Received June 28, 2021, accepted July 14, 2021, date of publication July 26, 2021, date of current version August 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3099836

3D Trajectory Planning of UAV Based on DPGA

JIN ZIRONG, ZHANG LIANG, AND ZOU ZHILONG

Department of Mathematics, School of Science, Wuhan University of Technology, Wuhan 430070, China

Corresponding author: Zhang Liang (thanleon@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61573012.

ABSTRACT The goal of trajectory planning is to shorten the flight distance as much as possible on the premise of ensuring the safety of UAV in flight. Therefore, the research of trajectory planning has broad prospects and great significance. As the key technology of trajectory planning, optimization algorithm has increasingly become one of the focuses of scholars at home and abroad. The dynamic programming algorithm is characterized by high computational efficiency and global optimization in trajectory planning. In 3D trajectory planning, as the spatial search space expands, the number of grid points increases faster, and time complexity of the dynamic programming algorithm is $O(n^3)$. It often leads to a "Curse of Dimension" phenomenon, which lowers its computational efficiency drastically. To solve this problem, this paper divides the entire planning space into stages based on Bellman's optimality principle. A dynamic programming-genetic algorithm(DPGA) is proposed by using genetic algorithm(GA) in each stage for optimization, while using dynamic programming algorithm(DP) in global planning. The global optimization ability of the algorithm is verified through convergence analysis. Moreover, based on a series of simulation experiments, it shows that the improved algorithm proposed in this paper is more efficient than the dynamic programming algorithm and genetic algorithm alone in global optimization.

INDEX TERMS Trajectory planning, dynamic programming algorithm, genetic algorithm.

I. INTRODUCTION

Path planning is one of the core issues of UAV control theory [1]. Its goal is to plan a optimal trajectory from the starting point to the terminate point according to the current complex mission environment, meeting the shortest flying range, successful obstacle avoidance, and various physical maneuverability constraints of the aircraft conditions. As a multi-objective optimization problem, trajectory planning involves multiple conditions such as obstacle terrain and the performance of the UAV itself, which often leads to difficulties in modeling and computing.

Unitil now many algorithms for trajectory plannying have been developed, among which two categories may be divided: exact algorithms and heuristic algorithms. Exact algorithms include A^* algorithms [2], sparse A^* algorithms, D^* algorithms, Dijkstra algorithms [3] and dynamic programming algorithms [4], etc. This type of trajectory planning algorithm is suitable for relatively simple models in two-dimensional space. While in 3D dimensional space, when the spatial planning space expands, the time for solution in tradictional algorithms may grow explosively. It means that the exact algorithms are difficult for complex 3D environment [5]. Heuristic algorithms such as the ant colony algorithms [6]–[9], genetic algorithms [10], particle swarm algorithms [11], [12], and simulated annealing algorithms have been widely used in solving problems of trajectory planning. The swarm intelligence algorithm avoids the "Curse of Dimension" phenomenon to a certain extent, but it is prone to fall into the local optimum. Furthermore, due to its insufficiency in global optimization and long solution time, it cannot meet requirements for engineering applications.

In recent years, there are works in literatures combining exact algorithms with heuristic algorithms. For example, in [13], the genetic algorithm mechanism was introduced into the dynamic programming algorithm to obtain remarkable effects in the application of reservoir optimization scheduling and real-time coordination scheduling of hybrid energy storage systems. For 3D trajectory planning, a series of hybrid algorithms based on simulating the annealing-ant

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

colony algorithm (SA-ACO) [14], the annealing-neural network algorithm (SA-CHNN) [15] and the annealing particle swarm algorithm (SA-PSO) [16] have achieved good results in solution, thereby providing some new methods for complex 3D trajectory planning problems. This paper develope a new hybrid algorithm, combining an exact algorithm and a heuristic algorithm, to build a new hybrid algorithm with high computational efficiency and good global convergence capability [17]. The new hybrid algorithm is called dynamic programming-genetic algorithm, which integrates the dynamic programming algorithm with global search capability and the genetic algorithm with efficient operation efficiency.

The essential idea of the new hybrid algorithm is to divide the spatial planning space into q stages based upon the Bellman optimality principle. There are several multi-objective decision-making subproblems in each stage, and each subproblem is solved by a hybrid coding multi-objective genetic algorithm. In terms of time complexity, the new algorithm not only achieves dimensionality reduction through the division of stages, but also uses the mechanism of genetic algorithm to avoid the ''Curse of Dimension'' problem caused by the spatial combination of discrete states. The main body of the hybrid algorithm is a dynamic programming algorithm, which has the ability of global optimization.

The paper is arranged as follows. In Section II and III, the navigational environment model and mathematical optimization model are established, respectively; In Section IV and V, the DPGA model is built, and a series of simulation experiments are carried out to verify the feasibility and advantages of the new algorithm in both theoretical and experimental aspects. In Sectioin VI and VII, the results are summarized, and the future research direction and work are prospected.

II. NAVIGATIONAL ENVIRONMENT MODEL

The establishment of a digital map composed of terrain information and threat information is a precondition for the path planning of UAV. Complex terrain and threat areas often show continuous and irregular characteristics.

A. ORIGINAL DIGITAL TERRAIN SIMULATION

Various digital maps are proposed for simulations. Among which [18] proposed a mathematical model based on the function method to simulate the original terrain:

$$\begin{aligned} z_1(x, y) &= \sin(x + a) + b \cdot \sin(x) \\ &\quad + c \cdot \cos(d \cdot \sqrt{x^2 + y^2}) + e \cdot \cos(y) \\ &\quad + f \cdot \sin(f \cdot \sqrt{x^2 + y^2}) + g \cdot \cos(y). \end{aligned} \tag{1}$$

where x and y represent the horizontal coordinates of the original terrain; z is the height of the terrain; a, b, c, d, e, f, g are topographic and geomorphic parameter. By the original topographic information, adjusting the value of the corresponding parameter can simulate a variety of rugged landforms.

TABLE 1. Digital map parameters.

Parameter	Value	Parameter	Value
a	5.0	e	1.0
b	0.2	f	0.1
c	0.1	g	0.1
d	0.3	h	[2, 3, 4.5, 1.5]
x_0	[25, 40, 60, 80]	y_0	[35, 80, 56, 20]
x_1	[10, 9, 12, 10]	y_1	[8, 9, 10, 10]

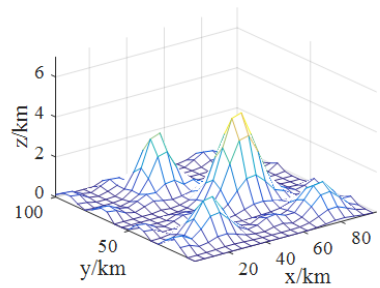

FIGURE 1. Digital terrain simulation map.

B. THREAT AREA SIMULATION

There are threat areas such as mountain peaks, no-fly zone or buildings in the navigation area. For such threat areas, this paper uses the mountain peak function, reprented as

$$z_2(x, y) = \sum_{i=1}^{n} h_i e^{-\frac{(x - x_0(i))^2}{x_1(i)^2} - \frac{(y - y_0(i))^2}{y_1(i)^2}}. \tag{2}$$

In formula (2), x and y represent the coordinates on the horizontal projection; z is the corresponding terrain height; n is the number of peaks; i denotes the i-th peak; h_i is the height of the mountain peak; $x_0(i)$ and $y_0(i)$ are horizontal coordinates of the apex of the i-th peak; $x_1(i)$ and $y_1(i)$ are terrain contour parameter, which corresponds to the slope of the i-th peak on the X-axis and Y-axis respectively.

C. EQUIVALENT DIGITAL MAP

The digital map information fusion processing technology is used to superimpose and fuse the mathematical models of terrain and landforms and threat areas to build an equivalent digital topographic map covering comprehensive planning space information [19], which has the form

$$z(x, y) = \max\{z_1(x, y), z_2(x, y)\}. \tag{3}$$

The digital map model consists of formula (1-3). As shown in Figure 1, we use the digital map model to simulate the complex 3D terrain environment. The scale of the terrain environment is 100 × 100 (km^2), and there are four threat areas. In addition, the values of various terrain parameters of the digital map model are shown in Table 1.

III. MATHEMATICAL MODEL OF TRAJECTORY PLANNING

The essence of the trajectory planning problem is a multi-objective optimization problem, mainly composed of two parts: objective function and constraint conditions.

1. Constraint conditions. By the physical limitations of the drone itself and the requirements of navigation accuracy, the continuous route curve often does not meet the flight requirements, so a set of ordered points $C = \{X_i|X_i = (x_i, y_i, z_i), i = 1, 2, \ldots, n\}$ is used to represent the trajectory curve. In addition, during the flight of the drone, It also needs to meet certain trajectory constraints, such as minimum trajectory segment length, maximum turning angle, maximum climb/dive angle, maximum trajectory length, minimum flight altitude, and obstacle avoidance altitude [20]–[22].

2. Objective function. The trajectory cost function is a concrete manifestation of trajectory evaluation indicator to measure the pros and cons of the planned trajectory. The objective function used in this paper mainly considers two aspects of navigation and flight altitude.

In summary, the optimization model of trjactory planning is shown as

$$\min f = \sum_{i=1}^{n-1} (\omega_1 L_i^2(X_i) + \omega_2 H_i(X_i))$$
$$s.t. \begin{cases} L_i(X_i) \geqslant L_{\min}, & i = 1, 2, \ldots, n-1, \\ \theta_i \leqslant \theta_1, & i = 1, 2, \ldots, n-2, \\ \alpha_i \leqslant \theta_2, & i = 1, 2, \ldots, n, \\ z_i > z(x_i, y_i), & i = 1, 2, \ldots, n, \\ z_i > H_{\min}, & i = 1, 2, \ldots, n. \end{cases} \tag{4}$$

In formula (4), f is the objective function, $L_i(X_i)$ is the distance between the trajectory node X_i and X_{i+1}; $H_i(X_i)$ is the average height of the trajectory section between X_i and X_{i+1}; ω_1 and ω_2 are the weight coefficient; θ_i and α_i represent the steering angle and dive angle at the route node X_i; $L_{\min}$, θ_1 and θ_2 are minimum trajectory segment length, maximum turning angle, maximum climb/dive angle, and minimum flight altitude; z_i Is the route node X_i, and the flight height, $z(x_i, y_i)$ is the height of the obstacle in the analogue-digital terrain at that position.

IV. TRAJECTORY PLANNING BASED ON DPGA

A. PRINCIPLES AND DEFECTS OF DYNAMIC PROGRAMMING ALGORITHM AND GENETIC ALGORITHM

1) DYNAMIC PROGRAMMING ALGORITHM

The classical dynamic programming is to transform a multi-stage process into a series of single-stage problems, use the relationships between the stages to solve one by one, and then find out solution to the optimization problem [23].

In this paper, the trajectory-planning problem is transformed into a mathematical optimization model at first [24]. Assume that the planning process is divided into n stages, with the first stage state set of the aircraft starting point S, the final stage state set of the target point T. The grid nodes in the k-th stage are the set of feasible solution points in this stage

$$S_k = \{(x_k, y_i, z_j)|i, j = 1, 2, \ldots, n\} \subset \Omega, \tag{5}$$

where the state variable is denoted as $s_k \in S_k$. The set of allowed decisions on the *k-th* stage is defined as

$$D_k = \{(h_x, i \cdot h_y, i \cdot h_z)| - \lambda_{\max} \leq i \leq \lambda_{\max}\}. \tag{6}$$

The aforementioned h_x, h_y and h_z are the step size of the grid division along the coordinate axis in the discrete planning space; $\lambda_{\max}$ represents the maximum perturbation. The decision variable is recorded as u_k, and $u_{k\in D_k}$. Define the state transition equation in the *k-th* stage as $T_k(s_k, u_k) = s_k + u_k$. Define the Euclidean distance between the initial state s_k of the *k-th* stage and the initial state s_{k-1} of the $(k-1)$-*th* stage as the stage fitness function of the *k-th* stage, that is $v_k(s_k, u_k) = d(s_k, s_{k-1})$. The Superimposing of the adaptation values for each stage equals the optimization index of the whole process, i.e.,

$$V_{1,n} = \sum_{k=1}^{n} v_k(s_k, u_k). \tag{7}$$

Obviously, $V_{1,n}$ satisfies the optimality principle. Therefore, as shown in equation 8, the optimization function of each stage can be defined as the optimal solution from all feasible points to the starting point.

$$\begin{cases} f_k(s_k) = \min\limits_{u_k \in D_k} \{v_k(s_k, u_k) + f_{k-1}(s_{k-1})\}, \\ f_1(s_1) = 0, \end{cases} \tag{8}$$

where $f_k(s_k)$ is the optimal function value from the *k-th* stage s_k to the starting point, and $f_1(s_1) = 0$ is the boundary condition of the dynamic programming algorithm.

The solution of the dynamic programming algorithm needs to traverse the entire search space, which has the global optimization capability. However, this algorithm may cause severe drawbacks, In 3D dimensional cases, the traversed point set is $\sum_{k=1}^{n} S_k$, which contains n^3 points, so the time complexity of dynamic programming algorithm is $O(n^3)$ [25], [26]. When the number of discrete space grid points increases, the time loss will explode, which is the "Curse of Dimension" problem.

2) GENETIC ALGORITHM

Genetic algorithm [27], [28] is a search algorithm based on the mechanism of natural selection and population genetics. It simulates the phenomenon of selection, crossover and mutation in the process of natural evolution and genetics. In the genetic algorithm model of trajectory planning, any route between the starting point S and the end T is a gene individual, and M individuals constitute a population. At the beginning of genetic algorithm, a group of initial individuals are randomly generated. Then the fitness value of each individual is calculated by the objective function in formula 4. According to the fitness value, some individuals are selected to produce the next generation. The selection operation reflects the principle of "survival of the fittest". The individuals with high fitness value produce the next generation, and the individuals with low fitness value are eliminated. The selected individuals are recombined by crossover and mutation operators to produce a new generation of individuals. The new generation of individuals inherited the excellent characteristics of the previous generation, and gradually evolved to the optimal solution. Therefore, genetic algorithm can be regarded as the evolution process of a group of initial feasible solutions.

As a kind of swarm intelligence algorithm, the genetic algorithm has the following advantages in trajectory planning: (1) High efficiency: Because genetic algorithm has probability mechanism, its operation efficiency is less affected by the size of search area; (2) Wide application: the crossover and mutation operations of the two positions of X_i and X_{i+1} in the curve C are independent of each other, so we can split and reconstruct the individual, and combine with different algorithms; (3) Parallelism: the entire search process is based on populations, which can compare and select multiple individuals at the same time, and perform parallel calculations to improve efficiency. Due to the iterative search mechanism of genetic algorithms, there are also many defects. The most severe shortcomings are that the algorithm has a specific dependence on the initial population, and it is easy to fall into the local optimum. In addition, the crossover operator and mutation operator directly affect the pros and cons of the results, but the values of these parameters often rely on the human experience.

B. THE PRINCIPLE OF DYNAMIC PROGRAMMING-GENETIC ALGORITHM

For the solution of the trajectory planning problem, both dynamic programming algorithm and genetic algorithm have their advantages, but there are also shortcomings. Therefore, this paper proposes a dynamic programming-genetic algorithm; By dividing the search area into several stages, the whole trajectory planning problem is divided into a series of subproblems with nodes on both sides of the plane as boundary conditions; The new algorithm uses genetic algorithm to solve the optimal solution of each subproblem and uses dynamic programming algorithm to globally optimize the optimal solution of each subproblem in order of stages. Compared with the single use of dynamic programming algorithm and genetic algorithm, the hybrid algorithm changes the problem scale from a trajectory with n nodes to a series of trajectory segments with p nodes, which reduces the time complexity of dynamic programming algorithm and avoids the "Curse of Dimension" caused by the increase of search area nodes. Moreover, the complementary mechanism of dynamic programming algorithm and genetic algorithm also eliminates the dependence of genetic algorithm on the initial population and solves the problem of easily falling into local optimum.

The principle of optimality requires that the optimal trajectory decision has the following properties: no matter how to choose the initial state and the initial decision, for any node of the optimal trajectory, the decision route from this node point to the starting point must constitute the optimal strategy [29]. Assuming that the trajectory planning problem is decomposed into q subproblems with the same properties, the past and the remaining trajectorys are described by the calculated trajectory cost and the current state respectively. According to the separability of the objectives of the dynamic programming problem, the current state and the calculated trajectory cost must constitute a multi-objective non inferior solution.

In order to describe the state of trajectory-planning, this paper proposes the definition of the best trajectory state as follows.

Definition 1: The grid points that meet the following two conditions are called the optimal trajectory status of trajectory planning:

(1) The line between the current node and the target point should be in the safe area as far as possible.

(2) The distance between the current node and the target is the shortest.

Thus, the best trajectory state evaluation function is written as

$$M_{i,j} = \frac{\sqrt{(x - x_\tau)^2 + (y - y_\tau)^2 + (z - z_\tau)^2}}{L(s_{i,j}, T)}, \tag{9}$$

where (x_τ, y_τ, z_τ) is the coordinate of the aim point T; $L(s_{i,j}, T)$ is the length of the trajectory that is not in the obstacle on the line connecting the current candidate node and the target point.

Because the terrain data is stored in a matrix form, it is difficult to specifically reflect the length relationship defined by $L(s_{i,j}, T)$ in formula (9). To facilitate the calculation, the formula (9) is improved to the following form:

$$M_{i,j} = \frac{\sqrt{(x - x_\tau)^2 + (y - y_\tau)^2 + (z - z_\tau)^2}}{N(s_{i,j}, T)}, \tag{10}$$

where $N(s_{i,j}, T)$ is the number of grid points on the line between the optional node and the target point that are not within the obstacle. To simplify the description, a 2D contour map is taken as an example of the analysis. The explicit calculation principle is shown in Figure 2. As shown in Figure 2, the intersection point of the dotted line and the straight line l in the figure is the unreachable state point, and the intersection point of the solid line and the straight line l is the reachable state point. $N(s_{i,j}, T)$ is the number of unreachable state points on the line segment l between the current node $s_{i,j}$ and the target point T.

If the trajectory planning problem requires planning the optimal trajectory with n trajectory nodes, this problem is decomposed into q stages by node division. The number of

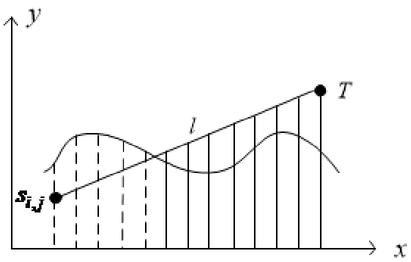

FIGURE 2. Schematic diagram of best trajectory node calculation.

trajectory nodes included in stage $i(1 \leq i \leq q)$ is

$$n_i = \begin{cases} \left[\dfrac{n}{q}\right], & i = 1, 2, \ldots, q-1, \\ N - \left[\dfrac{n}{q}\right] \cdot (q-1), & i = q. \end{cases} \tag{11}$$

In trajectory planning, genetic algorithm is used to plan *i-th* stage track segment, and formula (10) is used to evaluate the fitness of the track segment. The principle of DPGA is shown in Figure 3.

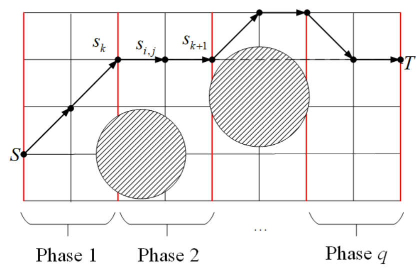

FIGURE 3. Schematic diagram of dynamic programming-genetic algorithm.

In Figure 3, the red plane S_k divides the search area into q stages along the x-direction. Take the k-th stage point set S_k searching to the $(k+1)$-th stage point set S_{k+1} as an example, for $\forall s_k \in S_k$, traverse the decision set D_k of s_k. For $\forall d_k \in D_k$, $s_{k+1} = s_k + d_k$, we use genetic algorithm to plan the optimal route l_k between s_k and s_{k+1}, and the value of the objective function of the trajectory segment l_k is $v_k(s_k, s_{k+1})$, then the function value of s_{k+1} to the starting point S through s_k is f_{k+1}; in the process of traversing S_k, the minimum of f_{k+1} is the value of the objective function from s_{k+1} to the starting point S. In the process of solving the optimal trajectory between s_k and s_{k+1} by genetic algorithm, the evaluation function of the planned trajectory segment l_k is $(f_{l,i}, \sum_{j=1}^{p} M_{i,j})$. $M_{i,j}$ is the state value corresponding to the j-th trajectory node s_{ij} in the i-th stage calculated according to formula 10. $f_{l,i}$ is the objective function value of each node in the trajectory L, and its form is show as

$$f_{l,i} = \sum_{j=1}^{i-1} (\omega_1 \cdot L_j^2(X_{i+1}) + \omega_2 \cdot H_j(X_{i+1})). \tag{12}$$

where, L_j is the length of the *j-th* stage, taking the distance between the two trajectory nodes; H_j is the height of the *j-th* stage, taking the average height of the trajectory nodes. Then it follows that $f_{l,i}$ and $M_{i,j}$ must constitute a set of non-inferior solutions. Conversely, if a strategy is inferior to the non-inferior solution, it must not be an optimal strategy. Therefore, if $f_{l,i}$ and $M_{i,j}$ corresponding to a grid point are a set of non-inferior solutions, then this grid point may be the best trajectory node.

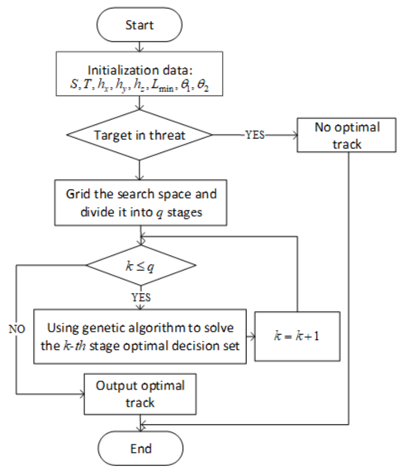

FIGURE 4. Dynamic programming-genetic algorithm flowchart.

The dynamic programming-genetic algorithm flowchart is shown in Figure 4. Figure 4 shows the fusion mechanism of dynamic programming algorithm and genetic algorithm. In addition, in the table of algorithm 1, we explain the details of the operation process of the algorithm in the form of pseudo code.

C. ALGORITHM CONVERGENCE ANALYSIS

The global convergence of genetic algorithm are shown in the following Theorem 1 and Theorem 2.

Theorem 1: If the genetic algorithm has the operation of retaining the optimal individual, it must converge to the global optimal solution.

The proof can be found in the literature [30].

Theorem 2: Suppose that $h(i) = \{x_1, x_2, \ldots, x_N\}$ is the *i-th* generation population of genetic algorithm, N is the total number of individuals in the population, $Z_{h(i)} = \max\{f(x_k)|k = 1, 2, \ldots, n\}$ is the maximum fitness of the current population, $f(x)$ is the fitness function. The global maximum fitness is expressed as $f^* = \max\{f(x)|x \in S\}$, S is the set of individuals of all populations. Then if and only if

$$p^* = \lim_{N \to \infty} \{Z_{h(i)} = f^*\} = 1,$$

the genetic algorithm converges to the global optimal solution.

Proof: If the optimal path with n nodes is planned, then the number of alternative decisions for each trajectory node is v. Both genetic algorithm and dynamic programming-genetic algorithm calculate the objective function $N \cdot M$ times, where N is the maximum number of

Algorithm 1 Dynamic Programming-Genetic Algorithm

Input:
Digital map model $z(x, y)$, start point S, target point T, and constraint parameter.

Output:
The optimal route consisting of a set of n-dimensional point series.

Initialize:
Set the value of parameter q, and divide the planning space according to formula 11.

function DPGA($z(x, y), S, T$)
  $p \leftarrow \frac{n}{q}, f_1(s_1) \leftarrow 0$
  for $i = 1 \rightarrow q$ **do**
    According to formula 5, $k \leftarrow i \cdot p$, the node set of stage i is $Point_k = S_k$
    for $\forall s_k \in S_k$ **do**
      for For the decision set, $\forall d_k \in D_k$ **do**
        if d_k satisfies the constraints **then**
          trajectory point row L_i,
          Objective function value v_i
          $s_{k+1} = s_k + d_k$
          $[L_i, v_i] = \mathrm{GA}(z(x, y), s_k, s_{k+1})$
          $f_{i+1}(s_{k+1}) \leftarrow$ equation 8
        end if
      end for
    end for
  end for
  The optimal value of the objective function is $f_q(s_{n+1})$
  $L_q \leftarrow l_q$
  for $i = q - 1 \rightarrow 1$ **do**
    if The end of l_i is the beginning of l_{i+1} **then**
      $L_i \leftarrow l_i$
    end if
  end for
  $L = \sum L_i$
  return $[L, f_q(s_{n+1})]$
end function
function GA($z(x, y), s_k, s_{k+1}$)
  Crossover probability $\leftarrow p_c$
  Mutation probability $\leftarrow p_m$
  Initialize the first generation population P_1
  Calculate the fitness function of each individual
  for $i = 1 \rightarrow N$ **do**
    if $rank < p_c$ **then**
      Cross operation
    end if
    if $rank < p_m$ **then**
      Mutation operation
    end if
    Choose operation to retain better individuals
  end for
  The optimal individual of the N-th generation population P_n^* and its fitness function value f_n
  return $[P_n^*, f_n]$
end function

iterations, and M is the number of populations. The convergence rate of genetic algorithm is k ($k > 1$) times that of pure random search. Therefore, after N times of iterative calculation, the probability p^* [31] of obtaining the global optimal trajectory by genetic algorithm is

$$p^* = \frac{N \cdot M \cdot k}{v^n}. \tag{13}$$

Because the DPGA algorithm needs to divide the track with n nodes into q track segments and each trajectory segment can be regarded as a subproblem, the number of nodes of each individual in the genetic algorithm becomes $p = \frac{n}{q}$. Since the probability of convergence of the optimal solution to each sub-problem is independent of each other, the probability of convergence of the global optimal solution using DPGA algorithm is

$$P_{DPGA} = \frac{(N \cdot M \cdot k)^q}{v^n}. \tag{14}$$

It is known that

$$k > 1, N \geq 1, M \geq 1, q > 1, \lim_{T \to \infty} p^* = 1,$$

therefore, $(N \cdot M \cdot k)^q > N \cdot M \cdot k$, $1 > P_{DPGA} > p^*$,

$$\lim_{T \to \infty} P_{DPGA} = 1. \tag{15}$$

The operation process of the dynamic programming-genetic algorithm proposed in this paper can be regarded as using the dynamic programming algorithm to optimize the trajectory segment planned by the genetic algorithm. The dynamic programming algorithm is a global traversal algorithm, which has an excellent global optimization capability. Therefore, the convergence of dynamic programming-genetic algorithm mainly depends on the genetic algorithm for trajectory segment planning. The genetic algorithm used in this paper introduces non-inferior solution $(f_{l,i}, \sum_{j=1}^{p} M_{i,j})$ to screen the population, and satisfies the condition of retaining good individuals in theorem 1, so the algorithm can converge to the optimal global solution; In addition, according to the proof conclusion of formula 15, the hybrid algorithm satisfies the convergence condition of theorem 2, and the convergence efficiency is better than the genetic algorithm.

D. ALGORITHM PERFORMANCE ANALYSIS

When the number of trajectory nodes is n, the time complexity of DP algorithm and GA algorithm are

$$\begin{aligned} T_{DP}(n) &= O(P(n)) = O(n^3), \\ T_{GA}(n) &= O(M \cdot N \cdot n). \end{aligned} \tag{16}$$

DPGA decomposes the entire trajectory-planning problem with n nodes into q sub-problem by DP algorithm, therefore, its time complexity is

$$\begin{aligned} T_{DPGA}(n) &= T_{DP}(q) \cdot T_{GA}(\frac{n}{q}) \\ &= O(q^2 \cdot N \cdot M \cdot n). \end{aligned} \tag{17}$$

Compared with DP algorithm, DPGA algorithm significantly reduces the time complexity, avoids the "Curse of Dimension" problem caused by the increase in planning space to some extent, and improves the performance of the algorithm. Compared with GA algorithm, the time complexity of DPGA algorithm is the same as that of GA algorithm, but the convergence probability P_{DPGA} and convergence efficiency of DPGA algorithm are greater than that of GA algorithm. In addition, by considering the influence of the number of stages q on algorithm performance, DPGA algorith educed to N_1 and M_1 as shown in formula 18:

$$N_1 \cdot M_1 = \frac{N \cdot M}{q^2}. \tag{18}$$

The performance of the adjusted DPGA is basically the same as that of the GA. Supposing the convergence expectation of the DPGA algorithm is expressed as a function $f(n, q)$ related to n and q, the boundary conditions can be obtained by comparing and analyzing the time complexity of the three algorithms. When $q \to 1$, the DPGA algorithm degenerates to a genetic algorithm; When $q \to n$, the DPGA algorithm degenerates to a dynamic programming algorithm. Therefore, the existence of the optimal number of stages q that makes the algorithm the most efficient can be proved. The proof process is as formula 19:

$$\begin{aligned}
&\textbf{proof}: \because f(n,q) = \frac{1}{v^n} \cdot (\frac{N \cdot M \cdot k}{q^2})^q,\\
&\therefore \lim_{q \to 1} \frac{\partial f(n,q)}{\partial q} > 0,\\
&\lim_{q \to n} \frac{\partial f(n,q)}{\partial q} < 0,\\
&\therefore \exists q_\zeta \in (1, n) \text{ satisfies}\\
&\lim_{q \to q_\zeta} \frac{\partial^2 f(n,q)}{\partial q^2} = 0.
\end{aligned} \tag{19}$$

There is an optimal number of stages q_ζ such that

$$\lim_{q \to q_\zeta} f(n,q) = \max f(n,q), \ q \in (1, n). \tag{20}$$

In addition, due to the boundary condition

$$\begin{cases} \lim\limits_{q \to 1} f(n,q) = f_{GA}(n), \\ \lim\limits_{q \to n} f(n,q) = f_{DP}(n), \end{cases} \tag{21}$$

we can verify that: when $q \in (1, n)$, the convergence expectation of hybrid algorithm is better than that of genetic algorithm and dynamic programming algorithm.

V. OPTIMIZATION RESULTS AND ANALYSIS

Under the hardware environment of Intel (R) Core (TM) i7-8700 CPU@3.20 GHz and 8GB of memory. The experiment is divided into three parts: algorithm performance analysis, the number of optimal stages of the algorithm analysis and trajectory planning simulation.

A. EXPERIMENT 1: PERFORMANCE ANALYSIS OF DP ALGORITHM AND GA ALGORITHM UNDER DIFFERENT TRAJECTORY NODES

By changing the number of trajectory nodes n, we compare the planning time of DP algorithm and GA algorithm and the cost function f (coefficient parameters ω_1 and ω_2 are 0.2 and 0.8) in formula 2 to verify the impact of planning space size on the optimization performance of the algorithm. The results are shown in Table 2.

TABLE 2. Performance comparison of DP and GA under different trajectory node number.

n	DP		GA	
	Value	Time/s	Value	Time/s
10	32.99	2.71	32.99	2.53
	32.99	3.24	36.18	3.15
20	37.51	12.09	45.76	4.17
	37.51	11.68	39.89	3.94

By analyzing the data onto Table 2, the following conclusions can be brought:

1. With the increase in the number of trajectory nodes n, the time loss of the DP algorithm will increase explosively and the running time of the GA algorithm increases less, which verifies the problem that the efficiency of the DP algorithm greatly decreases when the number of discrete space grid points increases.

2. Comparing the fitness function values of the optimal trajectory under $n = 10$ and $n = 20$, the fitness function value of DP algorithm is constant, which is generally better than genetic algorithm; The adaptive function value of GA algorithm will have extreme results. It is proved that GA algorithm is easy to fall into local optimum and its global optimization ability is weaker than DP algorithm.

B. EXPERIMENT 2: ANALYSIS OF THE OPTIMAL NUMBER OF STAGES

The dynamic planning-genetic algorithm for trajectory planning divides the entire optimization problem into several stages at first, then use the GA algorithm to solve them in stages, and finally obtain the solution to the entire trajectory-planning problem. In order to select the optimal number of stages, we take the stages separately with numbers of 1, 2, 3, 4 and 5 to perform 100 trajectory simulation experiments. The optimal solutions and the average running time of the algorithm are presented in Figure 5 and Figure 6, where Figure 5 is the probability of the optimal flight path with the number of stages, and Figure 6 is the algorithm running time.

By Figure 5 and Figure 6, the following conclusions can be drawn:

1. As the number of stages increases, the probability of the dynamic programming-genetic algorithm obtaining the optimal solution is increasing.

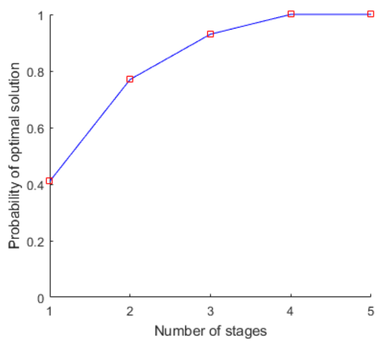

FIGURE 5. Optimal trajectory probability.

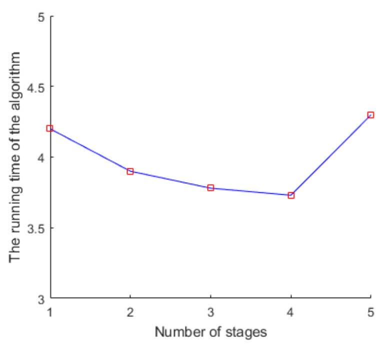

FIGURE 6. Algorithm time loss.

When the meshing step size and trajectory planning area are constant, the number of trajectory nodes is also invariant. As the number of stages increases, the number of nodes contained in each stage decreases. At this time, the probability of using the GA algorithm to calculate the optimal trajectory at each stage is increasing. The convergence probability of the GA algorithm is independent of each other. Therefore, the probability of the dynamic programming-genetic algorithm proposed for this paper converging on a global optimum is increasing.

2. As the number of phases increases, the time loss of the dynamic programming-genetic algorithm decreases first and then increases.

DPGA decomposes the planning problem into multiple stages and uses the relationship between two adjacent stages to solve the optimal trajectory of the next stage, and then finds the solution to the trajectory-planning problem. In this process, the hybrid algorithm can "remember" the best trajectory of the stage, so as to avoid repeated calculations and improve the speed of the algorithm. However, as the number of stages increases, the algorithm gradually degenerates into the DP algorithm, resulting in a significant reduction in the efficiency of the algorithm, and the running time will increase due to the iteration of the GA algorithm. Hence, as the number of stages increases, the running time of the dynamic programming-genetic algorithm decreases first and then increases.

In order to enable the dynamic planning-genetic algorithm proposed for this paper to obtain the optimal trajectory in a short time, the number of stages is taken to be 4.

C. EXPERIMENT 3: PERFORMANCE ANALYSIS OF DPGA ALGORITHM

In order to verify the performance of DPGA, dynamic planning algorithm, genetic algorithm and dynamic planning-genetic algorithm were used to conduct the trajectory planning simulation experiments. The parameters such as the starting point S, the target point T, and the maneuverability constraints are shown in Table 3. (units of length *km*.) The values of meshing step size h_x, h_y, h_z depend on the constraints of the minimum trajectory segment and the size of the threat zone. The value of maximum disturbance $y_{\max}, z_{\max}$ determines the size of the current node's backward search area, which needs to be considered in conjunction with various constraints and algorithm performance.

TABLE 3. 3D trajectory planning parameters.

Parameter name	Parameter value
Starting point S	(2.4, 5.4, 0)
Target point T	(99.4, 99.4, 0)
Minimum trajectory segment length $L_{\min}$	5
Meshing step size h_x, h_y, h_z	5, 5, 0.5
Maximum disturbance $y_{\max}, z_{\max}$	2, 2
Maximum turning angle θ_1	50°
Maximum climb angle θ_2	50°

The simulation results are shown in Figure 7. As shown in the first line is the DP algorithm, the second line is the GA algorithm, and the third line is the DPGA algorithm. The first column is a stereoscopic view of the three-dimensional trajectory. The second column is a projection of the three-dimensional trajectory of the *xoy* plane, and the third column is a projection of the three-dimensional trajectory of the *xoz* plane. The specific data of objective function value and time loss of simulation results in Figure 7 are shown in Table 4.

TABLE 4. Comparison of three algorithms' planned trajectory and running time.

Algorithm	Value	Time/s
DP algorithm	37.51	11.68
GA algorithm	39.89	3.94
DPGA algorithm	37.64	3.73

The DPGA algorithm is a combination of the DP algorithm and the GA algorithm. It inherits the advantages of the two algorithms and reduces the dimensionality of the trajectory-planning problem in terms of time by dividing the stages of DP algorithm. Natural selection and genetic mechanism of the GA algorithm are used to reduce spatial dimension. The experimental results in Figure 7 and the experimental data in Table 4 are analyzed to verify the following questions:

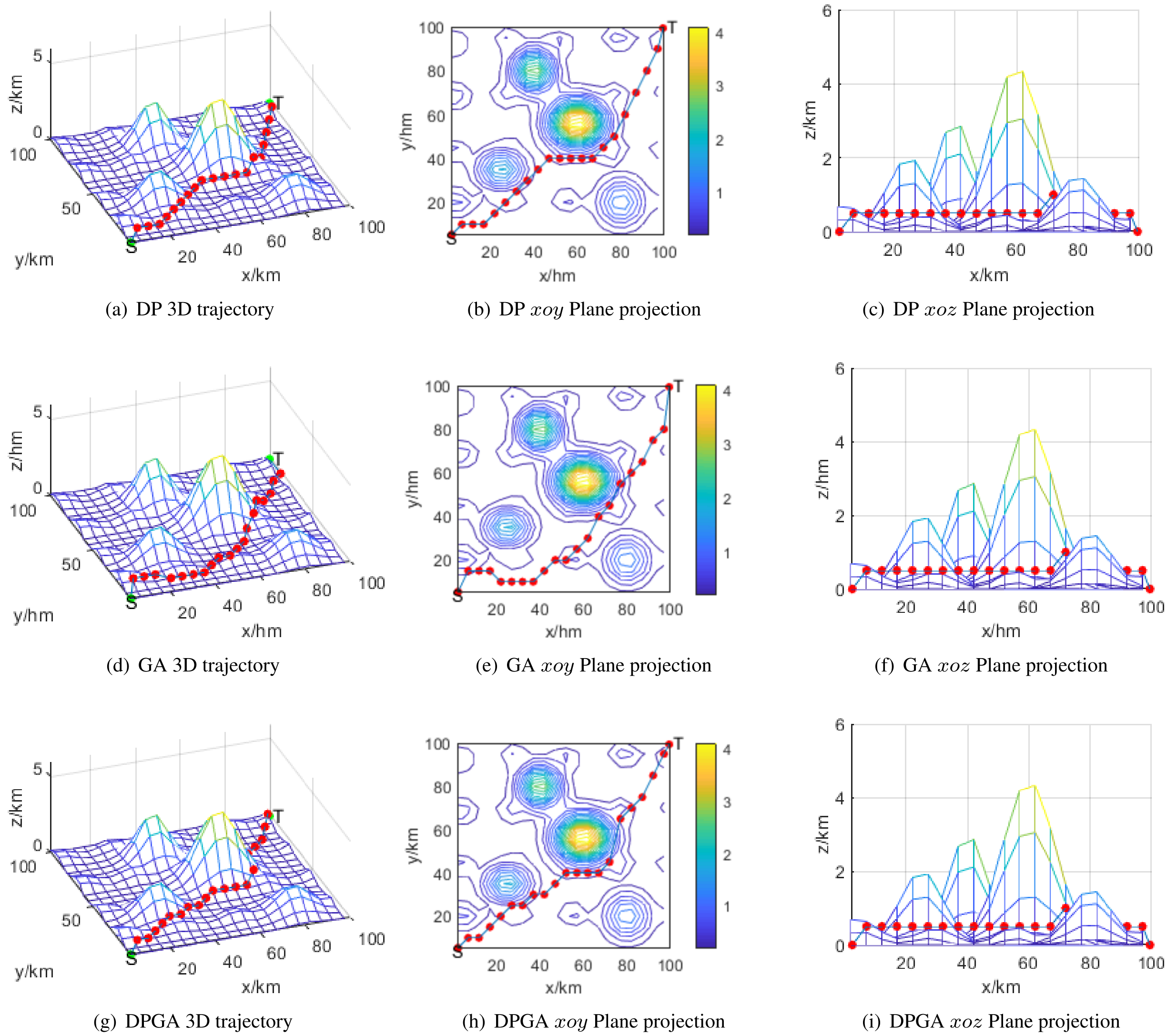

FIGURE 7. Simulation chart of each algorithm.

1. Combining Figure 7 and Table 4 shows that: the planned trajectory shown in Figure (d) of the GA algorithm (the value of the cost function is 30.89) is not an optimal trajectory compared with the planned trajectory of the DP algorithm (the value of the cost function is 37.51) shown in Figure (a). Because of the dependence of the initial population of genetic algorithm and the blindness of searching by probability, it is easy to fall into a local optimum. However, by observing Figure (e), Figure(f) and Table 4, although the genetic algorithm for trajectory-planning has the risk of falling into a local optimum, it can quickly plan a feasible trajectory, and the planning speed is almost $1/3$ of the DP algorithm. The main reason is that DP algorithm has a "Curse of Dimension" problem. When the number of spatial points increases, the algorithm running time increases exponentially. However, GA algorithm can improve the algorithm's convergence speed to a certain extent by natural selection and genetic mechanisms.

2. Compared with the trajectory planned by DP algorithm, the trajectory planned by DPGA algorithm has more inflection points. The reason is that the trajectory is divided into stages, and the GA algorithm is used to calculate the stage trajectory, which leads totrail nodes have a certain randomness. By Table 4, compared with the DP algorithm, the value of the trajectory cost function obtained by the DPGA algorithm is 37.64, which is extremely close to the optimal value of 37.51 obtained by the DP algorithm. Therefore, the algorithm can converge to a global optimal solution.

3. By comparing and analyzing Table 4, it can be seen that the cost function of the GA algorithm is 39.89, which takes 3.94s. The cost function value of the trajectory planned by the DPGA algorithm proposed for this paper is 37.64 and the time-consuming is 3.73s. This is because the number of nodes in the trajectory segment is reduced, which leads to the improvement of the efficiency of the genetic algorithm and the increase of the probability of convergence to the

global optimum. DPGA algorithm is used in the trajectory planning process, it can "remember" the best trajectory of the sought stage, thereby avoiding double counting. Therefore, compared with the GA algorithm, the DPGA algorithm converges faster and the resulting trajectory is shorter.

Through simulation experiments, several conclusions can be drawn. Compared with the other two algorithms, the DPGA algorithm can plan the optimal trajectory in a shorter time on the premise that all constraints are satisfied. Therefore, the DPGA algorithm proposed for this paper is more suitable for 3D trajectory planning.

VI. CONCLUSION

In this paper, a digital simulation of complex terrain environment is carried out by using mathematical functions, and the search area is discretized. Then, aiming at the "Curse of Dimension" problem of DP algorithm in 3D search area, the dynamic programming-genetic algorithm is proposed, and its feasibility is verified by theoretical derivation. In addition, this paper analyzes the optimality of the GA algorithm and DPGA from the perspective of probability, and verifies that the improved algorithm has better convergence; then the time complexity of the three algorithms is compared. Combined with the expected convergence time, the existence of the optimal number of stages is derived. Finally, this paper conducts three experiments on algorithm performance analysis, the number of optimal stages of the algorithm analysis and trajectory planning simulation, which confirmed the above relevant theories and conclusions, and also verified the advantages of the DPGA algorithm in solving complex 3D trajectory planning problems.

VII. FUTURE WORK

This paper propose a dynamic programming-genetic algorithm based on dynamic programming algorithm and genetic algorithm. Although this paper has achieved specific stage achievements, there are problems worthy of continuing to explore and improve. The future research work can be improved from the following two aspects:

1. Add a parallel computing system to the hybrid algorithm. It is mentioned in this paper that genetic algorithms have the advantage of parallelism. Parallel computing mechanisms can be introduced into the algorithm to improve the efficiency of the algorithm;

2. Establish a mathematical model of trajectory planning with variable step length. Most of the documents, including this article, use fixed-step sizes to divide the three coordinate axes. This division method is more conducive to establishing the model and the realization of the planning procedure. However, under certain conditions, the step size of trajectory planning is required to be adaptive or variable.

REFERENCES

[1] A. Trotta, F. D. Andreagiovanni, M. Di Felice, E. Natalizio, and K. R. Chowdhury, "When UAVs ride a bus: Towards energy-efficient city-scale video surveillance," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 1043–1051.

[2] X. Zhong, J. Tian, H. Hu, and X. Peng, "Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment," *J. Intell. Robot. Syst.*, vol. 99, no. 1, pp. 65–77, Jul. 2020.

[3] D. Baeza, C. F. Ihle, and J. M. Ortiz, "A comparison between ACO and Dijkstra algorithms for optimal ore concentrate pipeline routing," *J. Cleaner Prod.*, vol. 144, pp. 149–160, Feb. 2017.

[4] S. Ping, "Application of two multi-dimensional dynamic programming algorithms to optimal operation of cascade reservoirs," *J. Hydraulic Eng.*, vol. 45, no. 11, pp. 1327–1335, 2014.

[5] C. Xia, A. Yudi, and L. Hongli, "Research on three-dimensional path planning of UAV based on improved ant colony algorithm," *Tactical Missile Technol.*, vol. 1, no. 2, pp. 59–66, 2019.

[6] O. Montiel-Ross, R. Sepúlveda, O. Castillo, and P. Melin, "Ant colony test center for planning autonomous mobile robot navigation," *Comput. Appl. Eng. Educ.*, vol. 21, no. 2, pp. 214–229, Jun. 2013.

[7] J. Liu, J. Yang, H. Liu, X. Tian, and M. Gao, "An improved ant colony algorithm for robot path planning," *Soft Comput.*, vol. 21, no. 19, pp. 5829–5839, Oct. 2017.

[8] P. Xu, "Research on optimized model of travel route selection based on intelligent image information and ant colony algorithm," *Multimedia Tools Appl.*, vol. 4, pp. 1–17, Apr. 2019.

[9] A. Lazarowska, "Multi-criteria ACO-based algorithm for ship's trajectory planning," *Int. J. Mar. Navigat. Saf. Sea Transp.*, vol. 11, no. 1, pp. 31–36, 2017.

[10] K. P. Cheng, R. E. Mohan, N. H. K. Nhan, and A. V. Le, "Multi-objective genetic algorithm-based autonomous path planning for hinged-tetro reconfigurable tiling robot," *IEEE Access*, vol. 8, pp. 121267–121284, 2020.

[11] L. Zhiyang, J. Tao, and Z. Yunhui, "A 3D route planning method based on improved ISPO algorithm," *Electron. Opt. Control*, vol. 25, no. 7, pp. 48–53, 2018.

[12] 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.

[13] Z. Yongqiang, "Real-time coordinated dispatching and economic operation of hybrid energy storage based on dynamic programming-genetic algorithm," *Acta Energiae Solaris Sinica*, vol. 2019, no. 4, pp. 1059–1066, 2019.

[14] Q. Xu, S. Chen, and B. Li, "Combining the ant system algorithm and simulated annealing for 3D/2D fixed-outline floorplanning," *Appl. Soft Comput.*, vol. 40, pp. 150–160, Mar. 2016.

[15] L. Jiangwei and X. Lunhui, "Research on path planning pased on simulated annealing algorithm and neural network algorithm," *Automat. Instrum.*, vol. 2017, no. 11, pp. 11–14, 2017.

[16] T. Huiyu, "3D route planning of UAV based on SAPSO algorithm," *Tactical Missile Technol.*, vol. 2017, no. 2, pp. 62–68, 2017.

[17] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *Appl. Soft Comput.*, vol. 11, no. 6, pp. 4135–4151, Sep. 2011.

[18] 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.

[19] Z. Dan and Q. Long, "Research on 3D trajectory planning of UAV based on ant colony-particle swarm fusion algorithm," *J. Jilin Inst. Chem. Technol.*, vol. 34, no. 3, pp. 93–98, 2017.

[20] J. Yao, C. Zhong, Z. Liu, and J. Xu, "3D trajectory optimization for secure UAV communication with CoMP reception," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[21] C. You and R. Zhang, "3D trajectory optimization in Rician fading for UAV-enabled data harvesting," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3192–3207, Jun. 2019.

[22] L. Chiaraviglio, F. D'Andreagiovanni, W. Liu, J. A. Gutierrez, and N. Blefari-Melazzi, "Multi-area throughput and energy optimization of UAV-aided cellular networks powered by solar panels and grid," *IEEE Trans. Mobile Comput.*, vol. 20, no. 7, pp. 2427–2444, Jul. 2021.

[23] S. A. G. Shirazi and H. Amindavar, "Fixed channel assignment using new dynamic programming approach in cellular radio networks," *Comput. Electr. Eng.*, vol. 31, nos. 4–5, pp. 303–333, Jun. 2005.

[24] Z. Zhilong and Z. Liang, "Simulation of target trajectory planning for missile battlefield," *Comput. Simul.*, vol. 36, no. 1, pp. 104–108, 2019.

[25] H. Pei, S. Feng, Y. Zhang, and D. Yao, "A cooperative driving strategy for merging at on-ramps based on dynamic programming," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11646–11656, Dec. 2019.

[26] J. Bernal, G. Dogan, and C. R. Hagwood, "Fast dynamic programming for elastic registration of curves," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 111–118.

[27] H. Zhou, H. L. Xiong, Y. Liu, N. D. Tan, and L. Chen, "Trajectory planning algorithm of UAV based on system positioning accuracy constraints," *Electronics*, vol. 9, no. 2, pp. 250–271, 2020.

[28] S. Shirazi and M. B. Menhaj, "A new genetic based algorithm for channel assignment problems," *Computational Intelligence, Theory and Applications*, B. Reusch, Ed. Berlin, Germany: Springer, 2006, pp. 85–92.

[29] L. G. Crespo and J. Q. Sun, "Stochastic optimal control via Bellman's principle," *Automatica*, vol. 39, no. 12, pp. 2109–2114, 2003.

[30] W. Minle, "Research on the convergence efficiency of genetic algorithm," *Comput. Technol. Automat.*, vol. 34, no. 4, pp. 59–63, 2015.

[31] L. Pan, G. Shenglian, and Z. Tong, "Optimization of reservoir operation by using dynamic programming-genetic algorithm," *Eng. J. Wuhan Univ.*, vol. 40, no. 5, pp. 1–6, 2017.

JIN ZIRONG was born in Handan, Hebei, China, in 1996. He received the B.S. degree in science from the Wuhan University of Technology, Wuhan, Hubei, in 2019, where he is currently pursuing the master's degree. He is currently committed to the research of operations research and control theory with the Wuhan University of Technology. His research interests include acoustic far-field imaging and flight path planning based on nonlinear optimization theory and swarm intelligence algorithm.

ZHANG LIANG was born in Suizhou, Hubei, China, in 1977. He received the B.S. degree from Central Normal University, in 2000, the M.S. degree from the Huazhong University of Science and Technology, in 2003, and the Ph.D. degree in applied mathematics from Wuhan University, in 2006.

He was a Lecturer and an Associate Professor with the Department of Mathematics, Wuhan University of Technology, from 2006 to 2011 and from 2011 to 2017, respectively, and became a Professor, in 2017. He is the Principal Investigator for three projects of the National Natural Science Foundation of China. He has published 15 articles and one book. His research interests include control theory and intelligent computing.

ZOU ZHILONG was born in Xiangyang, Hubei, China, in 1993. He received the B.S. degree in science from Huanggang Normal University, Xiangyang, in 2016, and the master's degree from the Wuhan University of Technology, Wuhan, Hubei, in 2019. He is currently committed to the research of operations research and control theory with the Wuhan University of Technology. His research interests include nonlinear optimization theory and path planning problems based on intelligent algorithms.