

Received March 25, 2021, accepted April 8, 2021, date of publication April 12, 2021, date of current version April 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3072796

# Path Planning for Unmanned Aerial Vehicle Using a Mix-Strategy-Based Gravitational Search Algorithm

HONGGEN XU<sup>1,2</sup>, SHUAI JIANG<sup>3</sup>, AND AIZHU ZHANG<sup>4</sup> , (Member, IEEE)

<sup>1</sup>Wuhan Center of China Geological Survey (Central South China Innovation Center for Geosciences), China Geological Survey, Wuhan 430205, China

<sup>2</sup>Changsha Center of Natural Resources Comprehensive Survey, China Geological Survey, Changsha 410699, China

<sup>3</sup>Shengli Xinda New Material Company Ltd., Dongying 257000, China

<sup>4</sup>College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao 266580, China

Corresponding author: Aizhu Zhang (zhangaizhu789@163.com)

This work was supported in part by the Project of Civil Space Technology Pre-research of the 13th Five Year Plan under Grant D040104, in part by the National Natural Science Foundation of China under Grant 41871270, Grant 41971292, and Grant 41801275, and in part by the National Key Research and Development Program under Grant 2019YFE0126700.

**ABSTRACT** Path planning is a global optimization problem aims to program the optimal flight path for Unmanned Aerial Vehicle (UAV) that has short length and suffers from low threat. In this paper, we present a Mixed-Strategy based Gravitational Search Algorithm (MSGSA) for the path planning. In MSGSA, an adaptive adjustment strategy for the gravitational constant attenuation factor  $\alpha$  is presented firstly, in which the value of  $\alpha$  is adjusted based on the evolutionary state of the particles. This helps to adaptively balance the exploration and exploitation of the algorithm. In addition, to further alleviate the premature convergence problem, a Cauchy mutation strategy is developed for MSGSA. In this strategy, only when the global best particle cannot be further improved for several times the mutation is executed. In the MSGSA based path planning procedure, we construct an objective function using the flight length cost, threat area cost, and turning angle constraint to decrease the flight risk and obtain the smoother path. For performance evaluation, the MSGSA is applied to two typical simulated flight missions with complex flight environments, including user-defined forbidden flying areas, Radar, missile, artillery and anti-aircraft gun. The obtained flight paths are compared with that of the standard Gravitational Search Algorithm (GSA) and two improved variants of GSA, i.e. *gbest*-guided GSA (GGSA), and hybrid Particle Swarm Optimization and GSA (PSOGSA). The experimental results demonstrate the superiority of the MSGSA based method in terms of the solution quality, robustness, as well as the constraint-handling ability.


**INDEX TERMS** Path planning, unmanned aerial vehicle (UAV), gravitational search algorithm (GSA), adaptive alpha-adjusting, Cauchy mutation.

## I. INTRODUCTION

Unmanned aerial vehicle (UAV) is an aircraft that can be remotely controlled or can fly autonomously based on a preprogrammed flight path without pilots onboard [1]. The aforementioned properties of UAV have made it being successfully used in a wide range of real-world applications, especially in the military side under extremely hazardous environmental conditions [2], [3]. Generally, the preprogrammed flight path in these applications requires to be as safe and short as possible to allow the UAV flies from a predesigned start location to a target position safely and

quickly [4]. Thus a more rational path of UAV should with better safety and shorter flight length. Obviously, how to program the optimal flight path, i.e., the design of path planning algorithm, is of great importance to UAV.

The path planning can be treated as a global optimization problem with various constraints from the certain mission, environment and UAV physical property [5]. For effective path planning, a number of environmental constraints (such as flight length and threat constraints) and UAV's self-constraints (such as flight altitude and turning angle constraints) should be considered. To solve this global optimization problem, different approaches based on the graph-theory [6], the mathematical programming [7], and the bi-level programming Liu *et al.* [8] have been proposed.

The associate editor coordinating the review of this manuscript and approving it for publication was Maurizio Magarini .

The Voronoi diagram search method [6] is a typical graph-based method. In this method, the battle area is partitioned into a number of convex polygons and in every polygon only one threat is contained. The optimal path is decided using the Eppstein's  $k$ -best algorithm in the presence of simple disjoint polygonal obstacles. The Eppstein's  $k$ -best algorithm is also utilized in both the mathematical programming methods and the bi-level programming methods [5]. However, the Eppstein's  $k$ -best based Voronoi diagram search method has difficulty to consider the motion constraints of UAVs [9]. To realize real-time route planning for UAVs, the sparse  $A^*$  search algorithm [10] and the  $D^*$  Lite algorithm [11] were successively put forward. Unfortunately, the former can only work under the known environment and the latter may become time consuming when the problem space becomes larger and more complex.

Recently, the population-based meta-heuristic algorithms that can handle constrained optimization problems have become the efficient and effective techniques for path planning [2], [12]–[20]. By designing specific objective functions or constraint functions, both the self-constraints of UAV and environmental constraints can be taken into account during the planning process [21]–[23]. In [12], the genetic algorithm (GA) is combined with the Voronoi diagram to generate the optimal path for autonomous UAVs. In [23], the evolutionary algorithm is used to find the optimal path for the multiple UAVs based on many real-world simulations. Roberge *et al.* [2] compared the performance of parallel GA and particle swarm optimization (PSO) for real-time UAV path planning. In addition, the artificial bee colony (ABC) [13], ant colony optimization (ACO) [14], water drops optimization (WDO) [15], and their variations have also been utilized to search the optimal flight path for single or multiple UAVs. More critical situation such as scenarios with disaster and dynamic threats usually desires improved intelligent algorithms [18]–[20]. More recently, the development of deep learning also spawned the deep reinforcement learning based path planning [24].

Gravitational search algorithm (GSA) is one of the most popular meta-heuristic algorithms inspired by the law of gravity and mass interactions [25]. In GSA, particles with better fitness values are assigned larger masses. Following the law of gravity, those heavier masses can exert larger gravitational force to the other particles. Therefore, those particles with smaller masses, i.e. the particle with worse fitness values, can move towards those better ones. In the past ten years, the GSA has shown its simplicity in concept, fast speed in convergence, and high precision in optimization [26]–[30]. Moreover, GSA and its variants have been applied to many real-world problems, including feature selection [31], image segmentation [32], and design of pow flow [33]. Further, Li and Duan [34] has introduced GSA to the path planning area and verified the potential of GSA for solving the path planning problem.

However, the inherent characteristics of GSA limited its application in path planning [35]–[38]. Especially GSA

overly emphasize the fast convergence lead to the algorithm cannot widely explore the search space in the early stage and failed to sufficiently exploit the promising area. That is, GSA cannot balance the exploitation and exploration well when facing to the complicated problem. Additionally, this kind of imbalance easily lead to premature convergence. To overcome these issues, numerous strategies have been proposed. These strategies can be classified into three aspects. Firstly, hybridize GSA with other meta-heuristics algorithms. Typical methods including differential evolution (DE) [39], PSO [37], [40]–[42], GA [27], [43], ABC [44] and Simulated Annealing (SA) [45]. Secondly, modify GSA using some other learning strategies, such as neighborhood learning strategy [45], [46], black hole attraction [47], disruption [48], and quantum based methods [49], [50]. Thirdly, adjust the parameter in GSA. Actually, the only one parameter in GSA is the attenuation factor of gravitational constant. According to the law of gravity, gravitational force  $F = G \cdot Mm/R^2$ , thus the gravitational constant  $G^t$  can directly affect the value of gravitational force exerted on a particle. Larger gravitational force produces the bigger acceleration and thereby endow a particle with fast convergence speed. Obviously, the gravitational constant play an important role for promoting the balance between exploration and exploitation [29], [51], [52].

Although the aforementioned strategies can promote the performance of GSA, seldom strategy can improve it from different aspects. For example, the hybrid of GA and GSA, especially the usage of GA's mutation operator can diverse the population and thereby help GSA escape from local optima, while the modification of the neighborhood learning can adjust the global and local learning ability of GSA [27], [43]. However, these kinds of methods may increase the computational complexity of the GSA. The design of parameter  $G^t$  is useful to balance exploration and exploitation but it is helpless when the algorithm is trapped in stagnation [29], [51]–[54].

To tackle the aforementioned issues of GSA and promote its application in path planning of UAV, a mix-strategy based GSA (MSGSA) is proposed in this paper. The first strategy is focusing on the adaptive adjustment of  $G^t$ . As the attenuation factor  $\alpha$  directly affects the changing of  $G^t$ , we adjust the value of  $\alpha$  based on the evolutionary state of the global best particle ( $gbest$ ). This strategy is utilized to keep a balance between the exploration and exploitation in the searching process. Another strategy is the Cauchy mutation strategy which is introduced to revise the  $gbest$  when it cannot improve the self-solution and thus alleviate the premature convergence problem. In the path planning process, the MSGSA is used to search for the optimal flight path based on an objective function which considers the length cost, threat cost, and turning angle constraints simultaneously.

The remainder of this paper is organized as follows. Section II describes the main related works and details of the proposed method, including the route representation and the design of objective function, followed by the detailed introduction of the proposed MSGSA and its

implementation in path planning. In Section III, the parameter setting and experimental results are described. Then this study is discussed in Section IV. Finally, the paper is concluded in Section V.

## II. MATERIALS AND METHODS

### A. PATH REPRESENTATION OF UAV

The path planning of UAV aims to program a flight path for a start location to a target position  $T = (x_T, y_T, z_T)^T$ , which can be considered as a global optimization problem. The flight path from  $S$  to  $T$  can be defined as a curve with  $D$  turning point, i.e.  $\{S, P_1, P_2, \dots, P_D, T\}$ . The final route can be obtained by connecting all the adjacent  $D + 2$  points into a broken line and smoothing it at last. Obviously, the path planning requires the determination of  $D$  points with 3 dimensional coordinate. With the increase of point number, it will be complicate and time consuming.

Coordinate system transformation can help to simplify the computation as well as accelerate the searching speed. In specific, we first transform the coordinate system from  $O-XYZ$  to  $O'-X'Y'Z'$ , where the  $S$  is defined as the origin of coordinates ( $O'$ ) and the line from  $S$  to  $T$  is defined as the  $X'$ -axis. Following the law of coordinate system rotation, the geometrical relationship of the same waypoint ( $(x, y)$  and  $(x', y')$ ) in the two coordinate systems is [5]:

$$\begin{bmatrix} x' \\ y' \\ z_S \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x - x_S \\ y - y_S \\ z_S \end{bmatrix} \quad (1)$$

where

$$\theta = \arcsin \frac{y_T - y_S}{ST} \quad (2)$$

where  $(x, y)$  and  $(x', y')$  is the corresponding  $X$  and  $Y$  coordinates of one point in the original and rotated coordinate systems, respectively. Due to the  $Z$  coordinate is kept in the transform, the rotated angle  $\theta$  is the angle between the axes  $OX$  and  $O'X'$  as shown in Fig. 1.

As illustrated in Fig. 1, the  $D$  turning points are the segment points that divide the line  $ST$  in the  $O'X'$  axis into  $D$  equal segments, the abscissas of the  $D$  points in the rotated coordinate system are  $P'X = (x'_1, x'_2, \dots, x'_D) = (ST/D, 2*ST/D, \dots, ST)$ . Thus, finding the ordinates  $PY' = (y'_1, y'_2, y'_D)$  and flight altitude  $PZ' = (z'_1, z'_2, z'_D)$  of the  $D$  points and connecting the  $D$  points from the start to the target points will form a flight path.

Besides, the flight environment, mainly refer to the threat areas (includes missiles, radars, and artillery etc. as illustrated by blue circles in Fig. 1), is assumed as priori known followed the suggested in [4]. As shown Fig. 1, the threat radius of each threat area is limited which means each threat point can only effect a limited range. In other words, there is no threat out of the sphere. An effective flight path should be as short as possible without encountering any threat.

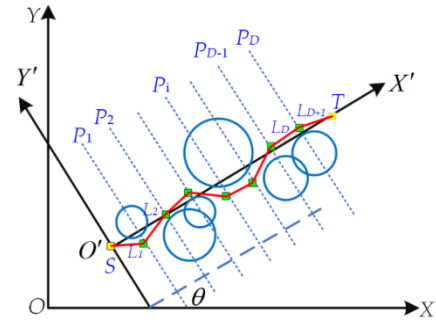


FIGURE 1. Representation of the path planning problem.

### B. DESIGN OF THE OBJECTIVE FUNCTION

The objective function is the core of an optimization problem. In the path planning problem, it is used to evaluate the quality of the candidate flight path. Generally, it is determined as the weight sum of the cost of the flight length and the cost of threat area constraints. The objective function is usually formulated as Eq. (3) below [34] and the purpose of path planning is to minimize the objective function:

$$\min J_{UAV} = w_1 \cdot J_{length} + w_2 \cdot J_{threat} \quad (3)$$

where  $J_{UAV}$  is the overall cost of the flight path,  $J_{length}$  is the length of the path and  $J_{threat}$  is exerted by all the threat areas in the flying space,  $w_1$  and  $w_2$  are the weighting coefficients for  $J_{length}$  and  $J_{threat}$ , respectively. Usually,  $w_1 + w_2 = 1$  is required. Obviously, the adjustment coefficients give the designer certain flexibility to dispose relations between the threat exposition degree and the fuel consumption. When  $w_1$  is more approaching 1, a shorter path is needed to be planned, and less attention is paid to exposure to threats. Otherwise, when  $w_2$  is more approaching 1, it requires avoiding the threat as far as possible on the cost of more fuel consumption [34]. Following the previous experiments in [34], both the coefficients are set to 0.5 in this paper. Fig. 2(a) shows the optimal flight path produced by MSGSA with the widely used objective function as given in Eq. (3).

Due to the objective function only considering the length cost and threat cost while ignoring the turning angle constraint, the generated flight path in Fig. 2(a) shows poor performance with very large turning angles. In realistic applications, larger turning angle will increase the risk of the flight mission. To decrease the flight risk and improve the smoothness of the path, in this paper, we take the turning angle constraint into consideration. Specifically, once the turning angle at the point  $P_i$  is larger than  $60^\circ$ , its position is reset by median interpolation method. Thereby, the path planning objective function should be written as:

$$\begin{aligned} \min J_{UAV} &= w_1 \cdot J_{length} + w_2 \cdot J_{threat} \\ s.t. \quad A_i &\leq \frac{\pi}{3} \quad (i = 1, 2, \dots, D) \end{aligned} \quad (4)$$

where  $A_i$  is the turning angle on  $i$ -th turning point,  $A_0$  is the turning angle between the start point  $S$  and  $P_1$ .

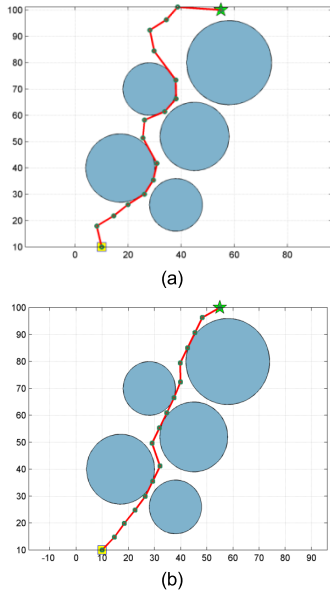


FIGURE 2. The optimal flight path produced by MSGSA using the objective function Eq. (3); (b) Optimal flight path based on Eq. (4).

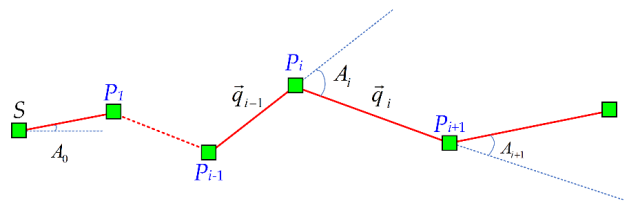


FIGURE 3. Calculation of turning angle  $A_i$ .  $S$  is the start point.  $A_0$  is the turning angle between the start point  $S$  and  $P_1$ .  $P_i$  ( $i \in \{1, 2, \dots, D\}$ ) is the  $i$ -th turning point,  $\vec{q}_{i-1}$  and  $\vec{q}_i$  respectively respects the flight vector from  $P_{i-1}$  to  $P_i$  and that from  $P_i$  to  $P_{i+1}$ .

As shown in Fig. 3, the turning angle is calculated by the flight vector. For the three turning points  $P_{i-1}$ ,  $P_i$ , and  $P_{i+1}$ , the flight vector from  $P_{i-1}$  to  $P_i$  is represented by  $\vec{q}_{i-1}$  and that from  $P_i$  to  $P_{i+1}$  is denoted by  $\vec{q}_i$ .  $A_i$  represents the angle between vector  $\vec{q}_{i-1}$  and vector  $\vec{q}_i$ , namely the turning angle of UAV at the  $i$ -th turning point.

$$A_i = \arccos \frac{\vec{q}_{i-1} \cdot \vec{q}_i}{\|\vec{q}_{i-1}\| \cdot \|\vec{q}_i\|} \quad (5)$$

where

$$\vec{q}_{i-1} = \frac{(P_i - P_{i-1})}{\|P_i - P_{i-1}\|}, \quad \vec{q}_i = \frac{(P_{i+1} - P_i)}{\|P_{i+1} - P_i\|} \quad (6)$$

Thus, the total turning angle can be summarized by:

$$J_{angle} = A_0 + \sum_{i=1}^D A_i \quad (7)$$

Fig. 2(b) displays the optimal flight path obtained by the MSGSA with the modified objective function Eq. (4). Obviously, the objective function design plays a key role for it is constructed by considering both the internal feature of UAV and the external features of the flying space, where optimization of the objective function produces the optimal

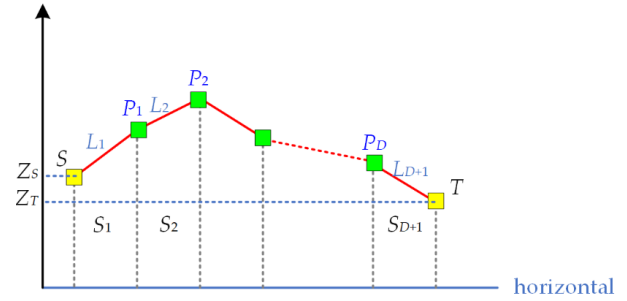


FIGURE 4. Illustration for the calculation of  $J_{length}$  in the 3-dimensional space.  $S$  and  $T$  is the start and target points, respectively.  $Z_s$  and  $Z_t$  are the corresponding altitude.  $P_i$  ( $i \in \{1, 2, \dots, D\}$ ) is the  $i$ -th turning point,  $L_i$  ( $i \in \{1, 2, \dots, D+1\}$ ) is the  $i$ -th flight path.  $S_i$  ( $i \in \{1, 2, \dots, D+1\}$ ) is trapezoid area enclosed by points  $P_i$  and  $P_{i-1}$  and their projection points on the horizontal plane.

flight path. The detailed formulas of the length cost and threat cost are given in the following subsections.

### 1) LENGTH COST

To decrease the energy consumption, such as fuel cost, a higher quality flight path should be of smaller length cost. As shown in Fig. 1, for the schematic flight path in a 2-dimensional space, the length cost  $J_{length}$  is the sum of  $D+1$  path segments, which is defined by:

$$J_{length} = \sum_{i=1}^{D+1} L_i \quad (8)$$

where

$$L_i = \sqrt{(x'_{i-1} - x'_i)^2 + (y'_{i-1} - y'_i)^2} \quad (9)$$

where  $(x'_i, y'_i)$  and  $(x'_{i-1}, y'_{i-1})$  are the  $i$ -th and  $i-1$ -th points of the flight path in the coordinate system  $O' - X' - Y'$ . To decrease the energy consumption, such as fuel cost, a higher quality flight path should be of smaller length cost.

For a 3-dimensional space, the  $J_{length}$  is also related to the flight altitude as shown in Fig. 4. Here the  $J_{length}$  is defined by the area of the vertical section composed of flight path and plane as follows.

$$J_{length} = \sum_{i=1}^{D+1} S_i \quad \text{where } S_i = (z_i + z_{i-1}) \times L_i / 2 \quad (10)$$

where  $z_i$  and  $z_{i-1}$  are the flight altitude of UAV at the two points.  $L_i$  is the length cost of the  $i$ -th path segment calculated by Eq. (9).

### 2) THREAT COST

The threat cost is used to represent the threat that exerted to the flight path. The main risk sources include Radar, missile, artillery, anti-aircraft gun, user-defined forbidden flying areas (i.e., the NFZs [4]) and so on. Except for the NFZs, the threat area of the other risk sources is looked as sphere as shown in Fig. 1. The function of the threat cost is often designed based on the distance of each path segment  $L_i$  (shown in Fig. 1) to the center of the threat areas and their corresponding

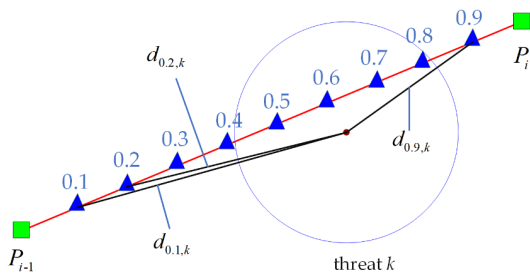


FIGURE 5. Calculation of threat cost of  $L_j$ .

threat levels [4]. In this paper, the threat cost of  $L_i$  is modeled by Eq. (11) follows [34]:

$$J_{threat} = \begin{cases} \frac{L_i^5}{5} \sum_{k=1}^{N_t} th_k \\ \times \left( \frac{1}{d_{0,1,k}} + \frac{1}{d_{0,2,k}} + \dots + \frac{1}{d_{0,9,k}} \right), & \text{if } d_{min} < R_k \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where  $N_t$  is the number of threat areas,  $k$ -th is the threat level of the  $k$ -th threat area,  $d$  is the distance from the center of the  $k$ -th threat area to the path segment  $L_i$  and  $d_{min}$  is the minimum distance between  $L_i$  and the center of the  $k$ -th threat area. Detailed description of  $d_{j,k}$  ( $j = 0.1, 0.2, \dots, 0.9$ ) is shown in Fig. 5.

Besides, due to UAV cannot appear in the NFZs, the threat cost inside the NFZs is defined as infinity while that outside the NFZs is set to zero, i.e.:

$$J_{threat}^{NFZ} = \begin{cases} \Lambda, & \text{if } L_i \cap NFZ \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

where  $\Lambda$  respects a very large number.

### C. THE PROPOSED MSGSA

#### 1) REVIEW OF THE STANDARD GSA

GSA search for the optimal solutions follows the gravitational forces exerted by its neighbors according to the law of gravity [26]. Specifically, each particle  $X_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$  respects one prepared solution in the  $D$ -dimensional search space with a velocity  $V_i = [v_{i1}, v_{i2}, \dots, v_{iD}]$ . Due to the gravitational force between two particles is directly proportional to their masses and inversely proportional to their distance, all the particles move towards those particles that have heavier masses [26]. The mass of each particle  $M_i^t$  at generation  $t$  is calculated from its fitness value  $fit_i^t$ . That is,

$$N_{fit_i^t} = \frac{fit_i^t - worst^t}{best^t - worst^t} \quad (13)$$

$$M_i^t = \frac{N_{fit_i^t}}{\sum_{j=1}^N N_{fit_j^t}} \quad (14)$$

For a minimization problem,  $worst^t$  and  $best^t$  are defined by Eqs. (15)-(16) as follows:

$$worst^t = \max_{j \in \{1, \dots, NP\}} fit_j^t \quad (15)$$

$$best^t = \min_{j \in \{1, \dots, NP\}} fit_j^t \quad (16)$$

where  $NP$  is the size of the particles in GSA.

The force acting on the particle  $i$  from the particle  $j$  at generation  $t$  is calculated by:

$$F_{id,jd}^t = G^t \frac{M_i^t \times M_j^t}{R_{ij}^t + \varepsilon} (x_{jd}^t - x_{id}^t) \quad (17)$$

where  $G^t$  is the gravitational constant;  $M_i^t$  and  $M_j^t$  are the gravitational mass of the  $i$ -th and  $j$ -th particles, respectively;  $x_{id}^t$  and  $x_{jd}^t$  are their corresponding position in the  $d$ -th dimension;  $R_{ij}^t$  is the Euclidian distance between the two particles in generation  $t$ ;  $\varepsilon$  is a small but positive constant which always is set to  $e^{-6}$ .

Note that the gravitational constant  $G^t$  in GSA is not keep unchanged in the whole convergence process. It is initialized as a  $G_0$  ( $G_0 = 20$  in standard GSA). With the convergence of the algorithm, to decrease the convergence speed and perform refine exploitation of the search space,  $G^t$  is decreased as defined in Eq. (18).

$$G^t = G^0 \times \exp\left(-\alpha \times \frac{t}{T_{max}}\right) \quad (18)$$

where  $\alpha$  is the attenuation factor of gravitational constant,  $T_{max}$  is the termination parameter, i.e. the maximum number of iterations. In the standard GSA,  $\alpha$  is set to 20.

According to the Newton's Second Law of Motion, the acceleration of particles ( $a_{id}^t$ ) can be obtained by:

$$a_{id}^t = \frac{F_{id}^t}{M_i^t} \quad (19)$$

Obviously, the gravitational constant  $G^t$  can control the effects of the gravitational force as it is the coefficient of all gravitational forces exerted on each particle. In other words,  $G^t$  defectively affect the acceleration of particles and thereby controls their movement steps and convergence speed in each iteration. Thus, designing of  $G^t$  and adjusting of  $\alpha$  is an important branch of GSA promote variants.

Further, note that in the standard GSA, Eq. (17) always written as Eq. (20) as follows to keep a better convergence accuracy.

$$F_{id}^t = \sum_{j \in K_{best}, j \neq i}^{NP} rand_j F_{id,id}^t \quad (20)$$

where  $K_{best}$  is the archive stores the  $K$  superior particles (with bigger masses) after fitness sorting in each iteration. The initial value of  $K_{best}$  is the same as the population size, i.e.  $NP$ . The size of  $K_{best}$  is then linearly decreased to 1 with time going.  $rand_j$  is a uniform random variable in the interval  $[0, 1]$ .

Accordingly, in generation  $t$ , the velocity and the position of the particle  $i$  are updated by:

$$v_{id}^{t+1} = rand_i \times v_{id}^t + a_{id}^t \quad (21)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (22)$$

### 2) ADAPTIVE ALPHA-ADJUSTING STRATEGY

As seen in Eqs. (18)-(22), a larger  $G^t$  makes particles move with bigger steps towards the center of  $K_{best}$  and the algorithm focuses on global exploration, while a smaller  $G^t$  makes particles move with smaller steps and the algorithm mainly concentrates on exploitation of the current search area. Obviously, the  $G^t$  plays a key role in adjusting the balance between exploration and exploitation of GSA. Moreover, due to the value of  $G^t$  are affected by the attenuation factor alpha ( $\alpha$ ) as introduced in Eq. (18), the parameter  $\alpha$  plays an important role controlling the searching ability of GSA.

In the standard GSA, the parameter is set to a constant in the whole evolutionary process which overlooks the evolutionary state of the population. In this paper, we propose an adaptive alpha-adjusting strategy to adjust the value of  $\alpha$  based on the evolutionary state of the global best particle ( $gbest$ ). Specifically, the convergence situation of the  $gbest$  is used to define its evolutionary state. When the  $gbest$  is improved in the current iteration, we assume the algorithm is converging to better solution space. In this case, other particles should move towards  $gbest$  with larger acceleration. Thereby, the decreased speed of  $G^t$  should be reduced and the value of  $\alpha$  should be increased. Otherwise, when the  $gbest$  cannot be improved, the population is assumed to be trapped into a local optimum and other particles should decrease their own velocity towards the  $gbest$ . Accordingly, the value of  $\alpha$  should be decreased and the decreased speed of  $G^t$  thereby can be raised. Therefore, the proposed strategy can be defined as follows:

$$\alpha^t = \begin{cases} \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \frac{t}{T_{\max}}, & \text{if } gbest \text{ is improved} \\ \alpha^{t-1} - (\alpha^{t-1} - \alpha_{\min}) \frac{t}{T_{\max}}, & \text{otherwise} \end{cases} \quad (23)$$

where  $\alpha_{\min}$  and  $\alpha_{\max}$  respectively represents the minimum and maximum value of the parameter  $\alpha$ . In the proposed MSGSA, is set to 15 and is set to 25 by trail and errors. By using this strategy, particles can adjust their move speed and tendency towards the  $gbest$ . Hence, this strategy can help to balance the exploration and exploitation of GSA based on the evolutionary state of the algorithm.

### 3) CAUCHY MUTATION STRATEGY

To further alleviate the premature convergence problem, a mutation strategy based on the Cauchy distribution density function is brought into GSA. This strategy is used to adjust the particles' positions once the  $gbest$  particle cannot be improved for  $lp$  (set to 5 in this paper) generation. The Cauchy distribution density function is selected to replace

### Algorithm 1 Mixed-Strategy Based Gravitational Search Algorithm (MSGSA)

- 1: Random initialize the position  $X_i$  and velocity  $V_i$  of  $NP$  particle.  $L_{best} = 1$ .
- 2: /\*Main Loop\*/
- 3: **for**  $t$  from 1 to  $T_{\max}$  **do**
- 4: Evaluate the fitness value of each particle.
- 5: Compute the  $worst^t$  and  $best^t$  respectively by Eq. (15) and Eq. (16).
- 6: **if**  $best^t < best^{t-1}$
- 7:  $L_{best} = L_{best} + 1$
- 8: **end if**
- 9: Compute the mass of each particle  $M_i^t$  by Eqs. (13)-(14).
- 10: Update the  $K_{best}$ .
- 11: Compute the parameter  $\alpha^t$  by Eq. (23).
- 12: Update the  $G^t$  by Eq. (18).
- 13: Compute the total force exerted on each particle based on Eq. (20).
- 14: Update the acceleration of each particle by Eq. (19).
- 15: Compute the velocity of each particle by Eq. (21).
- 16: Update the position of each particle by Eq. (22).
- 17: **if**  $L_{best} \geq lp$
- 18: Update the position of each particle by Eq. (25).
- 19: **end if**
- 20: **end for**
- 21: **Output** the particle with best fitness value.

the traditional Gaussian distribution mainly because of the Gaussian distribution, the Cauchy distribution at the origin of peak value is smaller. Moreover, the speed of both ends in Cauchy distribution close to zero is slower. Thus, its disturbance ability is stronger than that of the Gaussian mutation.

The probability density function is defined as follows:

$$f_c(z) = \frac{1}{\pi} \cdot \frac{\gamma}{\gamma^2 + (z - z_0)^2} \quad (24)$$

where  $\gamma > 0$  is the proportional parameter,  $z_0$  is the position of the peak, and  $C(z_0, \gamma_2)$  is the Cauchy distribution. In this paper,  $z_0$  is set to 0 and  $\gamma$  is set to 1, the distribution range is set to the boundary of the search space.

Based on the Cauchy distribution, we define a Cauchy mutation strategy for the MSGSA as follows:

$$x_{id}^{t+1} = x_{id}^t + x_{id}^t \cdot C(0, 1) \quad (25)$$

The particles in MSGSA are therefore more likely to jump out of local optima, make full use of the current state of the particles, avoid the blindness of random initialization, and improve the search speed.

The pseudocode of the proposed MSGSA is presented in **Algorithm 1** below.

### 4) PATH PLANNING USING MSGSA

Generally, the path planning procedure contains four parts: (1) internal feature descriptions, i.e. the description of the

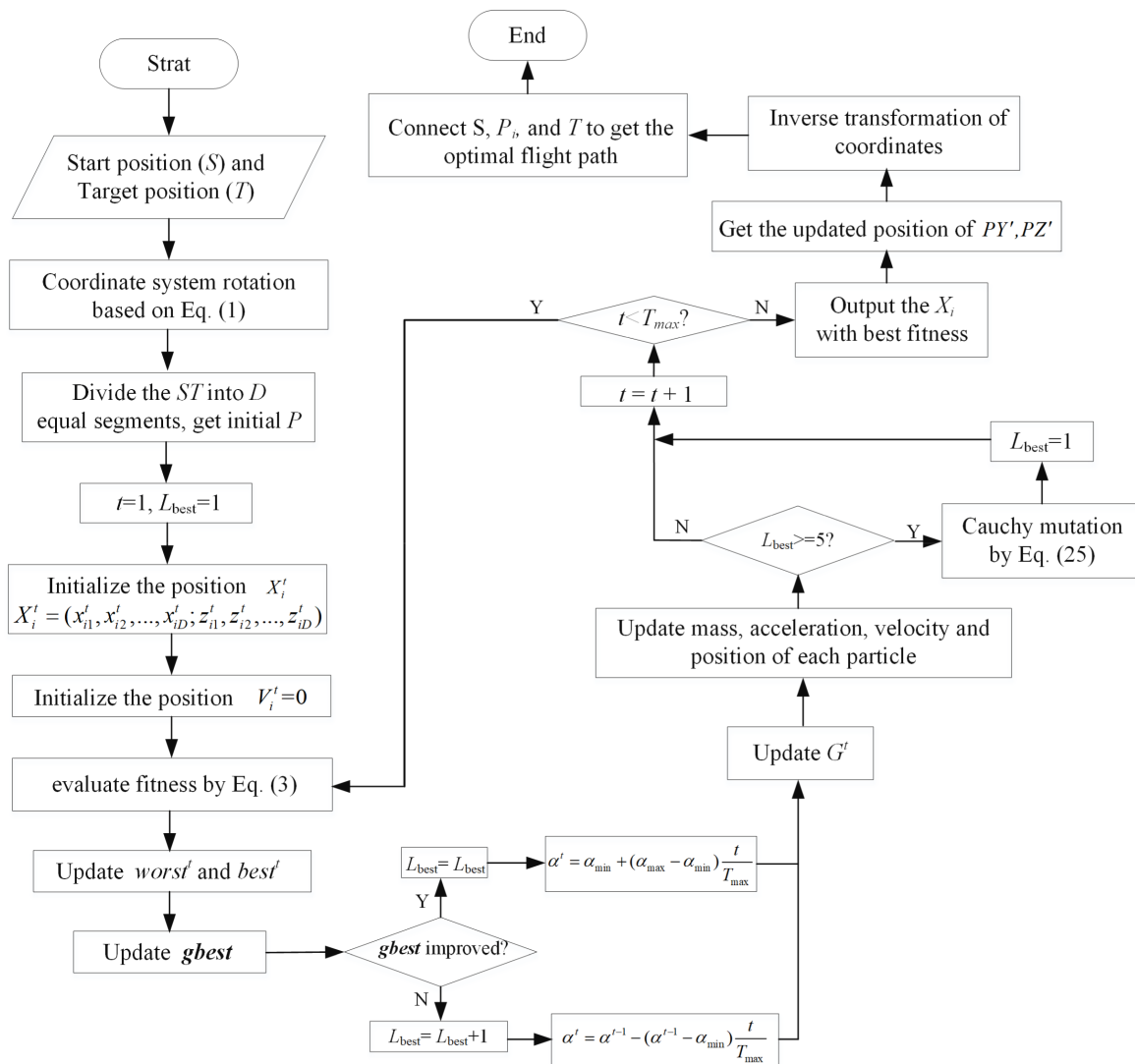


FIGURE 6. The flowchart of MSGSA-based path planning.

self-constraints of UAV, including the fuel cost and the angle constraints, (2) external feature descriptions, i.e. the threat modeling of the flying space, (3) objective function design, and (4) algorithm realization. Based on the methods introduced in the Sections II.A, II.B and II.C, the flowchart of the path planning using MSGSA is given in Fig. 6.

The detailed steps of preprogram of path planning using MSGSA are summarized as follows:

Step 1: Establish the rotating coordinate system and map the threat areas to the rotated coordinate system based on Eq. (1).

Step 2: Divide the line  $ST$  into  $D$  equal segments, the coordinate system is denoted as  $P = (PX', PY', PZ')$  where  $PX' = (ST/D, 2 * ST/D, \dots, ST)$ ,  $PY' = (y'_1, y'_2, \dots, y'_D) = (0, 0, \dots, 0)$ , and  $PZ' = (z_1, z_2, \dots, z_D)$

Step 3: Initialize the prepared solutions randomly by  $X_i = (y'_{i1}, y'_{i2}, \dots, y'_{iD}; z_{i1}, z_{i2}, \dots, z_{iD})$  ( $i = 1, 2, \dots, NP$ ) and the corresponding velocity  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD}) = (0, 0, \dots, 0)$ .

Step 4: Calculate the fitness value  $J_{UAV}$  of each flight path using Eq. (3) and update the  $gbest$ .

Step 5: If  $gbest$  cannot improve its self-solution for  $l_p$  generation, perform the Cauchy mutation strategy to escape from local optimum, otherwise go to Step 6.

Step 6: Update the parameter  $\alpha$  and  $G^t$  using the proposed adaptive alpha-adjusting strategy.

Step 7: Update the velocity and position of each particle and produce the  $NP$  flight paths.

Step 8: To determine whether the iteration is terminated, if not, return to Step 4. Otherwise, output the  $X_i$  with the smallest objective function value.

Step 9: Transform the coordinates of the  $D$  points in the outputted  $X_i$  to the  $O - XYZ$  coordinate system from the  $O' - X'Y'Z'$  coordinate system. Connecting the  $D$  points from the start to the target points will form the optimal flight path.

TABLE 1. Setting of the UAV scenarios.

	Mission	Threat center and NFZs	Threat radius	Threat level
Case 1	Start Point	Anti-aircraft gun [45,25]	13	2
	[10,10]	Radar 1 [17,40]	13	10
	Target point	Missile [28,70]	10	1
	[55,100]	Artillery [38,26]	10	2
		Radar 2 [58,80]	16	5
Case 2	Start Point	Anti-aircraft gun [250,150,20]	85	2
	[50,50,50]	Missile [300,500,50]	100	1
	Target point	Radar [600,600,10]	50	5
	[950,950,50]	Artillery [850,900,0]	80	3
		NFZs {[450,100],[550,400]}	100	1

TABLE 2. Parameter settings.

Algorithm	Parameters
MSGSA	$G_0=100, \alpha_{min}=15, \alpha_{max}=25, lp=5$
GSA	$G_0=100, \alpha=20$
GGSA	$G_0=100, \alpha=20, c_1=2-2t^3/T_{max}^2, c_2=2t^3/T_{max}^2$
PSOGSA	$G_0=100, \alpha=20, c_1=0.5, c_2=1.5$

III. RESULTS

In this section, a 2-D flight mission with 5 threat areas (Case 1) and a 3-D flight mission with 4 threat areas as well as 1 user-defined forbidden flying area (i.e. NFZ) (Case 2) are simulated to evaluate the performance of the proposed MSGSA. The vertices of the polygonal NFZ, the location of the start point, the target point, the threat centers and their threat radiuses/levels are listed in Table 1 (unit: km). The obtained results of the proposed algorithm were benchmarked with those of the standard GSA and two improved variants of GSA, i.e. *gbest*-guided GSA (GGSA), and hybrid Particle Swarm Optimization and Gravitational Search Algorithm (PSOGSA). All of the algorithms are implemented in MATLAB environment (Matlab 2018a) and ran on a server with a 3.0 GHz CPU (Inter Core i5-7400) and 8.0 GB of RAM.

A. PARAMETER SETING

Based on the description of the UAV scenario, the flying space of Case 1 and Case 2 is respectively defined as a square with a side length 100 km and 950 km in this paper. Besides, for Case 2, the flight altitude is limited to [0km, 100km]. To perform fair comparison, for all the four tested algorithm, the maximum number of iterations ( $T_{max}$ ), the number of prepared flight paths, i.e. the size of the population ( $NP$ ), and the number of segments, i.e. the turning point for each flight path ( $D$ ) are set to 1000, 100, and 15, respectively. Other parameters in GSA, GGSA, and PSOGSA are set as the parameters recommended in the previous studies as listed

in Table 2. Moreover, to reduce random discrepancy, all algorithms were independently run 30 times.

B. EXPERIMENTAL RESULTS

Following the experimental setup and parameter settings above, path planning experiments using MSGSA, GSA, GGSA, and PSOGSA are conducted.

To evaluate the search accuracy of these algorithms, the minimum (*Best*), maximum (*Worst*), average (*Mean*), and standard deviation (*Std*) of the  $J_{UAV}$ ,  $J_{length}$ ,  $J_{threat}$ , and  $J_{angle}$  (the sum of the turning constraints of the optimal path flight) of the 30 runs are taken as the performance metrics in this paper as listed in Tables 3-4. Moreover, the success rate,  $SR\%$ , is employed to reflect the search reliability of the corresponding algorithm as shown in Table 3. Note that when the  $J_{threat}$  of a candidate flight path is zero the algorithm is regarded as successful. The best result in each column is set in bold. The flight path of each algorithm obtained in the last iteration is given in Figs. 7-9.

For Case 1, as shown in Table 3, MSGSA obtains the best Mean value in terms of the objective function value  $J_{UAV}$ . In addition, the *Std* of MSGSA is the smallest which indicates its superior robustness. This mainly result from the evolutionary state-based alpha-adjusting and mutation strategies that promote the search ability of GSA. With respect to the  $SR\%$ , MSGSA is the only algorithm with 100% success. This shows the best searching reliability and the best risk aversion ability of MSGSA compared with other three algorithms. This is also confirmed by the statistic results of  $J_{threat}$  values as shown in Table 4. Moreover, the flight paths shown in Fig. 7, especially the position of the first three turning points highlighted by the blue square box, display that MSGSA can obtain more reasonable flight path.

Similarly, for Case 2, MSGSA generated the best results on all the metrics, followed by GSA. We can infer that the GGSA and PSOGSA may rely on the setting of the acceleration coefficients. For the more complicated 3D flight path planning problem, users need to take more time to adjusting them. The better robustness of MSGSA compared to the



TABLE 3. The  $J_{UAV}$  of path planning for Case 1 and Case 2.

Mission	Cost	Algorithm	Best	Worst	Mean	Std	SR%
Case 1	$J_{UAV}$	MSGSA	121.649	<b>122.813</b>	<b>122.221</b>	<b>0.385</b>	<b>100</b>
		GSA	121.722	124.518	122.798	0.980	80
		GGSA	121.477	126.594	123.230	1.559	90
		PSOGSA	<b>107.608</b>	134.128	125.040	7.478	90
Case 2	$J_{UAV}$	MSGSA	<b>6.082E+04</b>	<b>7.383E+04</b>	<b>6.510E+04</b>	<b>1.667E+03</b>	<b>100</b>
		GSA	6.957E+04	8.135E+04	7.499E+04	3.510E+03	86.7
		GGSA	7.445E+04	8.451E+05	8.143E+04	4.914E+03	36.7
		PSOGSA	8.655E+04	8.768E+04	8.079E+04	5.267E+03	33.3

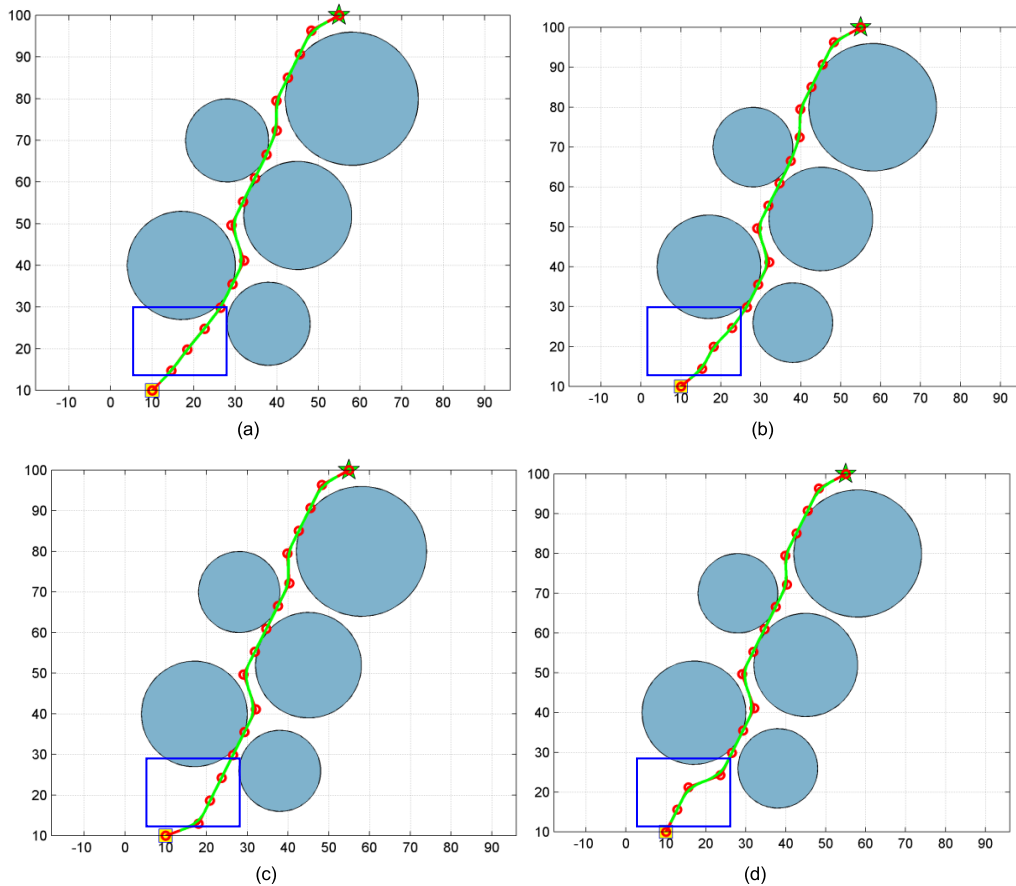
TABLE 4. The  $J_{threat}$ ,  $J_{length}$ , and jangle of path planning for Case 1 and Case 2.

Mission	Cost	Algorithm	Best	Worst	Mean	Std
Case 1	$J_{threat}$	MSGSA	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>
		GSA	<b>0.000</b>	6.848	1.803	2.978
		GGSA	<b>0.000</b>	6.916	3.378	3.567
		PSOGSA	<b>0.000</b>	5.448	0.545	1.723
	$J_{length}$	MSGSA	237.556	<b>239.676</b>	238.826	<b>0.661</b>
		GSA	238.131	239.800	239.183	0.669
		GGSA	237.508	242.119	239.795	1.355
		PSOGSA	<b>192.361</b>	239.799	<b>231.407</b>	15.703
	$J_{angle}$	MSGSA	284.007	318.246	294.989	<b>8.952</b>
		GSA	275.591	<b>295.791</b>	287.162	9.068
		GGSA	<b>226.348</b>	303.844	<b>279.768</b>	25.046
		PSOGSA	350.004	524.365	412.686	49.522
Case 2	$J_{threat}$	MSGSA	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>
		GSA	<b>0.000</b>	8.924E+02	2.231E+02	4.462E+02
		GGSA	<b>0.000</b>	1.977E+05	4.973E+04	4.863E+02
		PSOGSA	<b>0.000</b>	1.624E+04	6.168E+03	7.476E+03
	$J_{length}$	MSGSA	<b>1.124E+05</b>	<b>2.113E+05</b>	<b>1.505E+05</b>	<b>3.667E+04</b>
		GSA	1.448E+05	2.508E+05	1.947E+05	4.060E+04
		GGSA	1.664E+05	4.094E+05	3.334E+05	4.322E+04
		PSOGSA	1.880E+05	4.853E+05	3.973E+05	7.793E+04
	$J_{angle}$	MSGSA	<b>2.833E+02</b>	<b>3.378E+02</b>	<b>2.959E+02</b>	<b>2.984E+01</b>
		GSA	2.898E+02	4.002E+02	3.128E+02	3.891E+01
		GGSA	4.808E+02	1.107E+03	6.448E+02	4.770E+01
		PSOGSA	4.691E+02	9.377E+02	6.713E+02	5.486E+01

other three algorithms is also confirmed by the SR%. From Table 4, we can find that for the more complicated Case 2, the superiority of the MSGSA is more evident. As shown, for all the cost functions, the MSGSA based method obtained the best results, followed by the GSA based method. The flight paths shown in Fig. 8 also verify the advantage of the proposed method. Overall, MSGSA has the best risk aversion ability which can also balance the flight length and turning angle well.

#### IV. DISCUSSION

Path planning of the UAVs is required in many real-world applications, including military and scientific research areas. Although various techniques have been proposed for finding the optimal path, the mission is still challenging because of the changing environments [16]. In the past two decades, the population-based meta-heuristic algorithms including GSA have gained success in handling path planning problem with environmental and UAV's self-constraints [2]. However,



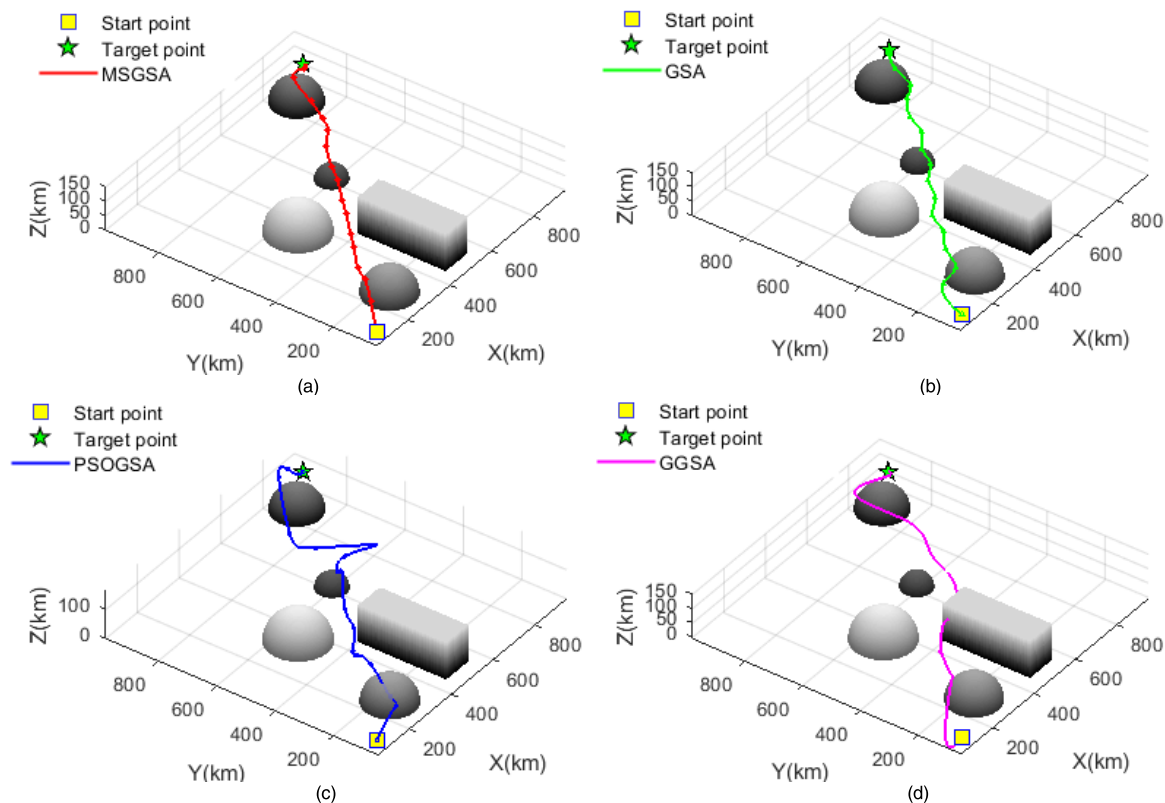
**FIGURE 7.** Flight paths and convergence curves comparisons in Case 1. (a) Flight path generated by MSGSA; (b) Flight path generated by GSA; (c) Flight path generated by PSOGSA; (d) Flight path generated by GGSA.

according to the “no free lunch theorem” [55], none of the meta-heuristic algorithm can perform well on all the path planning problems and each algorithm has its own inevitable flaws. Therefore, to alleviate the premature problem of GSA and design a more effective path planning method, the MSGSA is developed in this paper.

Owing to the adaptive alpha-adjusting strategy, the MSGSA can adjust the gravitational constant based on the evolutionary state of the algorithm. This makes the algorithm can balance the global and local search for the path planning with several constrains more adaptively. The Cauchy mutation strategy, on the other hand, is proposed to further promote the exploration ability of MSGSA when the objective function is a multimodal problem. Considering the general constraints of UAV, i.e. the length cost and threat cost, the MSGSA based method can project the route avoiding all the threats as shown in Fig. 2(a). However, due to the considered constraints are not enough, the Eq. (2) based method overly emphasizes the threat cost and hardly balance the length cost and threat cost. To meet this challenge, the turning angle constraint is taken into consideration in this paper and the experimental results confirmed the efficiency of the proposed method.

The comparison results given in the Section 3 also verified the superiority of the proposed method. Firstly, the MSGSA based method is the most robust algorithm because its  $SR\%$  is 100% as shown in Table 3. This may come from the adaptive alpha-adjusting strategy which improves the adaptability of the algorithm to the path planning problem. The robustness of the MSGSA is also confirmed by the  $Std$  of  $J_{threat}$ ,  $J_{length}$  and  $J_{angle}$  as displayed in Table 4. Secondly, the adaptive property and the mutation mechanism of MSGSA make the proposed method achieve the best balance among all the three constraints, i.e. the smallest  $J_{UAV}$  value. Although the PSOGSA can obtain the a much smaller  $J_{UAV}$  value for Case 1, i.e. 107.608 shown in Table 3, the PSOGSA always face to larger turning angle as cost for both Case 1 and Case 2 as presented in Table 4. Since large turning angle may lead to greater risk of flight and crash, the value of  $J_{angle}$  should be carefully considered.

Perhaps adaptive selection of the weight coefficients of different constraints as well as optimization of other parameters can further promote the efficiency of the proposed method. In addition, finding the optimal path for dynamic constraints [56] or by designing a multi-objective



**FIGURE 8.** Flight paths and convergence curves comparisons in Case 2. (a) Flight path generated by MSGSA; (b) Flight path generated by GSA; (c) Flight path generated by PSO GSA; (d) Flight path generated by GGSA.

function [57]–[59] can also be an alternative method for the future work.

**V. CONCLUSION**

In this paper, a mixed-strategy based GSA (MSGSA) is proposed and successfully applied to preprogram UAV path planning problem in two and three dimensional environments. In the proposed MSGSA, an adaptive alpha-adjusting strategy is presented first based on the evolutionary state of the global best particle to keep the balance between exploration and exploitation. Moreover, a Cauchy mutation strategy follows the evolutionary state of the whole population is introduced to further alleviate the premature problem of GSA. Additionally, in the path planning procedure, to decrease the flight risk and improve the path smoothness, the turning angle constraints is also considered to design a novel objective function combining with the length cost and threat cost. To evaluate the performance of the proposed path planning algorithm, the obtained results are compared with those of the three state-of-the-art algorithms, GSA, PSO GSA, and GGSA. The simulation results reveal the superiority of MSGSA in terms of the searching accuracy and searching reliability. MSGSA can not only avoid the threat from various threat areas but also balance the flight length and turning angle well. We just tested the availability of the proposed method for off-line path planning of UAV and its real-time application requires further studies.

**REFERENCES**

- [1] A. Babaie and J. Karimi, “Optimal trajectory planning for a UAV in presence of terrain and threats,” *Mech. J. Imam Hossein Univ.*, vol. 7, pp. 57–69, Jan. 2011.
- [2] V. Roberge, M. Tarbouchi, and G. Labonté, “Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2013, doi: 10.1109/TII.2012.2198665.
- [3] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, “Vector field following for miniature air vehicles,” *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 519–529, Jun. 2007, doi: 10.1109/TRO.2007.898976.
- [4] X. Zhang and H. Duan, “An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning,” *Appl. Soft Comput.*, vol. 26, pp. 270–284, Jan. 2015, doi: 10.1016/j.asoc.2014.09.046.
- [5] Y. Liu, X. Zhang, X. Guan, and D. Delahaye, “Adaptive sensitivity decision based path planning algorithm for unmanned aerial vehicle with improved particle swarm optimization,” *Aerosp. Sci. Technol.*, vol. 58, pp. 92–102, Nov. 2016, doi: 10.1016/j.ast.2016.08.017.
- [6] P. Bhattacharya and M. L. Gavrilova, “Voronoi diagram in optimal path planning,” in *Proc. 4th Int. Symp. Voronoi Diagrams Sci. Eng. (ISVD)*. Glamorgan, U.K.: IEEE, Jul. 2007, pp. 38–47.
- [7] Y. Wu and X. Qu, “Path planning for taxi of carrier aircraft launching,” *Sci. China Technol. Sci.*, vol. 56, no. 6, pp. 1561–1570, Jun. 2013, doi: 10.1007/s11431-013-5222-5.
- [8] W. Liu, Z. Zheng, and K.-Y. Cai, “Bi-level programming based real-time path planning for unmanned aerial vehicles,” *Knowl.-Based Syst.*, vol. 44, pp. 34–47, May 2013, doi: 10.1016/j.knosys.2013.01.011.
- [9] Y. Fu, M. Ding, C. Zhou, and H. Hu, “Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 6, pp. 1451–1465, Nov. 2013, doi: 10.1109/TSMC.2013.2248146.

- [10] R. J. Szczerba, P. Galkowski, I. S. Glicktein, and N. Ternullo, "Robust algorithm for real-time route planning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 3, pp. 869–878, Jul. 2000, doi: [10.1109/7.869506](https://doi.org/10.1109/7.869506).
- [11] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 354–363, Jun. 2005, doi: [10.1109/TRO.2004.838026](https://doi.org/10.1109/TRO.2004.838026).
- [12] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," *Aerosp. Sci. Technol.*, vol. 16, no. 1, pp. 47–55, Jan. 2012, doi: [10.1016/j.ast.2011.02.006](https://doi.org/10.1016/j.ast.2011.02.006).
- [13] C. Xu, H. Duan, and F. Liu, "Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning," *Aerosp. Sci. Technol.*, vol. 14, no. 8, pp. 535–541, Dec. 2010, doi: [10.1016/j.ast.2010.04.008](https://doi.org/10.1016/j.ast.2010.04.008).
- [14] G. Ma, H. Duan, and S. Liu, "Improved ant colony algorithm for global optimal trajectory planning of UAV under complex environment," *IJCSA*, vol. 4, no. 3, pp. 57–68, 2007, doi: [10.1016/j.ceramint.2018.10.039](https://doi.org/10.1016/j.ceramint.2018.10.039).
- [15] H. Duan, S. Liu, and J. Wu, "Novel intelligent water drops optimization approach to single UCAV smooth trajectory planning," *Aerosp. Sci. Technol.*, vol. 13, no. 8, pp. 442–449, Dec. 2009, doi: [10.1016/j.ast.2009.07.002](https://doi.org/10.1016/j.ast.2009.07.002).
- [16] 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, doi: [10.1109/TRO.2005.844684](https://doi.org/10.1109/TRO.2005.844684).
- [17] 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, doi: [10.1016/j.knsys.2018.05.033](https://doi.org/10.1016/j.knsys.2018.05.033).
- [18] J. D. S. Arantes, M. D. S. Arantes, C. F. M. Toledo, O. Trindade, Jr., and B. C. Williams, "Heuristic and genetic algorithm approaches for UAV path planning under critical situation," *Int. J. Artif. Intell. Tools*, vol. 26, no. 1, Feb. 2017, Art. no. 1760008, doi: [10.1142/S02182183017600089](https://doi.org/10.1142/S02182183017600089).
- [19] X. Yu, C. Li, and J. Zhou, "A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios," *Knowl.-Based Syst.*, vol. 204, Sep. 2020, Art. no. 106209, doi: [10.1016/j.knsys.2020.106209](https://doi.org/10.1016/j.knsys.2020.106209).
- [20] J.-J. Shin and H. Bang, "UAV path planning under dynamic threats using an improved PSO algorithm," *Int. J. Aerosp. Eng.*, vol. 2020, pp. 1–17, Dec. 2020, doi: [10.1155/2020/8820284](https://doi.org/10.1155/2020/8820284).
- [21] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 6, pp. 898–912, Dec. 2003, doi: [10.1109/TSMCB.2002.804370](https://doi.org/10.1109/TSMCB.2002.804370).
- [22] C. Hocaoglu and A. C. Sanderson, "Planning multiple paths with evolutionary speciation," *IEEE Trans. Evol. Comput.*, vol. 5, no. 3, pp. 169–191, Jun. 2001, doi: [10.1109/4235.930309](https://doi.org/10.1109/4235.930309).
- [23] E. Besada-Portas, L. de la Torre, J. M. de la Cruz, and B. de Andrés-Toro, "Evolutionary trajectory planner for multiple UAVs in realistic scenarios," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 619–634, Aug. 2010, doi: [10.1109/TRO.2010.2048610](https://doi.org/10.1109/TRO.2010.2048610).
- [24] C. Yan, X. Xiang, and C. Wang, "Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments," *J. Intell. Robot. Syst.*, vol. 98, no. 2, pp. 297–309, May 2020, doi: [10.1007/s10846-019-01073-3](https://doi.org/10.1007/s10846-019-01073-3).
- [25] S. Jiang, Y. Wang, and Z. Ji, "Convergence analysis and performance of an improved gravitational search algorithm," *Appl. Soft Comput.*, vol. 24, pp. 363–384, Nov. 2014, doi: [10.1016/j.asoc.2014.07.016](https://doi.org/10.1016/j.asoc.2014.07.016).
- [26] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009, doi: [10.1016/j.ins.2009.03.004](https://doi.org/10.1016/j.ins.2009.03.004).
- [27] A. Zhang, G. Sun, Z. Wang, and Y. Yao, "A hybrid genetic algorithm and gravitational search algorithm for global optimization," *Neural Netw. World*, vol. 25, no. 1, p. 53, 2015, doi: [10.14311/nnw.2015.25.003](https://doi.org/10.14311/nnw.2015.25.003).
- [28] S. Mirjalili, S. Z. M. Hashim, and H. M. Sardroudi, "Training feed-forward neural networks using hybrid particle swarm optimization and gravitational search algorithm," *Appl. Math. Comput.*, vol. 218, no. 22, pp. 11125–11137, Jul. 2012, doi: [10.1016/j.amc.2012.04.069](https://doi.org/10.1016/j.amc.2012.04.069).
- [29] J. V. Kumar, D. M. V. Kumar, and K. Edukondalu, "Strategic bidding using fuzzy adaptive gravitational search algorithm in a pool based electricity market," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2445–2455, May 2013, doi: [10.1016/j.asoc.2012.12.003](https://doi.org/10.1016/j.asoc.2012.12.003).
- [30] N. M. Sabri, M. Puteh, and M. R. Mahmood, "A review of gravitational search algorithm," *Int. J. Adv. Soft Comput. Appl.*, vol. 5, no. 3, pp. 1–39, 2013.
- [31] A. Zhang, P. Ma, S. Liu, G. Sun, H. Huang, J. Zabalza, Z. Wang, and C. Lin, "Hyperspectral band selection using crossover-based gravitational search algorithm," *IET Image Process.*, vol. 13, no. 2, pp. 280–286, Feb. 2019, doi: [10.1049/iet-ipr.2018.5362](https://doi.org/10.1049/iet-ipr.2018.5362).
- [32] G. Sun, A. Zhang, Y. Yao, and Z. Wang, "A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding," *Appl. Soft Comput.*, vol. 46, pp. 703–730, Sep. 2016, doi: [10.1016/j.asoc.2016.01.054](https://doi.org/10.1016/j.asoc.2016.01.054).
- [33] S. Duman, U. Güvenç, Y. Sönmez, and N. Yörükeren, "Optimal power flow using gravitational search algorithm," *Energy Convers. Manage.*, vol. 59, pp. 86–95, Jul. 2012, doi: [10.1016/j.enconman.2012.02.024](https://doi.org/10.1016/j.enconman.2012.02.024).
- [34] P. Li and H. Duan, "Path planning of unmanned aerial vehicle based on improved gravitational search algorithm," *Sci. China Technol. Sci.*, vol. 55, no. 10, pp. 2712–2719, Oct. 2012, doi: [10.1007/s11431-012-4890-x](https://doi.org/10.1007/s11431-012-4890-x).
- [35] S. Gao, C. Vairappan, Y. Wang, Q. Cao, and Z. Tang, "Gravitational search algorithm combined with chaos for unconstrained numerical optimization," *Appl. Math. Comput.*, vol. 231, pp. 48–62, Mar. 2014, doi: [10.1016/j.amc.2013.12.175](https://doi.org/10.1016/j.amc.2013.12.175).
- [36] N. Zhang, C. Li, R. Li, X. Lai, and Y. Zhang, "A mixed-strategy based gravitational search algorithm for parameter identification of hydraulic turbine governing system," *Knowl.-Based Syst.*, vol. 109, pp. 218–237, Oct. 2016, doi: [10.1016/j.knsys.2016.07.005](https://doi.org/10.1016/j.knsys.2016.07.005).
- [37] S. Mirjalili and A. Lewis, "Adaptive gbest-guided gravitational search algorithm," *Neural Comput. Appl.*, vol. 25, nos. 7–8, pp. 1569–1584, Dec. 2014, doi: [10.1007/s00521-014-1640-y](https://doi.org/10.1007/s00521-014-1640-y).
- [38] C. Li, L. Chang, Z. Huang, Y. Liu, and N. Zhang, "Parameter identification of a nonlinear model of hydraulic turbine governing system with an elastic water hammer based on a modified gravitational search algorithm," *Eng. Appl. Artif. Intell.*, vol. 50, pp. 177–191, Apr. 2016, doi: [10.1016/j.engappai.2015.12.016](https://doi.org/10.1016/j.engappai.2015.12.016).
- [39] X. Li, M. Yin, and Z. Ma, "Hybrid differential evolution and gravitation search algorithm for unconstrained optimization," *Int. J. Phys. Sci.*, vol. 6, pp. 5961–5981, Oct. 2011.
- [40] S. Jiang, Z. Ji, and Y. Shen, "A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints," *Int. J. Electr. Power Energy Syst.*, vol. 55, pp. 628–644, Feb. 2014, doi: [10.1016/j.ijepes.2013.10.006](https://doi.org/10.1016/j.ijepes.2013.10.006).
- [41] S. Mirjalili and S. Z. M. Hashim, "A new hybrid PSO-GSA algorithm for function optimization," in *Proc. Int. Conf. Comput. Inf. Appl.* Tianjin, China: IEEE, Dec. 2010, pp. 374–377.
- [42] H.-C. Tsai, Y.-Y. Tyan, Y.-W. Wu, and Y.-H. Lin, "Gravitational particle swarm," *Appl. Math. Comput.*, vol. 219, no. 17, pp. 9106–9117, May 2013, doi: [10.1016/j.amc.2013.03.098](https://doi.org/10.1016/j.amc.2013.03.098).
- [43] S. Sheikhpour, M. Sabouri, and S.-H. Zahiri, "A hybrid gravitational search algorithm-genetic algorithm for neural network training," in *Proc. 21st Iranian Conf. Electr. Eng. (ICEE)*. Mashhad, Iran: IEEE, May 2013, pp. 1–5.
- [44] Z. G., "A hybrid optimization algorithm based on artificial bee colony and gravitational search algorithm," *Int. J. Digit. Content Technol. Appl.*, vol. 6, no. 17, pp. 620–626, Sep. 2012, doi: [10.4156/jdcta.vol6.issue17.68](https://doi.org/10.4156/jdcta.vol6.issue17.68).
- [45] X. Li, J. Wang, J. Zhou, and M. Yin, "An effective GSA based memetic algorithm for permutation flow shop scheduling," in *Proc. IEEE Congr. Evol. Comput.* Barcelona, Spain: IEEE, Jul. 2010, pp. 1–6.
- [46] G. Sun, A. Zhang, Z. Wang, Y. Yao, J. Ma, and G. D. Couples, "Locally informed gravitational search algorithm," *Knowl.-Based Syst.*, vol. 104, pp. 134–144, Jul. 2016, doi: [10.1016/j.knsys.2016.04.017](https://doi.org/10.1016/j.knsys.2016.04.017).
- [47] M. Doraghinejad and H. Nezamabadi-Pour, "Black hole: A new operator for gravitational search algorithm," *Int. J. Comput. Intell. Syst.*, vol. 7, no. 5, pp. 809–826, Sep. 2014, doi: [10.1080/18756891.2014.966990](https://doi.org/10.1080/18756891.2014.966990).
- [48] S. Sarafrazi, H. Nezamabadi-Pour, and S. Saryazdi, "Disruption: A new operator in gravitational search algorithm," *Scientia Iranica*, vol. 18, no. 3, pp. 539–548, Jun. 2011, doi: [10.1016/j.scient.2011.04.003](https://doi.org/10.1016/j.scient.2011.04.003).
- [49] M. Soleimanpour-Moghadam and H. Nezamabadi-Pour, "An improved quantum behaved gravitational search algorithm," in *Proc. 20th Iranian Conf. Electr. Eng. (ICEE)*. Tehran, Iran: IEEE, May 2012, pp. 711–715.
- [50] M. Khajezadeh, M. R. Taha, A. El-Shafie, and M. Eslami, "A modified gravitational search algorithm for slope stability analysis," *Eng. Appl. Artif. Intell.*, vol. 25, no. 8, pp. 1589–1597, Dec. 2012, doi: [10.1016/j.engappai.2012.01.011](https://doi.org/10.1016/j.engappai.2012.01.011).
- [51] C. Li, H. Li, and P. Kou, "Piecewise function based gravitational search algorithm and its application on parameter identification of AVR system," *Neurocomputing*, vol. 124, pp. 139–148, Jan. 2014, doi: [10.1016/j.neucom.2013.07.018](https://doi.org/10.1016/j.neucom.2013.07.018).

- [52] R.-E. Precup, R.-C. David, E. M. Petriu, S. Preitl, and M.-B. Radac, "Novel adaptive gravitational search algorithm for fuzzy controlled servo systems," *IEEE Trans. Ind. Informat.*, vol. 8, no. 4, pp. 791–800, Nov. 2012, doi: [10.1109/TII.2012.2205393](https://doi.org/10.1109/TII.2012.2205393).
- [53] F.-S. Saeidi-Khabisi and E. Rashedi, "Fuzzy gravitational search algorithm," in *Proc. 2nd Int. eConference Comput. Knowl. Eng. (ICCKE)*. Mashhad, Iran: IEEE, Oct. 2012, pp. 156–160.
- [54] R. Storn and K. Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, pp. 341–359, Dec. 1997, doi: [10.1023/A:1008202821328](https://doi.org/10.1023/A:1008202821328).
- [55] Y.-C. Ho and D. L. Pepyne, "Simple explanation of the no free lunch theorem of optimization," *Cybern. Syst. Anal.*, vol. 38, no. 2, pp. 292–298, 2002, doi: [10.1023/A:1016355715164](https://doi.org/10.1023/A:1016355715164).
- [56] R. Zardashti, A. A. Nikkhah, and M. J. Yazdanpanah, "Constrained optimal terrain following/threat avoidance trajectory planning using network flow," *Aeronaut. J.*, vol. 118, no. 1203, pp. 523–539, May 2014, doi: [10.1017/S0001924000009349](https://doi.org/10.1017/S0001924000009349).
- [57] R. Zardashti, A. Nikkhah, and M. Yazdanpanah, "Multi-objective trajectory planning over terrain using label-setting greedy-based algorithm," *Proc. Inst. Mech. Eng. G, J. Aerosp. Eng.*, vol. 229, no. 8, pp. 1435–1453, Jun. 2015, doi: [10.1177/0954410014552089](https://doi.org/10.1177/0954410014552089).
- [58] R. Zardashti, M. Yazdanpanah, and A. Nikkhah, "Nonlinear multiobjective time-dependent TF/TA trajectory planning using a network flow-based algorithm," *J. Aerosp. Eng.*, vol. 29, no. 2, 2016, Art. no. 04015041, doi: [10.1061/\(ASCE\)AS.1943-5525.0000527](https://doi.org/10.1061/(ASCE)AS.1943-5525.0000527).
- [59] S. Hayat, E. Yanmaz, C. Bettstetter, and T. X. Brown, "Multi-objective drone path planning for search and rescue with quality-of-service requirements," *Auto. Robots*, vol. 44, no. 7, pp. 1183–1198, Sep. 2020, doi: [10.1007/s10514-020-09926-9](https://doi.org/10.1007/s10514-020-09926-9).



**SHUAI JIANG** received the B.Sc., M.Sc., and Ph.D. degrees from the China University of Petroleum (East China), Qingdao, China, in 2011, 2014, and 2019, respectively.

He is currently an Engineer with Shengli Xinda New Material Company Ltd., Dongying, China. His research interests include artificial intelligence, data mining, and geosciences.



**HONGGEN XU** received the B.S. and Ph.D. degrees in photogrammetric engineering and remote sensing from Wuhan University, Wuhan, China, in 2003 and 2008, respectively.

He is currently a Senior Engineer with the Wuhan Center of China Geological Survey (Central South China Innovation Center for Geosciences), China Geological Survey. His research interests include machine learning, remote sensing, data processing, and application of hyperspectral remote sensing.



**AIZHU ZHANG** (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from the China University of Petroleum (East China), Qingdao, China, in 2011, 2014, and 2017, respectively.

She is currently a Lecturer with the China University of Petroleum (East China). Her research interests include artificial intelligence, pattern recognition, city remote sensing, and wetland remote sensing.

...