IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 1, 2020, accepted April 14, 2020, date of publication April 24, 2020, date of current version May 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990153

Bi-Directional Adaptive A* Algorithm Toward Optimal Path Planning for Large-Scale UAV Under Multi-Constraints

XUELI WU[1,2], LEI XU[1], RAN ZHEN[1,2], AND XIAOJING WU[1,2]
[1]School of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China
[2]Hebei Engineering Technology Research Center of Production Process Automation, Shijiazhuang 050018, China

Corresponding author: Xiaojing Wu (wuxiaojing013@163.com)

This work was supported in part by the National Defense Basic Scientific Research Project JC***, in part by the Higher School Science and Technology Research Project of Hebei Province under Grant BJ2017041, and in part by the Hebei Graduate Innovation Funding Project under Grant CXZZSS2020098.

ABSTRACT For large-scale Unmanned Aerial Vehicle (UAV), it is a challenging problem to automatically generate an optimal path considering the multi-constraints. In order to improve the efficiency and optimality of conventional A* algorithm, Bi-directional adaptive A* algorithm is proposed in this paper. Regarding to the defect of expansion efficiency for conventional A* algorithm based on multi-direction, directional search strategy is adopted to improve the efficiency of expansion process and guarantee the smooth of the path. Adaptive step strategy and adaptive weight strategy are used to improve the exploration speed and the accuracy of the proposed algorithm. By adopting bi-directional search strategy, the exploration ability of the algorithm can be improved in some complex environment. Finally, rewiring process is introduced to optimize the path length. The simulation for UAV path planning under the multi-constraints conditions is carried out and the simulation results show that Bi-directional adaptive A* algorithm has the superiority in run time and path quality.

INDEX TERMS A* algorithm, adaptive strategy, unmanned aerial vehicle, path planning.

I. INTRODUCTION

Large-scale Unmanned Aerial Vehicle (UAV) [1] has good performance in terms of mobility and maneuverability under the complex environment. As the development of the smart devices and the algorithm, large-scale UAV is widely used in various fields, such as search and rescue [2], coastal monitoring [3], surveillance system [4], and so on. As the key technology of the large-scale UAV, path planning algorithm has recently gained great attention in the study of aerospace control community. At present, the primary algorithms of the path planning can be divided into two fields: traditional path planning algorithm and swarm intelligence optimization algorithm [5]. Artificial potential field algorithm [6] and simulated annealing [7] belong to the former, which are simplicity of operator and easy to use. However above algorithms have the problem of local minima. With the development of swarm intelligence technology, population-based evolutionary algorithms have made great progress in recent years [8], [9]. Common swarm intelligence optimization algorithms include Ant Colony Optimization (ACO) [10], Particle Swarm Optimization (PSO) [11], and Genetic algorithm(GA) [12]. Swarm intelligence optimization algorithms can efficiently solve the path planning problem with good robustness. However, according to the evolution principle of swarm intelligence optimization algorithms, algorithms with more uniform initial swarm distribution are easier to obtain global optimal solution [13], which means that the optimization effect of those algorithms relies on the initial distribution of the swarm. In order to avoid the local minima and reduce the dependence of the initial swarm, the sampling-based algorithms (e.g., rapidly-exploring random tree star (RRT*) [14], A* algorithm [15]) are introduced to provide a better solution for UAV path planning problem.

Among them, the path generated by A* algorithm based on graph search can converge to the optimal solution. However, the path quality generated by A* algorithm is to a great extent depends on the number of candidate nodes in the expansion process, thus, A* algorithm converges to optimal solution very slowly due to the inefficient node expansion process.

The associate editor coordinating the review of this manuscript and approving it for publication was P. K. Gupta.

Based on the conventional A* algorithm, a lot of works have been done by the researchers. Aimed at the dynamic obstacles, constrained A* algorithm [16] generated the safer path by creating a circular boundary for ship as a safety distance. However, constrained A* algorithm does not take into account the smoothness of the path, which means the path generated by them cannot meet the curvature constraint of ship. Considering the issue of path smooth, finite angle A* algorithm [17] explored more candidate nodes in node expansion process, which provided more abundant turning points for unmanned surface vehicles. In order to obtain a shorter and smoother path, the pre-processing stage and the post-processing stage were introduced into work [18] to reduce the redundant nodes. Meanwhile, the cost of the path generated by work [18] was optimized by reducing the local path cost. Smoothed A* algorithm [19] generated the path without 'jags' and redundant waypoints by adopting cubic spline interpolation and three path smoothers. Despite, the node expansion efficiency of the above-mentioned algorithms is low due to the node expansion process of them are similar with the conventional A* algorithm. Furthermore, additional smoothing optimization process leads to longer running time.

In order to improve the efficiency of node expansion and the convergence speed of the algorithm, researchers altered the node expansion process. By introducing difference engine into the node expansion process, work [20] proposed an improved differential A* algorithm for manipulator. Compared with conventional A* algorithm the reconfigurability and real-time capability of differential A* algorithm was superior. Word [21] proposed a multi-direction A* algorithm for ship which could generate the path similar with the real cases. Experimental result showed that multi-direction A* algorithm could enhance path safety to a large degree and improve efficiency in node expansion process by adding the penalty model into the conventional A* algorithm. Similarly, using a modified node evaluation function, the path created by light-assisted A* algorithm [22] could bypass the obstacles towards the goal in complex-shaped obstacles environment. Due to the efficient nodes expansion process, the convergence rate of light-assisted A* algorithm was improved. However above algorithms are barely considering the physical constraint of the machine which is not suitable for the actual motion.

Aimed at the problem of large-scale UAV path planning in complex environment, Bi-directional adaptive A* algorithm is proposed in this paper. In order to solve the defect of the inefficient node expansion for A* algorithm, directional search strategy is introduced into the proposed algorithm to generate the path which meet the constraint of the UAV attitude angle. Adaptive step strategy adjusts dynamically the step of A* algorithm by estimating the environment around the current node to improve the convergence rate. According to the flight demands of the UAV [23], the evaluation function is modified to generate the path which is far from obstacles and has the safe altitude. Adaptive weight strategy is used to balance the convergence rate and the path quality during the planning. Furthermore, the rewiring process is adopted to reduce the redundant nodes and the path cost. Comprehensively, comparing with the work [19] Bi-directional adaptive A* algorithm can converge to the optimal path faster.

Nomenclature	
$P(n) = (x_n, y_n, z_n)$	Waypoint coordinate
θ, ϕ	Pitch angle, yaw angle
ϕ_{max}	Maximum yaw angle
θ_{max}	Maximum pitch angle
$T^r = (x_0^r, y_0^r, z_0^r, R^r)$	The rth radar.
$f(P(n))$	Total cost
g(P(n)), $h(P(n))$	A* algorithm parameters
$P_{\text{init}}, P_{\text{goal}}$	Initial node and goal node
D_{eff}	effective radius of the obstacle
L, L_{max}, L_{min}	Step of the algorithm
A_a^*, A_b^*	The path generated by algorithm
S	Space environment
S_{obs}	Obstacle area
S_{free}	Obstacle-free area

The main contributions of the paper are as follows: Firstly, several improvements for A* algorithm, namely directional search strategy, adaptive step strategy, adaptive weight strategy, bi-directional search strategy, rewiring process, are proposed and analyzed in this paper. Secondly, compared with conventional A* algorithm and Smoothed A* algorithm, Bi-directional adaptive A* algorithm has the better performance in convergence speed and solution optimality. Thirdly, Bi-directional adaptive A* algorithm can plan a collision-free path, which meets UAV attitude angle and altitude constraints.

This paper is divided into following sections: Section II introduces the UAV physical constraint, and environmental representation. In Section III, the improvements and the procedure of Bi-directional adaptive A* algorithm are given. Section IV follows up with theoretical analysis of our proposed algorithm. Simulation results and comparison under different scenarios are introduced in Section V. Finally, conclusions are drawn in Section VI.

II. PROBLEM DEFINITION

A. PHYSICAL CONSTRAINTS

In order to obtain a feasible path, the following physical constraints need to be satisfied when UAV is in operation.

1) THE CONSTRAIN OF MAXIMUM YAW ANGLE

The maximum yaw angle is used to limit the turning angle of the UAV. Let $P(n) = (x_n, y_n, z_n)$ denote waypoint coordinate. Let $p_n = (x_n - x_{n-1}, y_n - y_{n-1}, z_n - z_{n-1})$ denote the path vector between waypoints $P(n) = (x_n, y_n, z_n)$ and $P(n-1) = (x_{n-1}, y_{n-1}, z_{n-1})$. Maximum yaw angle is represented by ϕ_{max}. Yaw angle of UAV should meet the following condition,

$$cos\phi_{max} \leq \frac{p_n^T p_{n+1}}{|p_n| \, |p_{n+1}|} \tag{1}$$

2) THE CONSTRAIN OF MAXIMUM PITCH ANGLE

The maximum pitch angle is used to restrict the climb/dive angle of the UAV. Let θ_{max} denote maximum pitch angle. Pitch angle of the UAV should meet the following condition,

$$tan\theta_{max} \geq \frac{|Z_n - Z_{n-1}|}{|p_n|} \tag{2}$$

B. ENVIRONMENTAL MODEL

In military applications, UAV not only needs to avoid the threat of the mountain, but also needs to stay away from radar, thus radar threat as obstacle is introduced into this paper. Let $T^r = (x_0^r, y_0^r, z_0^r, R^r)$ denote the rth radar. As an omnidirectional detection device, the mathematical model of radar is simplified to the follow,

$$(x - x_0^r)^2 + (y - y_0^r)^2 + (z - z_0^r)^2 = (R^r)^2 \tag{3}$$

where (x_0^r, y_0^r, z_0^r) is the center location coordinate of the rth radar, R^r is the effective radius of the rth radar.

It is necessary for UAV path planning algorithm to consider the effects of physical constraint and the threat to improve stability and safety of the UAV, thus distance cost, obstacle threat, altitude, and so on, are integrated into Bi-directional adaptive A* algorithm.

III. BI-DIRECTIONAL ADAPTIVE A* ALGORITHM

A. A* ALGORITHM

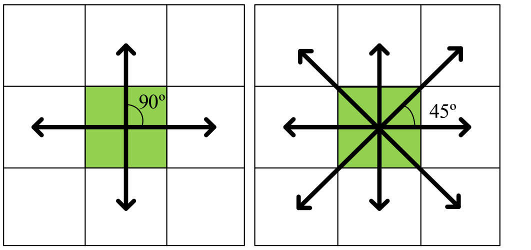
As a heuristic method, A* algorithm based on the evaluation function seeks the shortest path consisted by a finite set of waypoints $P_{\text{init}}, P(1), \cdots, P_{\text{goal}}$ between the initial node P_{init} and the goal node P_{goal}. During each iteration, A* algorithm calculates total cost $f(P(n))$ for each node $P(n)$, where $f(P(n))$ is calculated by the follow equation:

$$f(P(n)) = g(P(n)) + h(P(n)) \tag{4}$$

where $g(P(n))$ denote the distance cost from the current node $P(n)$ to the initial node P_{init}, $h(P(n))$ is the estimated cost of the path from the current node $P(n)$ to the goal node P_{goal}. In order to obtain the shortest path between the initial node and the goal node, A* algorithm seeks the node owning the minimum $f(P(n))$ value at each iteration process. And that process will continue till the algorithm returns an available path.

B. DIRECTIONAL SEARCH STRATEGYA

Based on the difference of candidate nodes number in the exploration process, conventional A* algorithm based on the grid method can be divided into 4-direction A* algorithm, 8-direction A* algorithm as follow in Fig.1. As the number of candidate nodes increases, the performance of multi-direction A* algorithm can be improved in the yaw angle. However, due to more candidate nodes are available, multi-direction A* algorithm will inevitably increase computation and sacrifice memory.



(a) 4-direction A* algorithm (b) 8-direction A* algorithm

FIGURE 1. Multi-direction A* algorithm.

Multi-direction A* algorithm is advantageous to improve the exploration ability; however, the invalid candidate nodes would increase the computational time. Therefore, a more efficient extension method needs to be introduced to reduce the computation. Combining the direction vector from the current node to the goal node in obstacle-free space with constrain of pitch angle θ and yaw angle ϕ of the UAV, the candidate nodes of the current node can be confirmed by directional search strategy.

It can be seen from Fig.2, the candidate nodes obtained by directional search strategy are distributed on the arc surface which satisfy the constrain of UAV pitch angle θ and yaw angle ϕ. Directional search strategy not only guarantees the exploration ability of the proposed algorithm, but improves the efficiency of seeking candidate nodes. Due to the constrains of UAV pitch angle θ and yaw angle ϕ are considered in directional search strategy, the extra smoothing process is not necessary for the path generated by it.

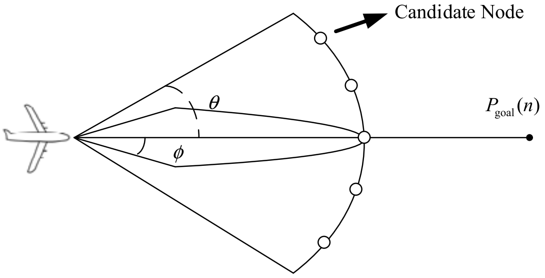


FIGURE 2. Directional search strategy.

C. ADAPTIVE STEP STRATEGY

Conventional A* algorithm connects the current node $P(n)$ and the child node $P_{\text{child}}(n)$ with fixed short step, which could delay the convergence of it in obstacle-free. In order to overcome this defect, an adaptive step strategy is proposed in this paper to adaptively adjust step while the current node is under the different environment, such as obstruction area and safe area.

The calculation equation of adaptive step strategy is shown below:

$$L = \begin{cases} L_{\max} - \frac{D_{\text{eff}}}{2*D} & D \geq D_{\text{eff}} \\ L_{\min} + L_{\min}*\frac{D}{D_{\text{eff}}} & D < D_{\text{eff}} \end{cases} \tag{5}$$

where L is the step, $L_{\max}$ and $L_{\min}$ represent the maximum step and the minimum step, respectively, let D denote the distance between the current node to its nearest obstacle. D_{eff} is the effective radius of the obstruction. As shown in Fig.3, when the UAV is far from the obstacle, the UAV is considered as in the safe area ($D \geq D_{\text{eff}}$). In this case, the large step is allowed to accelerate convergence. As shown in Fig.4, if the UAV is estimated in the area of obstruction ($D < D_{\text{eff}}$), the small step size is used to guarantee the granularity of the path.

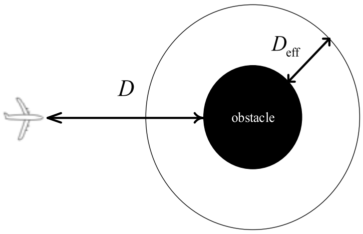


FIGURE 3. UAV in the safe area.

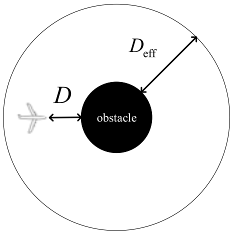


FIGURE 4. UAV in the obstruction area.

D. EVALUATION FUNCTION

Considering the special constraint of the UAV, such as distance cost, collision-free path and safe flight altitude, $g(P(n))$ is modified to the following:

$$g(P(n)) = \omega_1 g_{\text{obs}}(P(n)) + \omega_2 g_{\text{alt}}(P(n)) + \omega_3 g_{\text{cost}}(P(n)) \tag{6}$$

where $g_{\text{obs}}(P(n))$ represent the cost from the current position $P(n)$ to its nearest obstacle, $g_{\text{alt}}(P(n))$ is the flight altitude cost from the current position $P(n)$ to the reference altitude, $g_{\text{cost}}(P(n))$ denote the Euclidean distance cost from the current position $P(n)$ to the initial node P_{init}. ω_1, ω_2, and ω_3 are the weight of each parameter, respectively.

Different from the conventional A* algorithm, heuristic function $h(P(n))$ is calculated in terms of Manhattan distance [24] instead of Euclidean metric in this paper, due to the optimization effect is different when the adopted geometric measurement of heuristic function $h(P(n))$ is different. It can be seen from Fig.5(a), heuristic function $h(P(n))$ based on Euclidean distance will explore more space, which sacrifices the memory. Meanwhile the expensive square root in Euclidean metric increases the computation. On the contrary, heuristic function $h(P(n))$ based on Manhattan distance conducts more accurate space exploration with less redundant nodes. Manhattan distance adopted by this paper guarantees the efficiency of our proposed algorithm and the path quality.

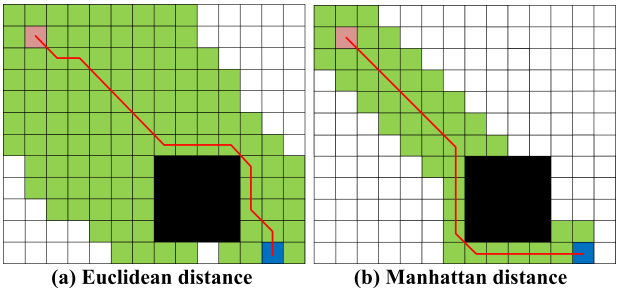


FIGURE 5. The path under different heuristic function.

$$h(P(n)) = \left|P_x(n) - P_{\text{goal\_x}}\right| + \left|P_y(n) - P_{\text{goal\_y}}\right| + \left|P_z(n) - P_{\text{goal\_z}}\right| \tag{7}$$

where $(P_x(n), P_y(n), P_z(n))$ denote the coordinate of node $P(n)$, $(P_{\text{goal\_x}}, P_{\text{goal\_y}}, P_{\text{goal\_z}})$ is the coordinate of goal node P_{goal}.

E. ADAPTIVE WEIGHT STRATEGY

When the weight of $h(P(n))$ is smaller, A* algorithm approaches to Dijkstra algorithm, which seeks the shortest path from the initial node P_{init} to the goal node P_{goal} by conduct abundant space exploration with slower convergence speed. When the weight of $g(P(n))$ is smaller, A* algorithm translates into Best-First-Search algorithm (BFS), finding a path quickly, however this path is not the optimal for UAV.

Thus, A* algorithm based on the Dijkstra algorithm and the BFS algorithm can balance the rate of convergence and the quality of the path by adjusting the weight of g(P(n)) and $h(P(n))$ dynamically. The evaluation function of our proposed algorithm is calculated by the following:

$$f(P(n)) = \omega_g g(P(n)) + \omega_h h(P(n)) \tag{8}$$

where ω_g and ω_h are the weight factor of $g(P(n))$ and $h(P(n))$, respectively.

The calculation equations for ω_g and ω_h are as follow:

$$\omega_h = \begin{cases} \omega_{\max} & D \geq D_{\text{eff}} \\ \omega_{\max} - (\omega_{\max} - \omega_{\min}) * (1 - \dfrac{D}{D_{\text{eff}}}) & D < D_{\text{eff}} \end{cases} \tag{9}$$

$$\omega_g = 1 - \omega_h \tag{10}$$

where $\omega_{\max}$ and $\omega_{\min}$ are respectively the maximum and minimum preset value for ω_h.

It can be seen from Equ. (9), the value of ω_h will be set as the maximum value to accelerate the rate of convergence in the obstacle-free area ($D \geq D_{\text{eff}}$). Meanwhile, according to Equ. (10), ω_g will be set as the minimum value to avoid the useless exploration. When UAV drive into the obstacles area ($D < D_{\text{eff}}$), accurate space exploration is needed to generate a high-quality path. Therefore, the value of ω_g is adaptively increased to find an optimal path in obstacles area.

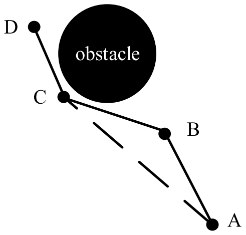


FIGURE 6. Schematic diagram of rewiring process.

F. BI-DIRECTIONAL SEARCH STRATEGY

In order to improve the exploration ability of the A* algorithm, bi-directional search strategy is introduced into our proposed algorithm which performs both forward and reverse searches at the same time. When the forward search path A_a^* rooted at the initial node P_{init} and the reverse search path A_b^* initialized at the goal node P_{goal} explore the same child node, algorithm returns a feasible path.

G. REWIRING PROCESS

The path generated by the above path exploration strategy consists of a number of short line segments linking a mass of path nodes, might including redundant nodes. In order to reduce the path length and the useless turning nodes, the rewiring process is introduced in this paper to delete the redundant nodes.

For each optimization process, rewiring process checks all path nodes successively in a groups of three continuous path nodes and above three continuous nodes are named as follows, (1) let node A denote the current checking node; (2) let node B denote the child node of the current checking node A in path set; (3) let node C denote the child node of the node B. If nodes A, B, and C satisfy Equ. (11) and Equ. (12), the original path $\overline{\mathrm{A-B-C}}$ will turns into $\overline{\mathrm{A-C}}$, and the redundant node B will be removed. Fig.6 illustrates the principle of the rewiring process.

$$\overline{A-C} \cap S_{\text{obs}} = \emptyset \tag{11}$$

$$\|\text{C-A}\| \leq \|\text{C-B}\| + \|\text{B-A}\| \tag{12}$$

where S_{obs} denote the obstacle area, $\|\bullet\|$ is the Euclidean norm, $\|\text{C-A}\|$ represent the Euclidean distance between C to A. $\|\text{C-B}\|$ and $\|\text{B-A}\|$ are the same definition as above.

Fig.6 illustrate how this method works, where $\overline{\mathrm{A-B-C-D}}$ is the path to be optimized, $\overline{\mathrm{A-C-D}}$ is the optimized path. Clearly, invalid turning node B increases path length and the instability of the UAV, thus, rewiring process is proposed to optimize the path in distance cost and the angular variation of the UAV.

H. ALGORITHM PROCESS

The detailed steps of Bi-directional adaptive A* algorithm are given in this section.

Step 1: Initialize three-dimensional solution space. Construct radars by Equ. (3) and set the initial position and the goal position for UAV. Set the algorithm parameters. Initialize close list and open list.

TABLE 1. Pseudo code of the bi-directional adaptive A* algorithm.

```
Initialize three-dimensional environment model;
Set the location of radars by (3);
Set the initial position and the goal position;
Set the algorithm parameters $\theta_{\max}$, $\emptyset_{\max}$, $\omega_{\max}$, $\omega_{\min}$, $\omega_1$, $\omega_2$, $\omega_3$, $L_{\max}$, $L_{\min}$;
Initialize close list Close_a={ }, Close_b={ };
Initialize open list Open_a={$P_{\text{init}}$}, Open_b={$P_{\text{goal}}$};
while Open_a≠∅ or Open_b≠∅ do
  Current node of $A_a^*$ $P_a(n) := \min\{f(P_a) | P_a \in$ Open_a$\}$, where $f(P_a)$ is calculated by (8)
  Current node of $A_b^*$ $P_b(n) := \min\{f(P_b) | P_b \in$ Open_b$\}$, where $f(P_b)$ is calculated by (8)
  if $P_a(n)$=$P_b(n)$ break
  Remove $P_a(n)$ from Open_a and add it to Close_a
  Remove $P_b(n)$ from Open_b and add it to Close_b
  Adopt the directional search strategy and adaptive step strategy to obtain the set of successor node $S_a$, $S_b$ of the current nodes $P_a(n)$, $P_b(n)$
  for each successor node $P_a(n+1) \in S_a$ ($P_b(n+1) \in S_b$) of $P_a(n)$ ($P_b(n)$) do
    if $P_a(n+1) \in$ Close_a ($P_b(n+1) \in$ Close_b) continue
    if $P_a(n+1) \notin$ Open_a ($P_b(n+1) \notin$ Open_b)
      Add $P_a(n+1)$ into Open_a (add $P_b(n+1)$ into Open_b)
      Set current node as father node of successor node
      Calculate $g(P_a(n+1))$, $h(P_a(n+1))$, $f(P_a(n+1))$ by (6), (7), (8)
      Calculate $g(P_b(n+1))$, $h(P_b(n+1))$, $f(P_b(n+1))$ by (6), (7), (8)
    if $P_a(n+1) \in$ Open_a ($P_b(n+1) \in$ Open_b)
      $g(P_a(n+1))' = g(P_a(n)) + \|P_a(n+1) - P_a(n)\|$
      $g(P_b(n+1))' = g(P_b(n)) + \|P_b(n+1) - P_b(n)\|$
      if $g(P_a(n+1))' \geq g(P_a(n+1))$ ($g(P_b(n+1))' \geq g(P_b(n+1))$) continue
      else
        Set current node as father node of it
        $g(P_a(n+1)) = g(P_a(n+1))'$
        $g(P_b(n+1)) = g(P_b(n+1))'$
        Calculate $f(P_a(n+1))(f(P_b(n+1)))$ by (8)
Rewiring the path by (11), (12)
```

Step 2: The node in the open list with the minimum value $f(\cdot)$ is marked as the current node by the calculation of Equ. (8). If current nodes of A_a^* and A_b^* are accessible for each other, then return a feasible path and go to Step 6.

Step 3: Remove current node from open list and add it to close list. Calculate successor node by the directional search strategy and adaptive step strategy.

Step 4: If successor node is absent in the open list, add it into the open list. Set the current node as its father node and calculate $\{g(P(n)), h(P(n)), f(P(n))\}$ by Eqs. (6)~(8), respectively. Return to Step 2.

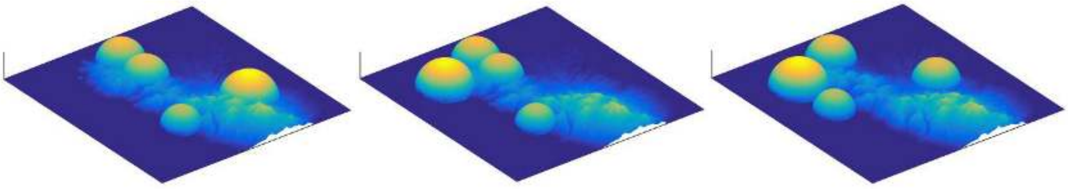
Step 5: If successor node is present in the open list and the cost is less than the current node cost, update cost of successor node. Return to Step 2.

Step 6: The optimal path is returned by the rewiring process.

IV. ANALYSIS

A. PROBABILISTIC COMPLETENESS

Let $D_k^a(p)$ denote the random variable of the minimum distance from node $p \in S_{\text{free}}$ to the latest node of the first path A_a^*,



(a) Map1 (b) Map2 (c) Map3

FIGURE 7. Simulation map.

where A_a^* is defined as above. $d_k^a(p)$ is the value of the random variable $D_k^a(p)$, where k is the number of iterations, S_{free} is the obstacle-free space. Similarly, let $D_k^b(p)$ denote the random variable of the minimum distance from node $p \in S_{\text{free}}$ to the latest node of the second path A_b^* and A_b^* is defined as above. $d_k^b(p)$ is the value of the random variable $D_k^b(p)$. $D_k(p_a, p_b)$ is the random variable of the minimum distance from path A_a^* to path A_b^*, where p_a and p_b is the latest nodes of path A_a^* and path A_b^*, respectively. $d_k(p_a, p_b)$ is the value of the random variable $D_k(p_a, p_b)$.

Theorem: Bi-directional adaptive A* algorithm is a probabilistically complete algorithm. For any space S, when the number of iterations approach infinity, the distance between p_a and p_b is less than a constant ε. Thus, the probability of searching a path goes to one.

$$\lim_{k\to\infty} P\left(d_k\left(p_a, p_b\right) < \varepsilon \,|\, p_a, p_b \in S_{\text{free}}\right) = 1 \tag{13}$$

Proof: Let $O_a^k(p_a)$ denote the closed ball of radius $\varepsilon > 0$ centered at $p_a \in A_a^*$ in kth iteration. Similarly, $O_b^k(p_b)$ represent the closed ball of radius $\varepsilon > 0$ centered at $p_b \in A_b^*$ in kth iteration.

$$O_a^k\left(p_a\right) : O_{a,\varepsilon}^k := \{p \in S \,|\, \|p - p_a\| \le \varepsilon\} \tag{14}$$
$$O_b^k\left(p_b\right) : O_{b,\varepsilon}^k := \{p \in S \,|\, \|p - p_b\| \le \varepsilon\} \tag{15}$$

Let $\dot{O}_a^k\left(p_a\right)$ denote the intersection of the closed ball $O_a^k\left(p_a\right)$ and the obstacle-free space S_{free}. $\dot{O}_b^k\left(p_b\right)$ is defined as same as the $\dot{O}_a^k\left(p_a\right)$.

$$\dot{O}_a^k(p_a) = O_a^k(p_a) \cap S_{\text{free}} \tag{16}$$
$$\dot{O}_b^k(p_b) = O_b^k(p_b) \cap S_{\text{free}} \tag{17}$$

For the kth iteration, it is assumed that the latest node of the path A_a^* is outside the ball $\dot{O}_b^k(p_b)$.

$$\dot{O}_b^k(p_b) \cap A_a^* = \emptyset \tag{18}$$

Similarly, let us assume that the second path A_b^* and the ball $\dot{O}_a^k(p_a)$ have no intersection,

$$\dot{O}_a^k(p_a) \cap A_b^* = \emptyset \tag{19}$$

For the next iteration, due to the node exploration process adopts the directional search strategy, path A_a^* is expanded toward the direction of the latest node of the path A_b^*. The mean of $D_{k+1}^a(p_b)$ is given by following,

$$E(D_{k+1}^a(p_b)) < E(D_k^a(p_b)) \tag{20}$$

Similarly, the directional search strategy act on the second path A_b^*. The mean of $D_{k+1}^b(p_a)$ is expressed below,

$$E(D_{k+1}^b(p_a)) < E(D_k^b(p_a)) \tag{21}$$

One can see from Equ. (20) and Equ. (21) that, for the $k+1$th iteration, the distance between the path A_a^* and the path A_b^* will decreases.

$$d_{k+1}(p_a, p_b) < d_k(p_a, p_b) \tag{22}$$

Thus, as the number of iterations k goes to infinity, the probability of searching a path goes to one.

$$\lim_{k\to\infty} P\left(d_k\left(p_a, p_b\right) < \varepsilon \,|\, p_a, p_b \in S_{\text{free}}\right) = 1 \tag{23}$$

V. SIMULATION RESULTS

In order to evaluate the performance of our proposed algorithm, several comparisons are carried out among Bi-directional adaptive A* algorithm (BAA*), Smoothed A* algorithm (SA*), and Conventional A* algorithm (CA*) in this section. Three different maps in Fig.7 are used to test the performance of BAA*, where radars are added as the threaten area in the maps. The characteristic parameters of BAA* are shown in Table 2. The parameters of the threaten area are given in Table 3.

TABLE 2. Characteristic parameters of BAA*.

Parameters	Symbol	Data
Maximum pitch angle	$\theta_{\max}$	$\pi/4$
Maximum yaw angle	$\phi_{\max}$	$\pi/4$
Weight of maximum	$\omega_{\max}$	1
Weight of minimum	$\omega_{\min}$	0.2
Weight of obstacle threat	ω_1	0.3
Weight of the altitude	ω_2	0.2
Weight of the cost	ω_3	0.5
Maximum step	$L_{\max}$	1
Minimum step	$L_{\min}$	0.3

TABLE 3. Parameters of the threaten area.

Environment	Parameter
Map1	$T^1 = (200{,}150{,}0{,}4), T^2 = (300{,}320{,}0{,}5),$ $T^3 = (150{,}320{,}0{,}3.5), T^4 = (200{,}80{,}0{,}4.2).$
Map2	$T^1 = (200{,}150{,}0{,}4), T^2 = (100{,}120{,}0{,}5),$ $T^3 = (150{,}320{,}0{,}3.5), T^4 = (200{,}80{,}0{,}4.2).$
Map3	$T^1 = (300{,}250{,}0{,}4), T^2 = (100{,}120{,}0{,}5),$ $T^3 = (100{,}220{,}0{,}3.5), T^4 = (200{,}80{,}0{,}4.2).$

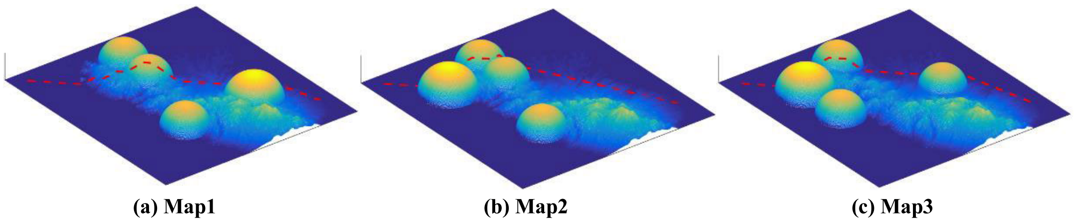

(a) Map1 (b) Map2 (c) Map3

FIGURE 8. Simulation results of CA*.

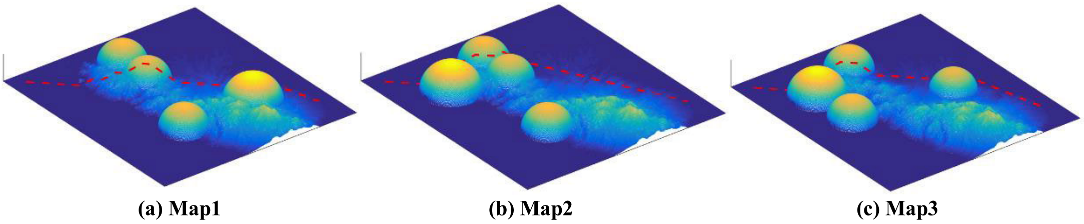

(a) Map1 (b) Map2 (c) Map3

FIGURE 9. Simulation results of SA*.

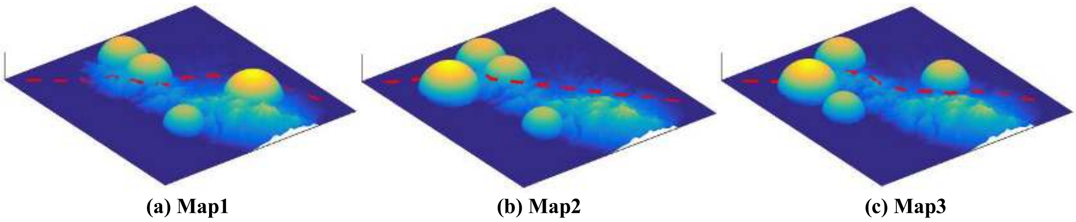

(a) Map1 (b) Map2 (c) Map3

FIGURE 10. Simulation results of BAA*.

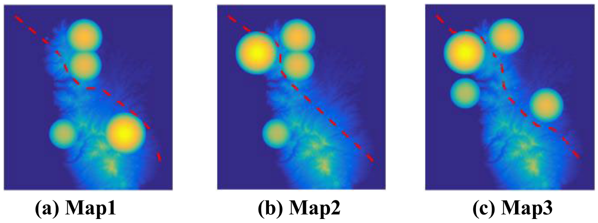

(a) Map1 (b) Map2 (c) Map3

FIGURE 11. Simulation results of BAA* in vertical view.

TABLE 4. Simulation result data.

Environment	Algorithm	Path Cost	Nodes	Time(s)
Map1	CA*	51.455km	44	50.308
	SA*	49.624km	22	55. 233
	BAA*	46.223km	14	20.376
Map2	CA*	59.094km	40	50.854
	SA*	52.796km	16	57.154
	BAA*	49.560km	14	21.742
Map3	CA*	55.351km	42	53.541
	SA*	54.003km	22	57.051
	BAA*	49.056km	22	22.695

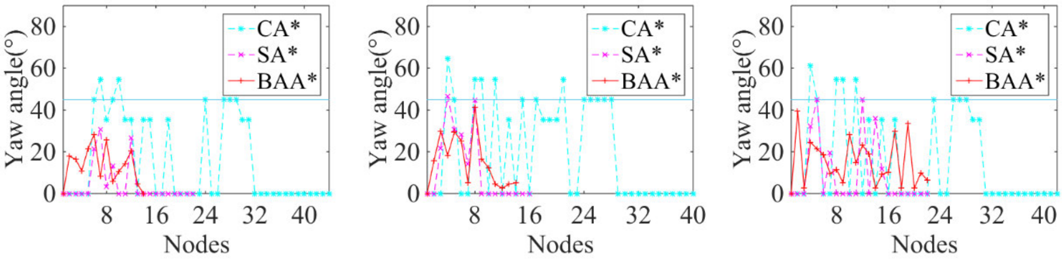
The simulation results of above algorithm running in three different maps are shown in Fig.8-11, where the simulation results of BAA* are shown in different visual angles. The data of simulation results are illustrated in Table 4. Furthermore, the data of attitude angle for these different A* algorithm are given and analyzed below, meanwhile the altitude comparison of them are also carried out. The related results of attitude angle and altitude are shown in Tables 5 and Fig.12-16.

A. OPTIMALITY COMPARISON

It can be seen from Table 4, the algorithm optimality includes the path cost, computational time and the number of the search grids, where the distance cost represents the basic performance of the algorithm, computational time and the number of the search grids represent the computing power of the algorithm.

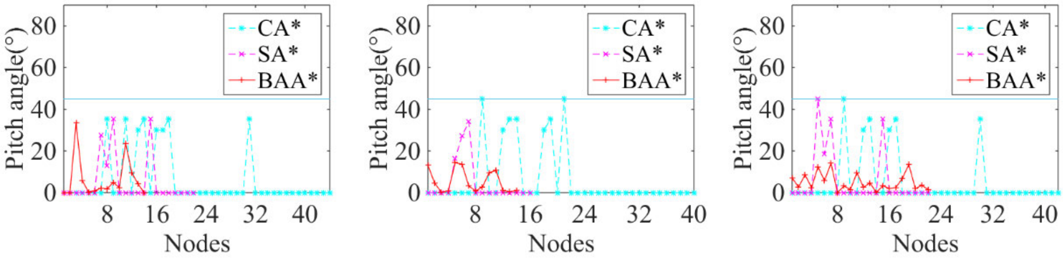
The simulation results of CA* in different environments are shown in Fig.8. Based on the 8-direction grid search strategy, CA* explores a mass of candidate nodes (Nodes = 44, 40, 42), which sacrifices the memory and decreases the convergence speed (Time = 50.308, 50.854, 53.541). Meanwhile, the path generated by CA* climbs over the obstacles to avoid the collision. Due to the climb process of the UAV, the path cost can be increased (Path Cost = 51.455, 59.094, 55.351) which lead to the increase in fuel consumption of UAV.

As seen from Fig.9, comparing with the CA*, SA* can generate smoother path with less jags which improve the



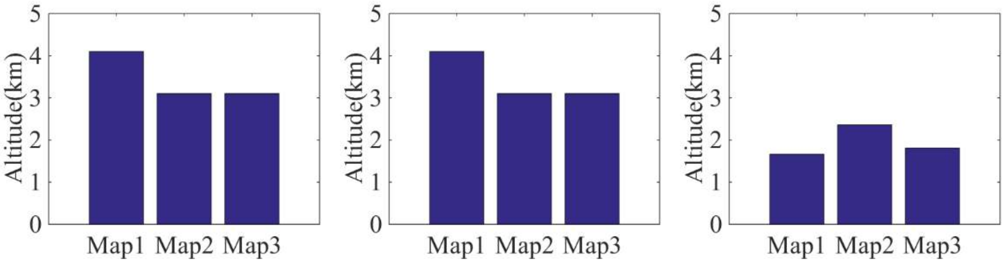
(a) Map1 (b) Map2 (c) Map3

FIGURE 12. Yaw angle change of UAV.



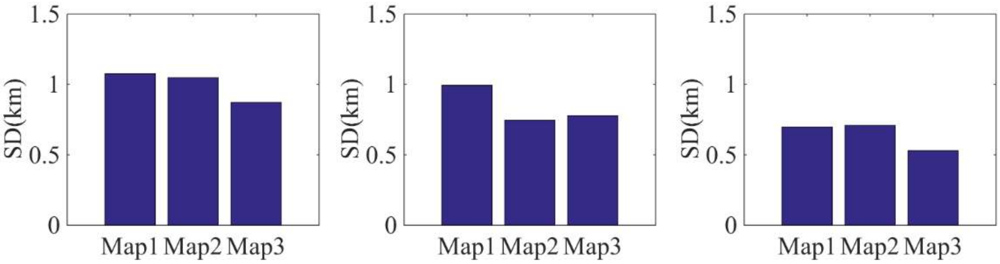
(a) Map1 (b) Map2 (c) Map3

FIGURE 13. Pitch angle change of UAV.



(a) CA* (b) SA* (c) BAA*

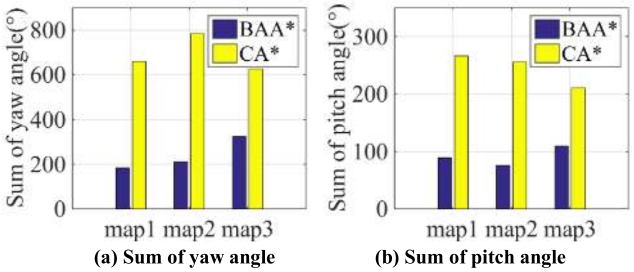
FIGURE 15. The maximum flight altitude of three different algorithms in different maps.



(a) CA* (b) SA* (c) BAA*

FIGURE 16. The flight altitude standard deviation of three algorithms in different maps.

stability of the UAV. It can be seen from Table 4, by using refining path smoother, the redundant nodes are removed (Nodes = 22, 16, 22) which reduces the path cost (Path Cost = 49.624, 52.796, 54.003) and the demand of the memory. However, the extra smoothing processes of SA* increases the computational time which reduces the real-time performance of the SA* (Time = 55. 233, 57.154, 57.051).



(a) Sum of yaw angle (b) Sum of pitch angle

FIGURE 14. Comparison of angle sum.

TABLE 5. Comparison of attitude angle sum of the different algorithms.

Environment	Algorithm	yaw angle sum	pitch angle sum
Map1	BAA*	183.851°	89.063°
	CA*	661.586°	266.322°
Map2	BAA*	210.352°	75.966°
	CA*	784.532°	255.793°
Map3	BAA*	324.074°	109.381°
	CA*	626.843°	210.793°

The results of BAA* are shown in Fig.10-11 in different perspective. Comparing with SA*, the extra smoothing process is not necessary for BAA* due to the directional search strategy based the attitude angle of the UAV. By adopting the adaptive step strategy and the adaptive weight strategy, BAA* extends quickly in the obstacle-free area to accelerate the convergence rate of algorithm (Time = 20.376, 21.742, 22.695) and accurately explores in the obstacles area to obtain the shortest path. Finally, the rewiring process is used to reduce the redundant nodes (Nodes = 14, 14, 22) and the path cost is decreased further (Path Cost = 46.223, 49.560, 49.056).

B. ATTITUDE ANGLE COMPARISON

As the change of attitude angle is closely related to the UAV stability and is a main index of path quality for UAV, attitude angle comparisons among different A* algorithm are given and analyzed with the combination of Fig.12-14 and tables 5.

It can be seen from Fig.12-13, since only the basic requirements of UAV path planning, such as obstacle avoidance and distance cost, are considered, the yaw angle of CA* inevitably exceeds the constraint when UAV avoids obstacles. The larger attitude angle change leads to the instability and insecurity of UAV. Comparing with CA*, the attitude angle of the path created by SA* is optimized by introduced three individual path smoothers. But there are some unbearable jags in the path while the UAV passes by the threat.

Thanks to the introduction of attitude angle constraint during the exploration process, the path generated by BAA* satisfies the angle constraint of UAV, which guarantees the stability during flight. Let γ_{ya} and γ_{pa} represent the sum of UAV yaw and pitch angle change, respectively. γ_{ya} and γ_{pa} are calculated by the follow,

$$\gamma_{ya} = \sum\nolimits_{v=1}^{V} \gamma_{y\_a}(v) \tag{24}$$

$$\gamma_{pa} = \sum\nolimits_{v=1}^{V} \gamma_{p\_a}(v) \tag{25}$$

where V denote the number of attitude angle change, γ_{y_a} and γ_{p_a} denote the set of yaw and pitch angle, respectively.

It can be seen from Table 5 and Fig.14, compared with CA*, the sum changes of yaw angle are reduced by 72.210%, 73.187%, and 48.300% while BAA* is run in map1, map2, and map3, respectively. For the pitch angle, compared with CA*, the path generated by BAA* in three different maps reduce it by 66.558%, 70.301%, and 48.109%, respectively.

C. ALTITUDE COMPARISON

It can be seen from Fig.15, due to altitude constraint is not introduced in CA*, the climbing process is used to avoid obstacles. The maximum altitude of CA* in map1, map2, and map3 reach to 4.1km, 3.1km, and 3.1km, respectively. Further, flight altitude standard deviations (SD) is shown in Fig.16, which reflects the stationarity of the UAV during the flying process. The flight altitude standard deviations of CA* is 1.076km, 1.048km, and 0.873km in three different maps respectively, which means the paths generated by CA* are unstable. Meanwhile, too much energy is wasted during the climbing process. Similarly, for SA*, the maximum altitude in map1, map2, and map3 is 4.1km, 3.1km, and 3.1km, respectively. However, due to the cubic spline interpolation function is introduces into SA*, the standard deviations of flight altitude are 0.994km, 0.747km, and 0.779km, which means the stability and smoothness of the path. However, similar to the conventional A*, the node growth process of SA* also ignores the altitude limitation of UAV, resulting in the path generated by SA* has the climbing process to avoid the obstacles.

By introducing the information of the obstacles and the altitude limit into the evaluation function, the path generated by BAA* can bypass the obstacles without the climbing process. As shown in Fig.15, the maximum altitude of the path in map1, map2, and map3 is 1.661km, 2.357km, 1.807km, respectively. And the standard deviations of flight altitude are 0.698km, 0.709km, 0.530km, which guarantees the safety and the stability of the UAV.

VI. CONCLUSION

A novel path planning algorithm based on A* algorithm underling multi-constraints is proposed in this paper namely Bi-directional adaptive A* algorithm. The directional search strategy, adaptive step strategy, modified evaluation function, adaptive weight strategy, bi-directional search strategy, and rewiring process are respectively introduced and analyzed in this paper. The main findings are summarized as follows: (1) Due to the adoption of the directional search strategy based on the attitude angle, the proposed algorithm generates the path meeting the constraint condition of large-scale UAV and increases the rate of convergence comparing with other A* algorithm. (2) Adaptive step strategy and the adaptive weight strategy are used to balance exploration speed and the path planning quality by analyzing the environmental conditions of the current node. (3) The altitude limitation and the obstacles information are introduced into the evaluation function to create the safety path. (4) Rewiring process is used to reduce the redundant nodes which decreases the path cost further. In order to verify the feasibility of Bi-directional adaptive A* algorithm, the proof of probabilistic completeness is given in section IV. The simulation comparison results given in section V indicate that comparing with conventional A* algorithm and smoothed A* algorithm [19], Bi-directional adaptive A* algorithm has superiorities for path planning problem of large-scale UAV in term of the path quality and the rate of convergence.

REFERENCES

[1] S. Mousavi, F. Afghah, J. D. Ashdown, and K. Turck, "Use of a quantum genetic algorithm for coalition formation in large-scale UAV networks," *Ad Hoc Netw.*, vol. 87, pp. 26–36, May 2019.

[2] M. Półka, S. Ptak, and Ł. Kuziora, "The use of UAV's for search and rescue operations," *Procedia Eng.*, vol. 192, pp. 748–752, Jan. 2017.

[3] M. Jaud, P. Letortu, C. Théry, P. Grandjean, S. Costa, O. Maquaire, R. Davidson, and N. L. Dantec, "UAV survey of a coastal cliff face–selection of the best imaging angle," *Measurement*, vol. 139, pp. 10–20, Jun. 2019.

[4] Y. Li, Y. Zhang, and L. Cai, "Optimal location of supplementary node in UAV surveillance system," *J. Netw. Comput. Appl.*, vol. 140, pp. 23–39, Aug. 2019.

[5] Z. Wang and J. Cai, "The path-planning in radioactive environment of nuclear facilities using an improved particle swarm optimization algorithm," *Nucl. Eng. Design*, vol. 326, pp. 79–86, Jan. 2018.

[6] L. Zhu, X. Cheng, and F.-G. Yuan, "A 3D collision avoidance strategy for UAV with physical constraints," *Measurement*, vol. 77, pp. 40–49, Jan. 2016.

[7] L. P. Behnck, D. Doering, C. E. Pereira, and A. Rettberg, "A modified simulated annealing algorithm for SUAVs path planning," *IFAC-PapersOnLine*, vol. 48, no. 10, pp. 63–68, 2015.

[8] H. Ma, S. Shen, M. Yu, Z. Yang, M. Fei, and H. Zhou, "Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey," *Swarm Evol. Comput.*, vol. 44, pp. 365–387, Feb. 2019.

[9] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based UAV path planning," *Knowl.-Based Syst.*, vol. 158, pp. 54–64, Oct. 2018.

[10] S. Perez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz, "Ant colony optimization for multi-UAV minimum time search in uncertain domains," *Appl. Soft Comput.*, vol. 62, pp. 789–806, Jan. 2018.

[11] Y. Liu, X. Zhang, Y. Zhang, and X. Guan, "Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach," *Chin. J. Aeronaut.*, vol. 32, no. 6, pp. 1504–1519, Jun. 2019.

[12] H. Guo, Z. Mao, W. Ding, and P. Liu, "Optimal search path planning for unmanned surface vehicle based on an improved genetic algorithm," *Comput. Electr. Eng.*, vol. 79, Oct. 2019, Art. no. 106467, doi: 10.1016/j.compeleceng.2019.106467.

[13] S. Shao, Y. Peng, C. He, and Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization," *ISA Trans.*, vol. 97, pp. 415–430, Feb. 2020.

[14] P. Pharpatara, B. Herisse, and Y. Bestaoui, "3-D trajectory planning of aerial vehicles using RRT∗," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 3, pp. 1116–1123, May 2017.

[15] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[16] Y. Singh, S. Sharma, R. Sutton, D. Hatton, and A. Khan, "A constrained A∗ approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents," *Ocean Eng.*, vol. 169, pp. 187–201, Dec. 2018.

[17] J.-M. Yang, C.-M. Tseng, and P. S. Tseng, "Path planning on satellite images for unmanned surface vehicles," *Int. J. Nav. Archit. Ocean Eng.*, vol. 7, no. 1, pp. 87–99, Jan. 2015.

[18] B. Fu, L. Chen, Y. Zhou, D. Zheng, Z. Wei, J. Dai, and H. Pan, "An improved A∗ algorithm for the industrial robot path planning with high success rate and short length," *Robot. Auto. Syst.*, vol. 106, pp. 26–37, Aug. 2018.

[19] R. Song, Y. Liu, and R. Bucknall, "Smoothed A∗ algorithm for practical unmanned surface vehicle path planning," *Appl. Ocean Res.*, vol. 83, pp. 9–20, Feb. 2019.

[20] K. I. Trovato and L. Dorst, "Differential A∗," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 6, pp. 1218–1229, Nov. 2002.

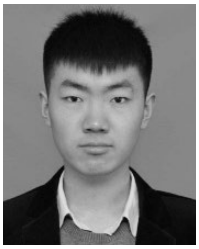
[21] L. Xie, S. Xue, J. Zhang, M. Zhang, W. Tian, and S. Haugen, "A path planning approach based on multi-direction A∗nmmm algorithm for ships navigating within wind farm waters," *Ocean Eng.*, vol. 184, pp. 311–322, Jul. 2019.

[22] M. Hawa, "Light-assisted A∗ path planning," *Eng. Appl. Artif. Intell.*, vol. 26, no. 2, pp. 888–898, Feb. 2013.

[23] M. V. Ramana, S. A. Varma, and M. Kothari, "Motion planning for a fixed-wing UAV in urban environments," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 419–424, 2016.

[24] G. J. Tielen, T. Lulek, M. R. M. J. Traa, M. Kuzma, and W. J. Caspers, "The role of the manhattan distance in antiferromagnetic ordering," *Phys. A, Stat. Mech. Appl.*, vol. 246, nos. 1–2, pp. 199–220, Nov. 1997.

XUELI WU received the B.S. and M.S. degrees from Yanshan University, in 1983 and 1988, respectively, and the Ph.D. degree from the Huazhong University of Science and Technology, in 2005. He is currently a Professor and the Dean of electrical engineering with the Hebei University of Science and Technology, China. His research interests include intelligent control and UAV risk avoidance.



LEI XU received the B.S. degree from the Hebei University of Engineering, China, in 2017, where he is currently pursuing the M.S. degree in electrical engineering with the Department of Electrical Engineering. His research interest includes path planning of UAV.

RAN ZHEN received the B.S. degree from the North China University of Science and Technology University, in 1994, and the M.S. degree from the Huazhong University of Science and Technology, in 2004. She is currently a Professor of electrical engineering with the Hebei University of Science and Technology, China. Her research interest includes intelligent control for nonlinear systems.

XIAOJING WU received the B.S. and M.S. degrees from Yanshan University, China, in 2006 and 2009, respectively, and the Ph.D. degree in control theory and control engineering from Yanshan University. She is currently an Associate Professor with the Hebei University of Science and Technology, China. Her research interests include nonlinear control, UAV control, and path planning.