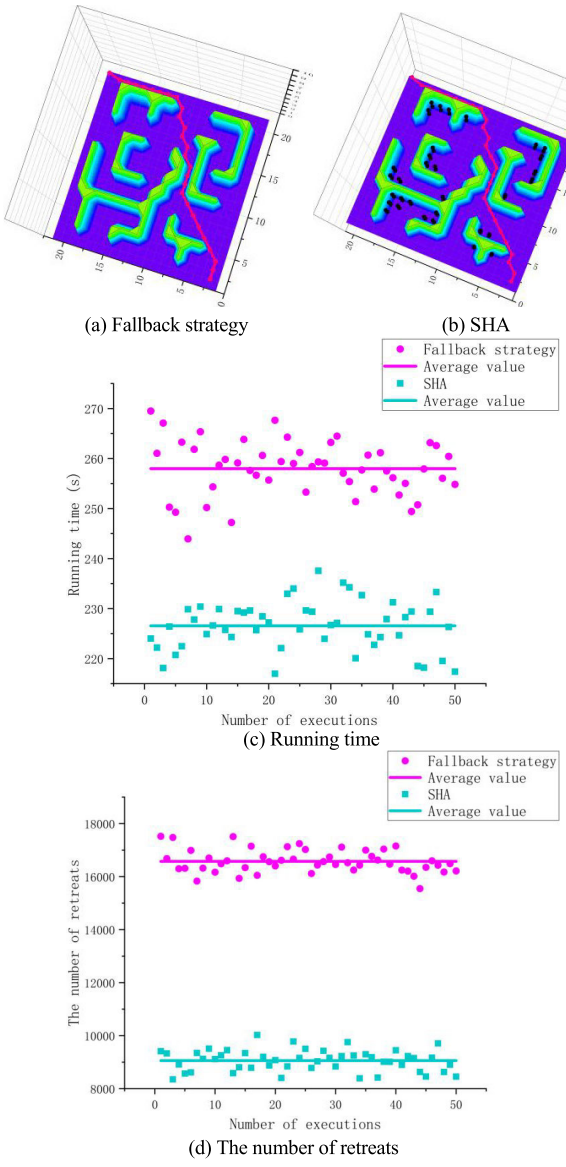


FIGURE 18. Map 3, Group 1, Line 1.

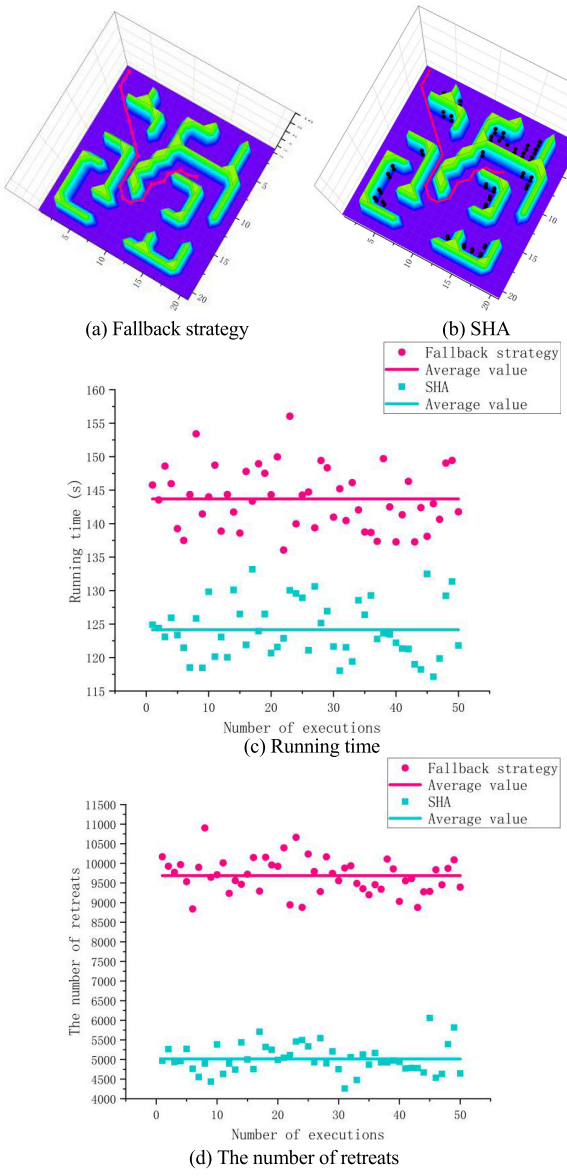


FIGURE 19. Map 3, Group 1, Line 2.

algorithm when applying SHA are less than that when the general fallback strategy is applied. Thus, the effectiveness of SHA is verified. However, not all the number of retreats and the running time of algorithm when SHA is applied are smaller than that when fallback strategy is applied. The reason could be that the workspace with many concave obstacles is complex and certain randomization is contained in ACO when searching for new nodes. e.g., if it goes well that the ants reach the goal without entering many concave obstacles, the number of retreats and the running time of the algorithm could be low too even without the SHA, because SHA will only take effect when the ant enters the U-type traps. Besides, retreats may also exist outside the concave obstacles, although the number of retreats inside U-type traps may be decreased by the SHA, the number of retreats outside the U-type traps may be pretty high. In this case, the number of retreats when SHA is applied could be high.

By comparing the last two charts of each figure from 14 to 21, the number of retreats could be low in spite of a long time taken by algorithm. i.e., the running time of the algorithm is not just depend on the number of retreats. In a simulation, the ants may not retreat frequently, but the optimal path obtained in each generation of the ants may be quite different or in low quality, so the convergence speed of ACO could be relatively slow. In this case, although the number of retreats could be obviously reduced by SHA, the total time consumed by algorithm cannot be lowered.

But still, under the influence of valid taboo nodes in the SHA, ants are effectively prevented from revisiting partial nodes which inside the concave obstacles, i.e., the size of space that ants can traverse is shrunk. As a result, the running time of algorithm is reduced. To thoroughly explain how SHA affects the ACO , more analyses are as following.

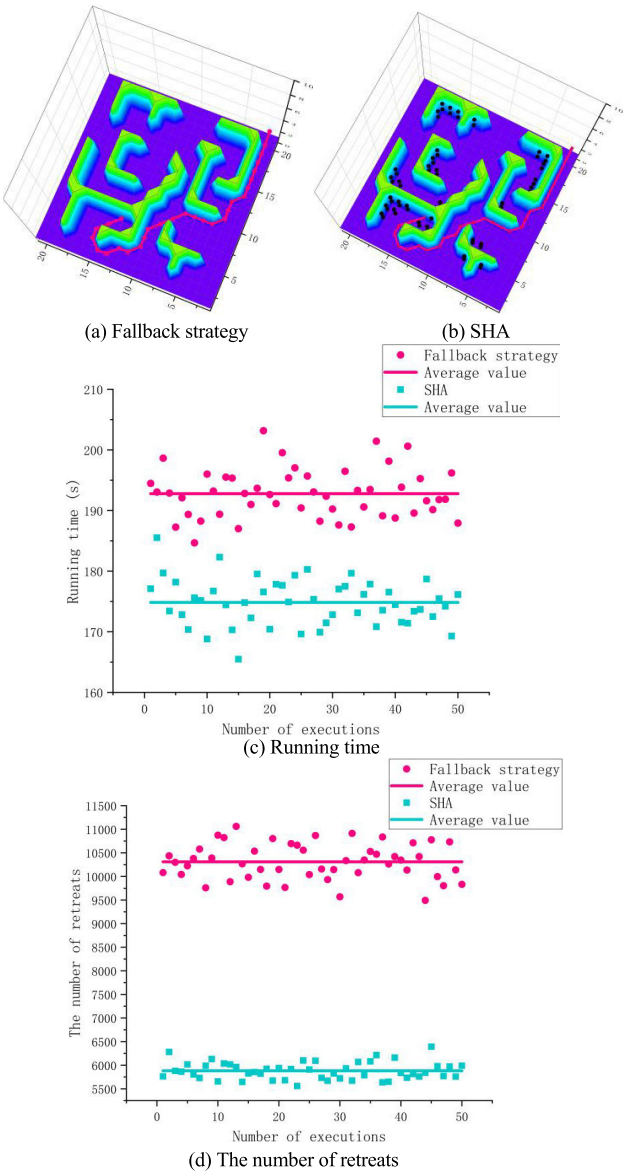


FIGURE 20. Map 3, Group 2, Line 1.

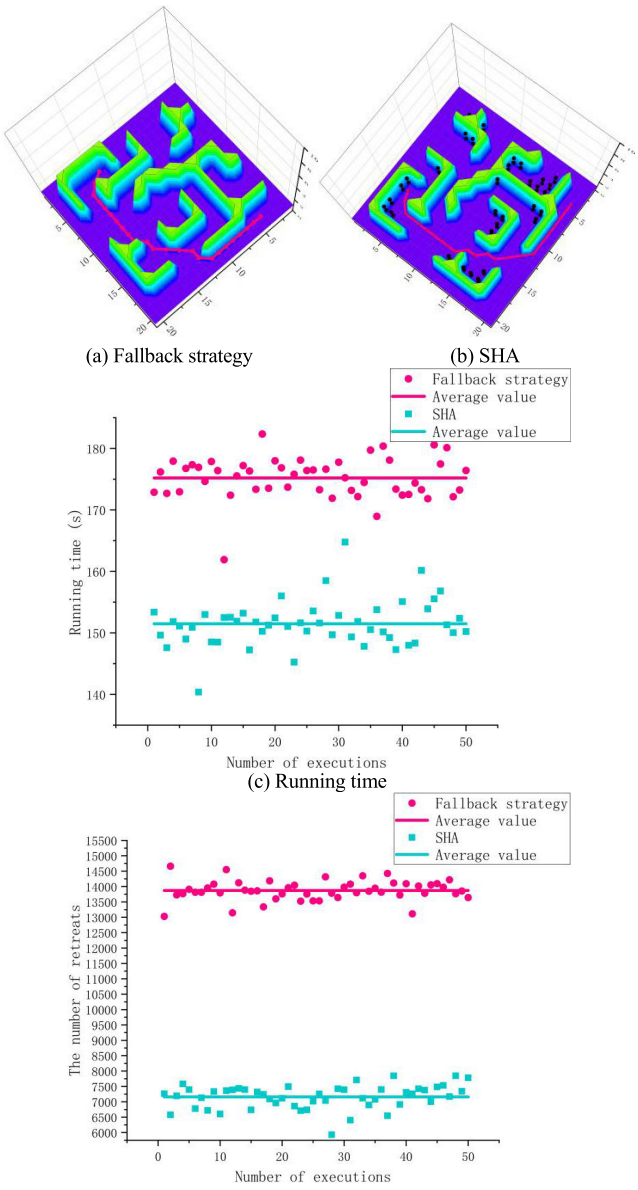


FIGURE 21. Map 3, Group 2, Line 2.

In Fig. 22 (a), there are five flyable paths between the origin and the goal, which are symbolized by different colors. Based on the principles of SHA, several nodes would be marked as the valid taboo nodes, which are represented by black dots and the result is as Fig 22 (b). They can prevent the ants from moving back from the location of these nodes. In addition, the blue path is excluded and the ants can only select the optimal path from the remaining four paths. Compared to the five paths in Fig. 22 (a), the scale of pathfinding problem is reduced and ants can focus on exploring in the areas which may contain the best path.

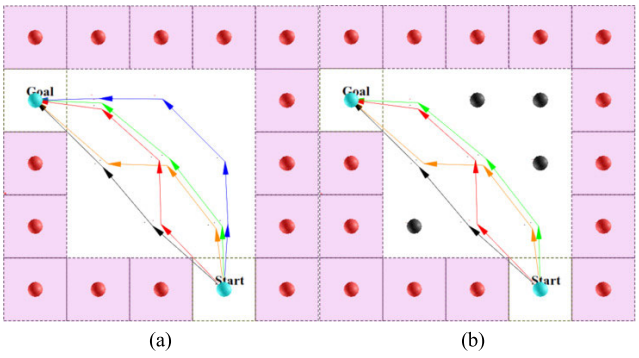
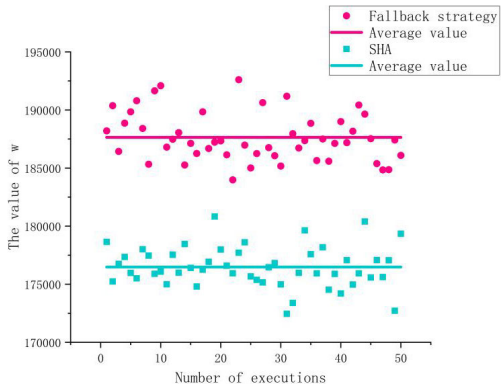
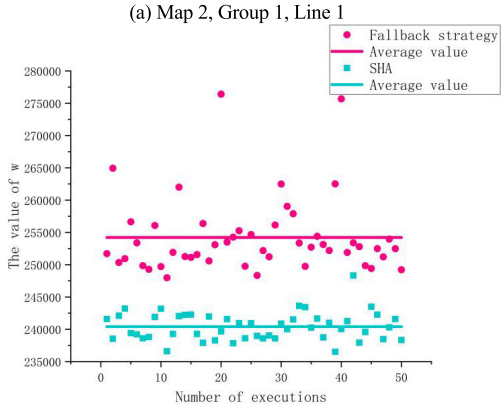


FIGURE 22. The size of space affected by SHA.

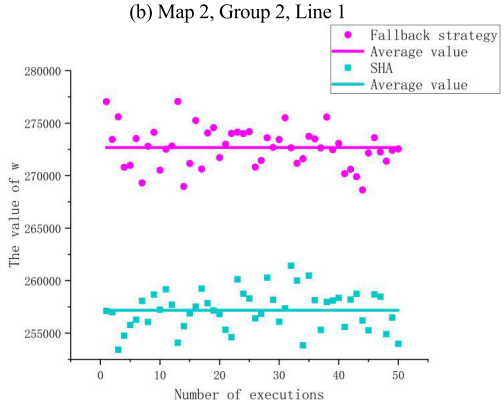
Except the above, the time complexity is analyzed. In most literature, the complexity of ACO is analyzed on travelling salesman problem (TSP). Set the number of iterations to N_c, the number of ants in each generation to m, and the size of cities to n. Each ant needs to select a new city from n cities once it arrives a city, set the number of executions of this process to w. Because all the n cities need to be visited, therefore w equals n and the complexity is $O(N_C \times n^2 \times m)$



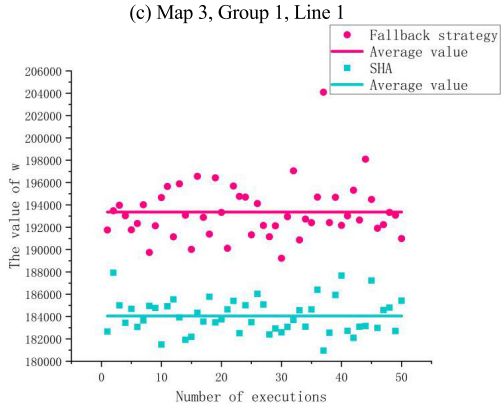
(a) Map 2, Group 1, Line 1



(b) Map 2, Group 2, Line 1



(c) Map 3, Group 1, Line 1



(d) Map 3, Group 2, Line 1

FIGURE 23. **The value of *w*.**

if the lower power is ignored. When ACO is used for 3-D path planning, the number of nodes adjacent to the current node is 26. Each ant needs to select the next node from 26 adjacent nodes once it reaches a new node and this process is executed w times. Therefore the complexity is $O(N_C \times 26 \times m \times w)$ when the lower power is ignored. However in the path planning problem, the number of nodes which visited by each ant before it reaches the goal is uncertain, so unlike TSP, w is unknown. Based on above, it might be hard to determine an accurate value of the complexity with the O-notation in the path planning problem. In order to compare the complexity between the ACO with SHA applied and the ACO with fallback strategy applied, the values of w in the simulations above are recorded and partial results are shown in Fig. 23.

If the average value of w in the ACO with SHA applied is X_1, and the average value of w in the ACO with fallback strategy applied is X_2, according to Fig. 23, $O(N_C \times 26 \times m \times X_1) < O(N_C \times 26 \times m \times X_2)$. Thus, when the lower power is ignored, the time complexity of applying SHA to ACO is lower than it when fallback strategy is applied. In addition, the value of w in the ACO with SHA applied is not necessarily less than it in the ACO with fallback strategy applied, the reason can be referred in the discussion about the running time and the number of retreats.

D. LIMITATIONS

1) THE LIMITATIONS OF OBSTACLES

In this research, dynamic obstacles are not considered. We think the path planning can be briefly divided into global planning and local planning. For the former, by applying ACO, the initial optimal path from the start to goal is produced based on the known map information, then the vehicle travels along this initial path. For the latter, when travelling along the initial path, the vehicle may have to leave the initial path to avoid new obstacles ahead or the dynamic obstacles. Once the vehicle has avoided the obstacles, the ACO is applied again and a new initial optimal path would be generated from the current location to goal. Because the purpose of SHA is to make ACO better generate initial optimal path in an environment full of U-type obstacles, which belongs to the global planning, therefore, the dynamic obstacles which belongs to local planning is not considered. But still, to apply the algorithm to an real scene, local planning should be combined, which is our next step.

2) THE LIMITATIONS OF VALID TABOO NODES

In the simulation section, it is shown that the interior of some U-type obstacles are not fully filled, and some valid taboo nodes in Fig. 15 (b) are slightly different from what it is in Fig. 16 (b). Firstly, valid taboo nodes are generated based on the L-shaped corner, so there must be an area inside the U-type obstacle that cannot be filled by valid taboo nodes. Secondly, under the influence of condition 1 and 2, although the valid taboo nodes can be generated layer by layer, the ants may not move back from the expected node which should be marked as the valid taboo nodes. As a result, condition 1

cannot be fulfilled to fill the traps and valid taboo nodes may be different in different experiments. Therefore, SHA remains to be improved to fill the U-type obstacles more complete and effective.

VII. CONCLUSION

In this paper, EM1 and EM2 which modified search strategy to improve the deadlock state were discussed. The former one prevents ants from revisiting the previous path node of current node while the latter one allows ants to revisit all path nodes. To solve the problem of redundant path segments caused by these two methods, the Splicing method and the Dijkstra method are adopted. By simulation, the relationship between the degree of freedom when searching for the next node and the deadlock state occurs was studied. These new search strategies could improve the deadlock state though the efficiency was not high enough. In order to improve the efficiency of fallback strategy and make full use of each ant as well as the taboo nodes, SHA was proposed and a new communion mechanism between each ant was established besides the pheromone matrix. The performance of SHA was evaluated by a lot of experiments and the results indicated that the number of retreats could be reduced effectively, more importantly, the time performance of ACO was improved. In addition, the reasons why SHA can improve the ability of fallback strategy are analyzed from the perspectives of time complexity and the scale of path-finding problem. At last, based on the limitations of SHA, further studies might focus on how to fill the U-type obstacles more complete and effective, how to combine the dynamic obstacles. e.g., to completely fill the traps, valid taboo nodes could be generated based on the U-type obstacle rather than just the L-shaped corner. Another strategy for using valid taboo nodes more efficiently is to reserve them whenever the algorithm is done. In this way, when the starting points and targets are changed, some of these valid taboo nodes can still be used, which can shrink the entire flyable space before the algorithm starts and save a lot of time spent generating the valid taboo nodes.

ACKNOWLEDGMENT

The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] L. D. Burns, "Sustainable mobility: A vision of our transport future," *Nature*, vol. 497, no. 7448, pp. 181–182, May 2013.
[2] S. L. Vine, A. Zolfaghari, and J. Polak, "Autonomous cars: The tension between occupant experience and intersection capacity," *Transp. Res. C, Emerg. Technol.*, vol. 52, pp. 1–14, Mar. 2015.
[3] V. E. Balas and M. M. Balas, "Driver assisting by inverse time to collision," in *Proc. World Autom. Congr.*, Budapest, Hungary, Jul. 2006, pp. 1–6.
[4] M. Caccia, R. Bono, and G. B. Berge, "Variable-configuration UUVs for marine science applications," *IEEE Robot. Autom. Mag.*, vol. 6, no. 2, pp. 22–23, Jun. 1999.
[5] P. R. Bandyopadhyay, "Trends in biorobotic autonomous undersea vehicles," *IEEE J. Ocean. Eng.*, vol. 30, no. 1, pp. 109–139, Jan. 2005.
[6] C. Zheng, L. Li, F. Xu, F. Sun, and M. Ding, "Evolutionary route planner for unmanned air vehicles," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 609–620, Aug. 2005.
[7] B. Yang and W. Liu, "A improved method of robot's path planning based visibility graph," *Comput. Knowl. Technol.*, vol. 5, no. 2, pp. 434–435, Feb. 2009.
[8] J. Luo, W. Su, and D. Wang, "The improvement of the artificial potential field robot path planning based on 3-D space," in *Proc. Int. Conf. Autom. Control Artif. Intell.*, Xiamen, China, Mar. 2012, pp. 2128–2131.
[9] P. G. Tzionas, A. Thanailakis, and P. G. Tsalides, "Collision-free path planning for a diamond-shaped robot using two-dimensional cellular automata," *IEEE Trans. Robot. Autom.*, vol. 13, no. 2, pp. 237–250, Apr. 1997.
[10] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of A," *J. ACM*, vol. 32, no. 3, pp. 505–536, Jul. 1985.
[11] T. Chen, G. Zhang, X. Hu, and J. Xiao, "Unmanned aerial vehicle route planning method based on a star algorithm," in *Proc. ICIEA*, Wuhan, China, May/Jun. 2018, pp. 1510–1514.
[12] K. Gopalakrishnan, S. Ramakrishnan, and C. Dagli, "Optimal path planning of mobile robot with multiple target using ant colony optimization," in *Proc. Intell. Eng. Syst. Through Artif. Neural Netw.*, New York, NY, USA, vol. 16, 2006, pp. 25–30.
[13] M. A. P. Garcia, O. Montiel, R. Castillo, R. Sepúlveda, and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Appl. Soft Comput.*, vol. 9, no. 3, pp. 1102–1110, Jun. 2009.
[14] C. Zhang, Z. Zhen, D. Wang, and M. Li, "UAV path planning method based on ant colony optimization," in *Proc. Chin. Control Decis. Conf.*, Xuzhou, China, May 2010, pp. 3790–3792.
[15] S.-H. Chia, K.-L. Su, J.-H. Guo, and C.-Y. Chung, "Ant colony system based mobile robot path planning," in *Proc. 4th Int. Conf. Genetic Evol. Comput.*, Shenzhen, China, Dec. 2010, pp. 210–213.
[16] H. Duan, D. Wang, and X. Yu, "Review on research progress in ant colony algorithm," *Chin. J. Nature*, vol. 28, no. 2, pp. 102–105, 2006.
[17] S. Liu, Y. Tian, and J. Liu, "Multi mobile robot path planning based on genetic algorithm," in *Proc. 5th. World Congr. Intell. Control Automat.*, Hangzhou, China, Jun. 2004, pp. 4706–4709.
[18] Q. Li, X. Tong, and S. Xie, "Optimum path planning for mobile robots based on a hybrid genetic algorithm," in *Proc. 6th Int. Conf. Hybrid Intell. Syst. (HIS)*, Rio de Janeiro, Brazil, Dec. 2006, pp. 4–7.
[19] B. B. V. L. Deepak and D. Parhi, "Intelligent adaptive immune-based motion planner of a mobile robot in cluttered environment," *Intell. Service Robot.*, vol. 6, no. 3, pp. 155–162, Jul. 2013.
[20] M. Yuan, S. Wang, C. Wu, and N. Chen, "A novel immune network strategy for robot path planning in complicated environments," *J. Intell. Robot. Syst.*, vol. 60, no. 1, pp. 111–131, Oct. 2010.
[21] P. K. Mohanty and D. R. Parhi, "Cuckoo search algorithm for the mobile robot navigation," in *Proc. Int. Conf. Swarm, Evol., Memetic Comput.*, 2013, pp. 527–536.
[22] X. Liang, L. Li, J. Wu, and H. Chen, "Mobile robot path planning based on adaptive bacterial foraging algorithm," *J. Central South Univ.*, vol. 20, no. 12, pp. 3391–3400, Dec. 2013.
[23] L. Lu and D. Gong, "Robot path planning in unknown environments using particle Swarm optimization," in *Proc. 4th Int. Conf. Natural Comput.*, Jinan, China, Oct. 2008, pp. 422–426.
[24] C.-F. Juang and Y.-C. Chang, "Evolutionary-group-based particle-Swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 2, pp. 379–392, Apr. 2011.
[25] H. Mo and L. Xu, "Research of biogeography particle Swarm optimization for robot path planning," *Neurocomputing*, vol. 148, pp. 91–99, Jan. 2015.
[26] M. Saska, M. Macas, L. Preucil, and L. Lhotska, "Robot path planning using particle Swarm optimization of Ferguson splines," in *Proc. IEEE Conf. Emerg. Technol. Factory Autom.*, Prague, Czech Republic, Sep. 2006, pp. 833–839.
[27] Y. Zhang, D.-W. Gong, and J.-H. Zhang, "Robot path planning in uncertain environment using multi-objective particle Swarm optimization," *Neurocomputing*, vol. 103, pp. 172–185, Mar. 2013.
[28] S. H. Tang, F. Kamil, W. Khaksar, N. Zulkifli, and S. A. Ahmad, "Robotic motion planning in unknown dynamic environments: Existing approaches and challenges," in *Proc. IEEE Int. Symp. Robot. Intell. Sensors (IRIS)*, Langkawi, Malaysia, Oct. 2015, pp. 288–294.
[29] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.

[30] M. Dorigo and T. Stützle, "Ant colony optimization: Overview and recent advances," in *Handbook of Metaheuristics* (International Series in Operations Research Management Science), vol. 146, M. Gendreau and J. Y. Potvin, Eds. Boston, MA, USA: Springer, 2010, pp. 227–263.
[31] G. Liu, T. Li, Y. Peng, and X. Hou, "The ant algorithm for solving robot path planning problem," in *Proc. 3rd Int. Conf. Inf. Technol. Appl. (ICITA)*, Sydney, NSW, Australia, Jul. 2005, pp. 25–27.
[32] Z. Wen and Z. Cai, "Global path planning approach based on ant colony optimization algorithm," *J. Central South Univ. Technol.*, vol. 13, no. 6, pp. 707–712, Dec. 2006.
[33] Q. Zhu and L. Wang, "A new algorithm for robot path planning based on scout ant cooperation," in *Proc. 4th Int. Conf. Natural Comput.*, Jinan, China, Oct. 2008, pp. 444–449.
[34] C. Cui, N. Wang, and J. Chen, "Improved ant colony optimization algorithm for UAV path planning," in *Proc. Int. Conf. Softw. Eng. Service Sci.*, Beijing, China, Jun. 2014, pp. 291–295.
[35] Y. Shan, "Study on submarine path planning based on modified ant colony optimization algorithm," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Changchun, China, Aug. 2018, pp. 288–292.
[36] H. Wang and W. Xiong, "Research on global path planning based on ant colony optimization for AUV," *J. Marine Sci. Appl.*, vol. 8, no. 1, pp. 58–64, Mar. 2009.
[37] M. Yuan, S. Wang, and P. Li, "A model of ant colony and immune network and its application in path planning," in *Proc. IEEE Conf. Ind. Electron. Appl.*, Singapore, Jun. 2008, pp. 102–107.
[38] H. Duan, Y. Yu, X. Zhang, and S. Shao, "Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm," *Simul. Model. Pract. Theory*, vol. 18, no. 8, pp. 1104–1115, Sep. 2010.
[39] K. Ioannidis, G. C. Sirakoulis, and I. Andreadis, "Cellular ants: A method to create collision free trajectories for a cooperative robot team," *Robot. Auto. Syst.*, vol. 59, no. 2, pp. 113–127, Feb. 2011.
[40] Z. Zhou, Y. Nie, and G. Min, "Enhanced ant colony optimization algorithm for global path planning of mobile robots," in *Proc. Int. Conf. Comput. Inf. Sci.*, Shiyang, China, Jun. 2013, pp. 698–701.
[41] J. Zhao, X. Gao, X. Fu, and J. Liu, "Improved ant colony algorithm of path planning for mobile robot," *Control Theory Appl.*, vol. 28, no. 4, pp. 457–461, Apr. 2011.
[42] G. Ma, H. Duan, S. Liu, and Y. Yu, "UCAV path planning based on MAX-MIN self-adaptive ant colony optimization," *Acta Aeronautica Et Astronautica Sinica*, vol. 29, no. 5S, pp. 243–248, 2008.
[43] C.-C. Hsu, R.-Y. Hou, and W.-Y. Wang, "Path planning for mobile robots based on improved ant colony optimization," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Manchester, U.K., Oct. 2013, pp. 2777–2782.
[44] R. Claes and T. Holvoet, "Ant colony optimization applied to route planning using link travel time predictions," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Workshops Phd Forum*, Shanghai, China, May 2011, pp. 358–365.
[45] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behavior," *Nature*, vol. 406, no. 6791, pp. 39–42, Jul. 2000.
[46] Jayadeva, S. Shah, A. Bhaya, R. Kothari, and S. Chandra, "Ants find the shortest path: A mathematical proof," *Swarm Intell.*, vol. 7, no. 1, pp. 43–62, Mar. 2013.
[47] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Dept. Electron. Eng., Politecnico di Milano, Milan, Italy, 1992.

CHAO ZHANG was born in Hubei, China, in 1977. He received the M.S. degree in communication and information system and the Ph.D. degree in information and communication engineering from Northwestern Polytechnical University, Xi'an, China, in 2004 and 2009, respectively. Since 2004, he has been with Northwestern Polytechnical University, where he is currently a Professor of electronic information engineering with the School of Aeronautics. From March 2013 to February 2014, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Connecticut, CT, USA. His research interests are in the areas of system fault diagnosis and prognosis, reliability analysis, unmanned aerial vehicle, and machine learning.

CHENXI HU was born in Ningxia, China, in 1994. He received the B.S. degree in electrical engineering and automation from Northwestern Polytechnical University, Xi'an, China, in 2017, where he is currently pursuing the M.S. degree in aviation engineering. His main research interests include path planning of unmanned aerial vehicle and artificial intelligent algorithms.

JIANRUI FENG was born in Ningxia, China, in 1994. He received the B.S. degree in electrical information engineering from Northwestern Polytechnical University, Xi'an, China, in 2017, where he is currently pursuing the M.S. degree in aviation engineering. His current research interest includes fault diagnosis.

ZHENBAO LIU (M'11–SM'18) was born in Shandong, China, in 1979. He received the B.S. and M.S. degrees in electrical engineering and automation from Northwestern Polytechnical University, Xi'an, China, in 2001 and 2004, respectively, and the Ph.D. degree in electrical engineering and automation from the University of Tsukuba, Tsukuba, Japan, in 2009. He was a Visiting Scholar with Simon Fraser University, Canada, in 2012. He is currently a Professor with Northwestern Polytechnical University. His research interests include unmanned aerial vehicle, prognostics and health management, and aircraft fault diagnosis. He is currently an Associate Editor of IEEE Access.

YONG ZHOU was born in Beijing, China, in 1978. He received the Ph.D. degree in vehicle engineering from Northwestern Polytechnical University, Xi'an, China, in 2011. In 2001, he joined the School of Aeronautics, Northwestern Polytechnical University, China, as an Instructor, and became an Associate Professor, in 2013. His research interests include modeling and control, motion control systems, and electric vehicles.

ZEXU ZHANG was born in Ningxia, China, in 1994. He received the B.S. degree in mechanical design, manufacturing and automation and the M.S. degree in shipbuilding and oceanography engineering from Northwestern Polytechnical University, Xi'an, China, in 2016 and 2019, respectively. His research interests include mechanism design, dynamics and control, and bio-inspired robotics.