

Research Article

A Heuristic Evolutionary Algorithm of UAV Path Planning

Zhangjie Fu ^{1,2}, Jingnan Yu,¹ Guowu Xie,³ Yiming Chen,¹ and Yuanhang Mao¹

¹School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China

²College of Information Science and Technology, Jinan University, Guangzhou 510632, China

³Department of Computer Science and Engineering, University of California, Riverside, CA 92521, USA

Correspondence should be addressed to Zhangjie Fu; wwwfzj@126.com

Received 4 May 2018; Accepted 20 August 2018; Published 9 September 2018

Academic Editor: Naixue Xiong

Copyright © 2018 Zhangjie Fu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of the network and the informatization of society, how to improve the accuracy of information is an urgent problem to be solved. The existing method is to use an intelligent robot to carry sensors to collect data and transmit the data to the server in real time. Many intelligent robots have emerged in life; the UAV (unmanned aerial vehicle) is one of them. With the popularization of UAV applications, the security of UAV has also been exposed. In addition to some human factors, there is a major factor in the UAV's endurance. UAVs will face a problem of short battery life when performing flying missions. In order to solve this problem, the existing method is to plan the path of UAV flight. In order to find the optimal path for a UAV flight, we propose three cost functions: path security cost, length cost, and smoothness cost. The path security cost is used to determine whether the path is feasible; the length cost and smoothness cost of the path directly affect the cost of the energy consumption of the UAV flight. We proposed a heuristic evolutionary algorithm that designed several evolutionary operations: substitution operations, crossover operations, mutation operations, length operations, and smoothness operations. Through these operations to enhance our build path effect. Under the analysis of experimental results, we proved that our solution is feasible.

1. Introduction

With the rapid development of the network and the informatization of society, how to improve the accuracy of information is an urgent problem to be solved. The existing method is to use an intelligent robot to carry sensors to collect data and transmit the data to the server in real time [1]. Many intelligent robots have emerged in life; the UAV (unmanned aerial vehicle) is one of them. UAV is the unmanned aircraft operated by radio remote control devices and self-contained program; it is operated entirely or intermittently by on-board computers. Compared to the manned aircraft, it has many advantages, such as small size, low cost, easy to use, and portable camera or other sensors. UAVs are used in various tasks in industries such as commerce, agriculture, and military applications, including environmental monitoring, target identification, border patrols, and search and rescue assistance. However, UAVs also exposed a lot of problems, the most important issue is the endurance of UAVs. Energy scheduling problem has always been a thorny issue in wireless

network applications [2–4]; UAVs are the same. Due to the limited payload of UAVs, it is not possible to add more batteries to the UAVs. Only when the UAVs perform their tasks, can they fly as effective paths as possible. So path planning is the primary issue we want to solve when we study UAVs.

UAV path planning refers to the optimal path planning problem of UAVs. The main purpose of the optimization path is to find a safe flight path with minimum energy consumption on the premise of completing the UAV mission. That is to say, the essence of path planning is to find one in the workspace according to one or some optimization criteria (e.g., minimum working cost, shortest walking path, and shortest walking time). To find the optimal path for UAV, recently, a lot of work has been proposed [5, 6]. These proposed algorithms mainly are divided into two categories in terms of the algorithm form, including optimal algorithm and the heuristic algorithm. The optimal algorithms mainly include mathematical programming algorithm, parameter optimization algorithm and exhaustion

method, while the heuristic algorithms include A* algorithm, ant colony algorithm, genetic algorithm, and simulated annealing algorithm. There are many differences between two kinds of algorithms. First, the optimal algorithm will have high computational complexity when the scale is large, while the heuristic algorithm will have certain advantages in efficiency. Second, the heuristic algorithm exists in a random probability, while the optimal algorithm can solve this problem. Moreover, algorithms can be divided into coverage paths and optimal path planning algorithms according to the tasks performed by UAVs. Coverage path planning is mainly used in environmental monitoring and regional surveys. The main algorithms designed for environmental monitoring and regional surveys are raster scanning [7] and regional segmentation [8–13]. The purpose of the optimal path planning algorithm is to find the minimum flight cost path; it means that UAV fly in the effective path within the effective time. Both of these algorithms is aimed at minimizing the cost of UAV flight. Although existing algorithms have made some achievements on the UAV path planning, these methods only take the UAV's flight length or the number of turns of UAV flight paths (the number of UAV's corners) into consideration when referring to the UAV's flight cost. The classical evolutionary algorithm extends from one point to the surrounding adjacent points while traversing the path and cannot skip over the adjacent points.

In this paper, there are two contributions. (1) The first one is analysis of UAV flight energy consumption factors and feasibility of flight paths. We propose three cost functions: path security cost, path length cost, and path smoothness cost. The path security cost is used to determine whether the path is feasible; the length cost and smoothness cost of the path directly affect the cost of the energy consumption of the UAV's flight. (2) A heuristic evolutionary algorithm is proposed and several evolutionary operations are designed: substitution operation, cross operation, mutation operation, length operation, and smoothness operation, through these operations to enhance our build path effect.

The structure of this paper is as follows. In the second section, the current research status of intelligent robot path planning algorithms is introduced. Section 3 mainly introduces the construction of cost functions. Section 4 mainly introduces the design of path planning algorithms. Section 5 mainly introduces the experimental results. It mainly introduces some conclusions and further discussion.

2. Related Work

The core of path planning is the design of algorithms. Currently, path planning algorithms have attracted widespread attention. Whether they are global path planning or local path planning, the essence of the algorithm is to solve the travelling salesman problem [15]; the algorithms can be roughly classified into the following categories: traditional algorithms, heuristic algorithms, and intelligent bionic algorithms.

2.1. Traditional Path Planning Algorithms. The traditional path planning algorithms mainly include the visibility graph, artificial potential field, simulate anneal arithmetic method, and fuzzy logic algorithm.

Visibility Graph. The viability graph is to regard the robot as a point and connect the vertices of the robot, the target point, and the polygonal obstacle. These lines cannot cross the obstacle. So a visibility graph is formed [16]. Since the vertices of any two lines are visible, for the robot, all paths from the starting point along these lines to the target point are collision-free paths. Taking advantage of an optimized search algorithm to search for the visibility graph, a shortest path can eventually be found. The advantage is that the shortest path can be found. The disadvantage is the lack of flexibility. Once the starting point and target change, the visibility graph can be reconstructed.

Artificial Potential Field. The artificial potential field is a virtual force method proposed by Khatib [17]. The basic idea of this method is to consider the movement of robot in the environment as a movement of the robot in a virtual force field, where the obstacle generates a repulsive force on the robot and the target point generates a gravitational force on the robot. The combined force of gravity and repulsion acts as the robot control force to control the robot to avoid the obstacles and reach the target point. The advantage of the artificial potential field method is that it has good real-time performance and is convenient for robot bottom control. Nevertheless, the traditional artificial potential field method has the following disadvantages: local minimum points and target inaccessibility [18].

Simulate Anneal Arithmetic. Simulate anneal arithmetic is an effective approximation algorithm that is suitable for large-scale combinatorial optimization problems. It imitates the annealing process of solid materials. By setting the initial temperature, initial state, and cooling rate, the temperature is controlled to continuously decrease. Combining the probabilistic jump characteristics, make use of the domain of solution space for random search to avoid falling into a local optimum. It has the advantage of simple and efficient operation, etc. However, there are shortcomings such as slow convergence and randomness. The setting of its parameters is a key part to the realization of the algorithm [19].

Fuzzy Logic Algorithm. The fuzzy logic algorithm simulates the driver's driving experience and combines the physiological perception and action. According to the real-time sensor information of the system, the data elements are converted into fuzzy sets, and the output results are determined according to the membership function, and then the table information is obtained from the table [20]. The advantage of fuzzy logic algorithm is its robustness, which does not require the establishment of complex mathematical models, and avoids the disadvantages of mobile robots in other algorithms that are strongly dependent on the environment. The disadvantage is that it is more difficult to establish fuzzy rules [21].

2.2. Heuristic Algorithms. The heuristic algorithm has a strong path search capability and can be used in a discrete path topology. Common heuristic search algorithms include the A* algorithm, the Dijkstra algorithm and the Floyd algorithm.

(1) *A* Algorithm.* By setting appropriate heuristic functions, the A* algorithm comprehensively evaluates the value of each extended node and expands by comparing the nodes of each extended node to select the lowest cost node until the target point is found. The advantage of the A* algorithm is that there are few extension nodes and it has good real-time performance. The disadvantage is that the size of the robot itself is ignored in the actual motion process.

(2) *Dijkstra's Algorithm.* Dijkstra's algorithm is a typical shortest path algorithm. It extends from the starting point to the center layer until it reaches the end point and then obtains the shortest path by comparing the nodes' forward traversal [22]. The advantage is that the shortest path has a high success rate and good robustness. The disadvantage is that all nodes need to be traversed to obtain the shortest path. Compared to the A* algorithm, the efficiency is poor, and it is less effective for complex topological networks. In addition, the algorithm cannot handle the problem with negative edges.

(3) *Floyd Algorithm.* Floyd algorithm is an algorithm for finding the shortest path between vertices in a given weighted path topology network. It first converts the path network into a weight matrix and then finds the shortest distance between any two points in the weight matrix [23]. It has higher efficiency than Dijkstra's algorithm, but it also has disadvantages such as high time complexity and unsuitable for big data calculation.

2.3. Intelligent Bionic Path Planning Algorithms. Intelligent bionic algorithm is an algorithm discovered by people through bionics research in a series of natural phenomena, including ant colony optimization, particle swarm optimization algorithm, genetic algorithm, and neural network algorithm.

(1) *Ant Colony Optimization.* The ant colony optimization is an algorithm inspired by the phenomenon in which ants search for food. Each ant leaves a certain concentration of pheromone from the starting point to the foraging road. The pheromone will slowly evaporate with the passage of time, and the ants will use the pheromone concentration as the basis for path selection. The higher the pheromone concentration, the greater the probability that the path will be selected. As time goes on, the higher the concentration of pheromone left by the shorter path due to more ants' traversing, the higher the probability of ant selection on the higher concentration path and the more pheromones would be left. The shortest path will be discovered quickly after repeated iterations, which achieves the goal of path planning. Ant colony algorithm is essentially a parallel algorithm that is easy to implement by computers and has good global optimization ability. However, as the environment expands,

the computational volume will increase exponentially, and it is easy to fall into the local optimum [24–26].

(2) *Particle Swarm Optimization.* Particle Swarm Optimization (PSO) is also an iterative algorithm. It simulates bird's flight predation behavior and makes use of the sharing of information among individuals in a group to make the movement of the entire population produce an evolutionary process from disorder to order in the problem solving space, thereby obtaining an optimal solution. The system is initialized as a set of random solutions. The individual updates the position through the individual historical best solution and the global history optimal solution makes it follow the optimal particle and iteratively searches for the optimal value. Compared with genetic algorithms, PSO does not have many parameters that need to be adjusted. The algorithm has the advantages of simplicity, easy implementation, good robustness, and fast convergence, but it is easy to fall into the local optimal solution [27].

(3) *Genetic Algorithm.* Genetic algorithm is a computational model that simulates the natural selection of Darwin's biological evolution theory and the genetic evolution process of genetic mechanisms. Starting from a population that may have potential solution set, the population is first coded, and the adaptive value of each individual is calculated. According to the principle of survivability of the fittest, selection, crossover, and mutation are performed, the poorer individuals are eliminated, and the remaining individuals are repeatedly iterated again until the best individual is produced. The greatest advantage of the genetic algorithm is that it is easy to combine with other algorithms and give full play to its own iterative advantages. The disadvantage is that because there is no feedback information, the computational efficiency will be greatly reduced after the later period [28].

(4) *Neural Network Algorithm.* Artificial neural network system appeared after the 1940s. It is made up of a number of connective weights with adjustable neurons. It has features such as large-scale parallel processing, distributed information storage, and good self-organizing self-learning capabilities [29]. The neural network algorithm is used in path planning as follows: the neural network is used to describe the environmental constraints and the collision energy function is calculated. The sum of the collision energy function and the distance function is used as the optimization objective function, and the point set is determined by optimizing the extreme value of the objective function. The equation of motion ultimately makes the iterative path point set tend to be the optimal planning path. Although neural network has excellent learning ability, its poor generalization ability is its fatal weakness, but because of its strong learning ability and good robustness, its combination with other algorithms has become a research hotspot in the field of path planning [30].

3. The Cost Function

In some classical optimization algorithms, the length of the path represents the energy consumption of UAV flight.

TABLE 1: The meaning of the parameters in this paper.

Parameter	
$Cost_s$	the security cost of the path
$Cost_l$	the length cost of the path
$Cost_a$	the smoothness cost of the path
Cost	the effective cost of the path
l_i	the length of the i th obstacle covering the path
L	the total length of the path
m	the number of obstacles on the path
n	the number of points traversed by the path
(x_n, y_n)	the coordinate of the n th traversal point
θ_n	the rotation angle of the path at the n th traversal point
W_l	the energy consumption of UAV flying in a straight line
W_a	the energy consumption of drones flying at corners
ω_1, ω_2	two weights

Thinking about the problem of energy consumption and the feasibility of the path of UAV flight, this paper proposes three costs. The cost of path security is to determine whether the path is feasible. The length cost of the path and the smoothness cost of the path represent the energy cost of the path flight of the UAV. Table 1 shows the meaning of the parameters in our paper.

3.1. Path Security. During the flight of a UAV, the most important thing is not to collide with obstacles in the environment. The purpose of the security of the route is to avoid obstacles on the route. Therefore, the number of obstacles on a path is the primary determinant of the feasibility of the path. We define the degree of security of the path to be calculated by the occupancy of obstacles that have passed through the UAV flight path. Equation (1) shows the calculation of the path security $Cost_s$.

$$Cost_s = \sum_{i=1}^m \frac{l_i}{L} \quad (1)$$

$Cost_s$ indicates the security cost above the path, m is the number of obstacles on the path, and l_i is the length of the i th obstacle covering the path. L is the total length of the path.

We ultimately find that the theoretical number of path obstructions should be 0. This cost is used as a factor to determine the feasibility of the path.

3.2. The Length of the Path. The length of the path refers to the path from the starting point to the end point of the UAV. The length of the path directly relates to the UAV flight time, which directly affects the energy consumption of the UAV. Therefore, the shorter the UAV's flight path means less energy consumed by UAV flight. In mathematics, we mainly calculate the Euclidean metric distance of the flight path of each UAV. The length cost of the path is shown in

$$Cost_l = \sum_{n=1}^{n-1} \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2} \quad (2)$$

$Cost_l$ is the length of the path of which UAV go through the threat area and $Cost_l$ represents the length cost of the path, which is the most important factor affecting the flight energy consumption of the UAV. We can calculate $Cost_l$ by point coordinates on the UAV's flight path. n is the number of points traversed by the flight path and (x_n, y_n) is the coordinate of the n th traversal point.

3.3. The Smoothness of the Path. The principle of the corner of a multicopter UAV is to obtain a body rotation force by changing the speed of the adjacent motor by a UAV. Therefore, the angle of rotation is also a factor that determines energy consumption. Considering this situation, we define a cost function for path smoothness. Equation (3) shows the smoothness cost $Cost_a$.

$$Cost_a = \begin{cases} 0, & n \leq 2 \\ \sum_{n=2}^{n-1} \theta_n, & n > 2 \end{cases} \quad (3)$$

$Cost_a$ represents the smoothness cost of the UAV's flight path. At a fixed angular speed, $Cost_a$ also affects the UAV's flight energy consumption, which is the angle number of the UAV's path rotation. θ_n is the rotation angle of the path at the n th traversal point, which can be calculated by the law of cosines.

3.4. Effective Cost Function. When searching for the best flight path of UAV, we must consider the three cost functions above. Since the three cost function units and the meaning of inclusion are different, the sum of the three costs cannot be calculated directly. Therefore, we set the weight based on the weight of each cost function. Since both the length of the path and the smoothness of the path can affect the energy consumption of the UAV flight, we can determine the weight by the energy loss value of the UAV between the length of the path and the smoothness. Jalil Modares et al. [31] verified that the energy consumption of UAV and the path distance and number of angles are approximately linear. We use them to establish the energy model to determine the weight values of the length cost and the angle cost. Energy equation (4) shows the relationship between UAV energy W_l and flight distance:

$$W_l = \omega_1 Cost_l \quad (4)$$

Equation (5) shows the relationship between the UAV energy W_a and the angle of rotation:

$$W_a = \omega_2 Cost_a \quad (5)$$

The energy cost of the flight path is shown in

$$Cost = \omega_1 Cost_l + \omega_2 Cost_a \quad (6)$$

where Cost represents an effective cost that directly affect the energy consumption of the UAV and is directly used as an important factor in determining the fitness function of the path in the algorithm and the value method of ω_1 and ω_2 is shown in experiment.

4. Algorithm Design

Similar to evolutionary algorithms, the methods our proposed include initializing the population, calculating the individual's cost fitness, evolving operations, and jumping out of the loop. The optimization mode proposed by this paper is different from the classical optimization algorithm. Our method is to minimize the energy consumption of the path and proposes several evolutionary operations: replacing operation, crossover operation, length operation, and smoothness operation. Our method retains the infeasible path, which improves the diversity of paths, so that it is possible to jump out of the local optimization. The classical evolutionary algorithm extends from one point to the surrounding adjacent points while traversing the path and cannot skip over the adjacent points. When traversing a path, our method can skip some adjacent traversal points, which can effectively reduce the length and smoothness of the path. The algorithm flow is shown in Figure 1.

Algorithm 1 is the pseudocode of this algorithm. Step 1 is initialization population; we randomly generate a set of solutions. Steps 2-4 are a replacement operation. During the initialization and evolution of classical optimization algorithms, the solution generated by our method may be an infeasible solution. Therefore, we propose a replacement operation. The replacement operation is mainly to turn the infeasible solutions into feasible solutions. Steps 6-13 are to find the optimal solution by the evolutionary operations. The paper describes in detail the evolutionary operation in the later. Our termination condition is the number of iterations.

4.1. Replacing Operation. During the initialization and evolution of classical optimization algorithms, their paths are feasible paths. Such operations easily fall into local optimization. In our method, the infeasible path is retained, which improves the diversity of paths, so that it is possible to jump out of the local optimization. A replacing operation is proposed to deal with an infeasible path in our method. The replacement operation is mainly to turn the infeasible path into a feasible path and determine the feasibility of the path through the path security cost $Cost_s$ (see (1)). When the path security cost is 0, the path is feasible; otherwise the path is not feasible. When the path is not feasible, finding a new feasible path replaces the infeasible path. The specific steps are as follows: First, the collision path segmentation is determined. Then the distance of the obstacle vertex to the segmented path is calculated, respectively. The point of the maximum distance between the two sides of the piecewise path is taken as the candidate for in terms of passing the obstacle and connecting the endpoints of the segmentation path and the candidate points, respectively, to generate two new paths. Calculating the effective cost $Cost$ (see (5)) of the two paths, the path with small energy consumption cost is selected to replace the original collision. As shown in Figure 2, when the segmented path SE is infeasible, the two points P, Q are found that are farthest from the obstacle vertex on both sides of the collision path SE to the segmented path. The infeasible path SE is replaced by two feasible paths SPE and SQE, and then the effective costs of SPE and SQE are calculated, respectively.

1	Initialization population
2	Calculating the security cost of each individual $Cost_s$
3	If $Cost_s \neq 0$
4	Replacing operation
5	Updating population, go to step 2
6	If $Cost_s = 0$
7	Calculating effective cost of each individual $Cost$
8	If has not terminated
9	Evolutionary operations
10	Updating population, go to step 2
11	If termination
12	Output optimal path
13	End

ALGORITHM 1

Last, the minimum energy cost path SPE is chosen to take the replace of the path SE.

4.2. Crossover Operation. The crossover operation is to generate a new path through cross replacement. When there are two or more cross points in two paths, two cross points are randomly selected, and the path between the two selected cross points is exchanged to generate new paths. The effective cost of four paths is calculated, and the optimal effective cost path is chosen. Figure 3 is the result of the crossover operation. As shown in Figure 3(a), when the initial split path L1 and L2 have more than two intersections, Figure 3(b) shows that the crossover produces two new paths L3 and L4.

4.3. Mutation Operation. Mutation refers to replacing certain gene values in individual code strings with other gene values based on the probability of mutation to form a new individual. In this paper, the feasibility of the path will change whenever the coordinate node of the path mutates. In order to solve this problem, two different probability mutation operations are proposed to deal with the feasible path and the infeasible path respectively. When the path is feasible, the coordinates of the path node are mutated in a small range with a small probability, and the variation path is still feasible; when the path is not feasible, the coordinates of the path node are mutated in a large range with a large probability. And ensure better individual costs after the mutation. When the path is not feasible, we change the path to a feasible path by replacing the operation.

4.4. Length Operation. The purpose of the path length operation is to reduce the distance of the drone flight path. When the path is feasible, some paths usually contain additional path segments. Considering the length cost of the path and the feasibility of the path, if the path is still feasible after randomly deleting one or more path nodes, the path is deleted from the original path node and a new path is generated. Figure 4 shows an example of a path length operation. As shown in Figure 4(a), ABCD is the initial path. As shown in

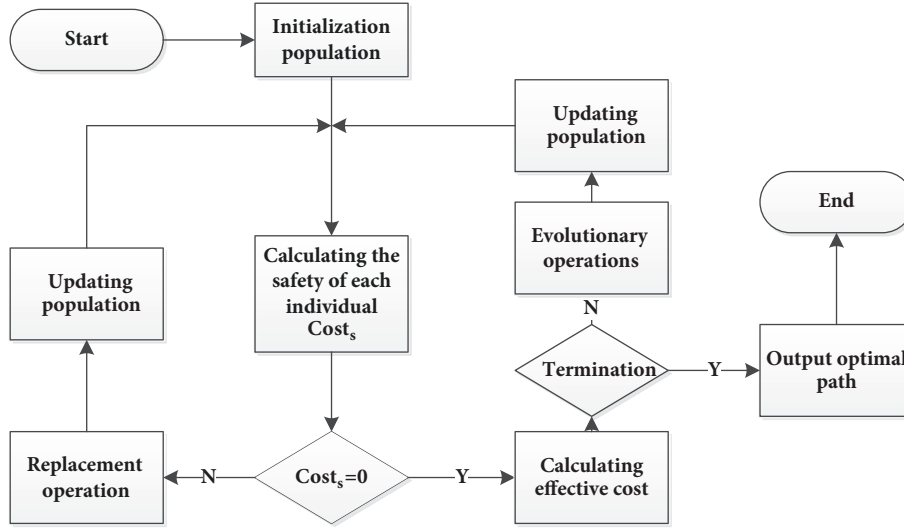


FIGURE 1: Flowchart of our method.

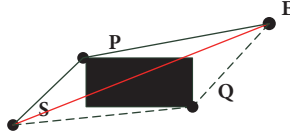


FIGURE 2: An example of an infeasible path replacing operation.

Figure 4(b), if the point B is randomly deleted in the path and the path is still feasible, a new path ACD is generated.

4.5. Smoothness Operation. The purpose of the path smoothness operation is to reduce the number of turns of the UAV and randomly select two adjacent straight-line paths, which are used as input. And then the angle between them is changed. In two adjacent straight paths, the midpoint of a straight path is randomly selected to replace the intersection of the two straight paths, and a new path is generated. Figure 5 shows an example of path smoothness operation. ABC is an adjacent straight-line path, and ADC is a new path through path smoothing.

5. Experiment

5.1. Experimental Model. In the experiment, according to the wireless network data acquisition model proposed by Zheng et al. [32], we built a system platform, as shown in Figure 6. The system platform includes the UAV, on-board computers manifold, ground station, and power sensors. Table 2 shows the specific parameters of the system platform equipment. The main use of UAV in this system platform is DJI GO M100, because DJI GO M100 supports secondary development, which is conducive to the monitoring of data. The use of on-board computer in this system platform is DJI GO manifold. The system of manifold is Linux; the system has Mobile SDK and on-board SDK development interfaces. The system platform uses manifold and data acquisition sensors to build

TABLE 2: Specific parameters of the system platform equipment.

Equipment	
UAV	DJI GO M100
Onboard computer	DJI GO Manifold
Ground station	Iphone7 plus+ Lenovo ThinkPad x260
Power sensors	ADS1115 IC

a detection module. The system platform uses the manifold to send the detected data to the ground station. Through our system platform, the UAV's own data and other inspection data can be transmitted to the ground station in real time, and the flight status of the UAV can be effectively observed in real time.

5.2. Setting of Weights. In this experiment, we analyzed the relationship between flight path distance and rotation angle with the energy consumption of UAV based on the energy model proposed by Jalil Modares et al. [31]. We determine the value of ω_1 and ω_2 by analyzing their relationship. Note that the values ω_1 and ω_2 detected by different UAVs are different. The UAV used in this experiment is the DJI GO M100.

In this experiment, through analyzing the relationship between path distance and energy consumption, the fly speed of UAV is set as 10m/s to conduct 20 times flights in the line distance of 50m, 100m, and 150m, respectively. To study the relations between rotation angle and energy consumption, the UAV is allowed to fly 20 times in the following four situations: (1) straight flight distance 100m, (2) straight distance 50m, 45 degree angle, and straight flight 50m, (3) straight distance 50m, 90 degree rotation, and 50m Straight flight, (4) straight distance 50m, degree angle 45, and straight flight 50m. To analyze the relationship between the angle number and the energy consumption, the measured value is subtracted from the average test value of the line distance. Figure 7 shows the relationship between the drone

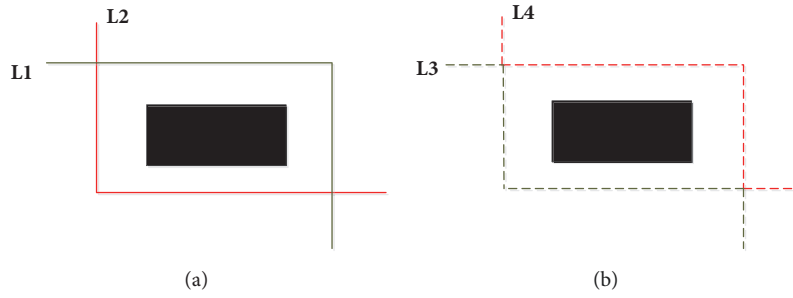


FIGURE 3: An example of a crossover operation.

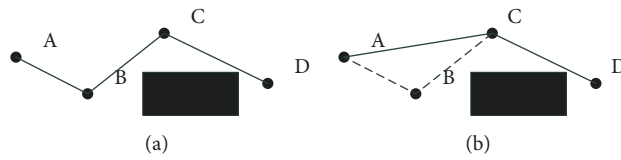


FIGURE 4: An example of a path length calculation.

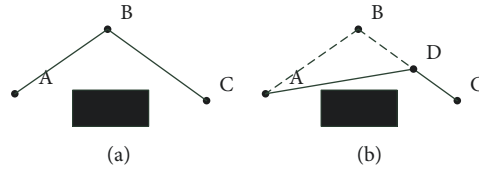


FIGURE 5: An example of a path length calculation.

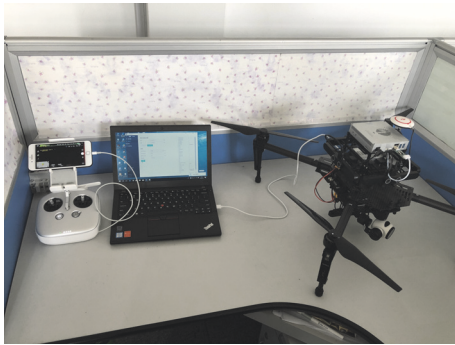


FIGURE 6: System model.

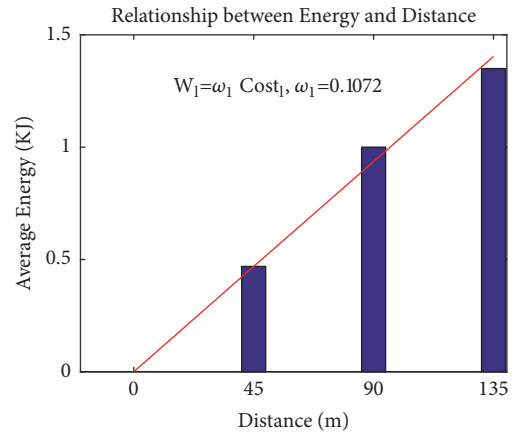


FIGURE 7: Relationship between energy and distance.

energy loss and the length of the flight path; Figure 8 shows the relationship between energy loss and rotational angle of the UAV. Through the experiment, we get the value of ω_1 and ω_2 , $\omega_1 = 0.1072$ and $\omega_2 = 0.0104$.

5.3. Experimental Results. This section mainly discusses the effect of our proposed method. The result of our method is compared with other algorithms. To assess the feasibility of our program, we use matlab2014a for simulation experiments. The implementation platform of us is Windows 7 server with Core CPU 2.85GHz.

The method proposed in this paper is based on an evolutionary algorithm of genetic algorithm, so we choose

several classical evolutionary algorithms as the comparison algorithm. We have done some work comparisons in our previous works [33]. In this paper, we make different comparisons between the efficiency of the algorithm and the results generated by the algorithm. We make different comparisons between the efficiency of the algorithm and the results generated by the algorithm. We use the same modeling environment for comparison experiments. I set the lower left corner as the starting point and the upper right corner as the end point to generate an optimal path; the grid ratio is 1:10; the

TABLE 3: Comparison of algorithms.

Algorithm	Time (s)	Path length (10m)	Angle (degree)	Theoretical Cost (KJ)	Actual Cost (KJ)
GA[14]	137.76	13.899	-	14.900	19.910
PSO	201.46	13.899	-	14900	19.816
Our method	181.25	13.291	71.565	14.992	16.071

TABLE 4: The result of using the path length as a cost function.

Algorithm	Time (s)	Path length (10m)	Theoretical Cost (KJ)	Actual Cost (KJ)
GA	137.76	13.899	14.900	19.910
Our method	120.25	13.291	14.248	16.071

TABLE 5: The result of using our proposed cost as a cost function.

Algorithm	Time (s)	Path length (10m)	Angle (degree)	Theoretical Cost (KJ)	Actual Cost (KJ)
GA	210.46	13.899	135	16.304	18.673
Our method	181.25	13.291	71.565	14.992	16.071

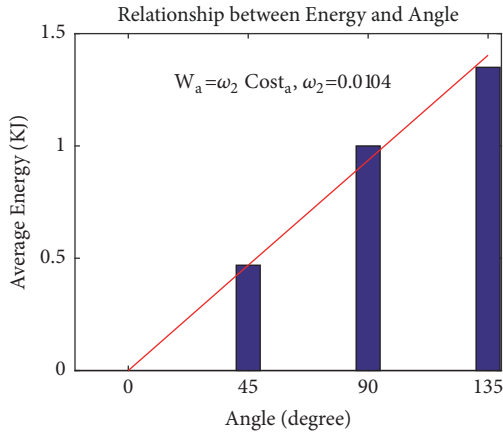


FIGURE 8: Relationship between energy and angle.

number of iterations is set to 500 times. Figure 9(a) shows the path generated by the classical GA [14]. Figure 9(b) shows the path effect generated by PSO. Figure 9(c) is the result of the algorithm generation path proposed by this paper.

In Table 3, GA and PSO methods use the path length as a cost function. Our method uses the path security, smoothness, and length as cost functions. Time represents the time of each method run. The path length is the length of the path generated by each method. The angle is the angle of the path generated by each method. The theoretical cost of our method is given by (6). The theoretical costs of GA and PSO are given by (4). The actual cost is to measure energy consumption through our system platform of the path generated by each method. Because GA and PSO methods use the path length as a cost function, we only discuss their run time and the length of the path. According to the data in Table 3, the running time of GA is better than that of PSO. This conclusion has been verified by Roberge V et al. [34]. In aspect of the running time of the algorithm, because our method needs to calculate the path smoothness cost, the time

cost of our method is larger. However, the time overhead of our method is better than PSO in this case. According to the data in Table 3, our method is superior to PSO in other aspects. Compared with GA, although our method has a large time cost, the path length generated by our method is dominant.

Because the two methods use different cost functions, we cannot judge whether the two schemes are good or bad. So we set up two sets of comparative experiments. One set of experiments used the length of the path as a cost function, and the other set used our proposed cost as a cost function. Figure 10 shows the result of using the path length as a cost function. Figure 10(a) shows the path generated by the classical GA. Figure 10(b) is the result of the algorithm generation path proposed by this paper. Table 4 shows the parameters of the results in Figure 10. Figure 11 shows the result of using our proposed cost as a cost function. Figure 11(a) shows the path generated by the classical GA. Figure 11(b) is the result of the algorithm generation path proposed by this paper. Table 5 shows the parameters of the results in Figure 11.

In Table 4, the theoretical cost is the effective cost (see (4)) of the path generated by each method. According to the data in Table 4, two methods use the path length as a cost function. In aspect of the running time of the algorithm, our method is better than GA. In aspect of path length our method is shorter than GA. In Table 5, the theoretical cost is the effective cost (see (6)) of the path generated by each method. According to the data in Table 5, two methods used our proposed cost as a cost function. In aspect of the running time of the algorithm, our method is better than GA. In aspect of path length our method is shorter than GA. According to the data in Tables 4 and 5, we can know that our method is better than GA in the case of using the same cost function.

The theoretical cost is the effective cost (see (6)) of the path generated by each method. We designed a set of experiments. From the above experiments, we can know that the path effects generated by our method using different cost functions are the same in our envisaged environment. Note

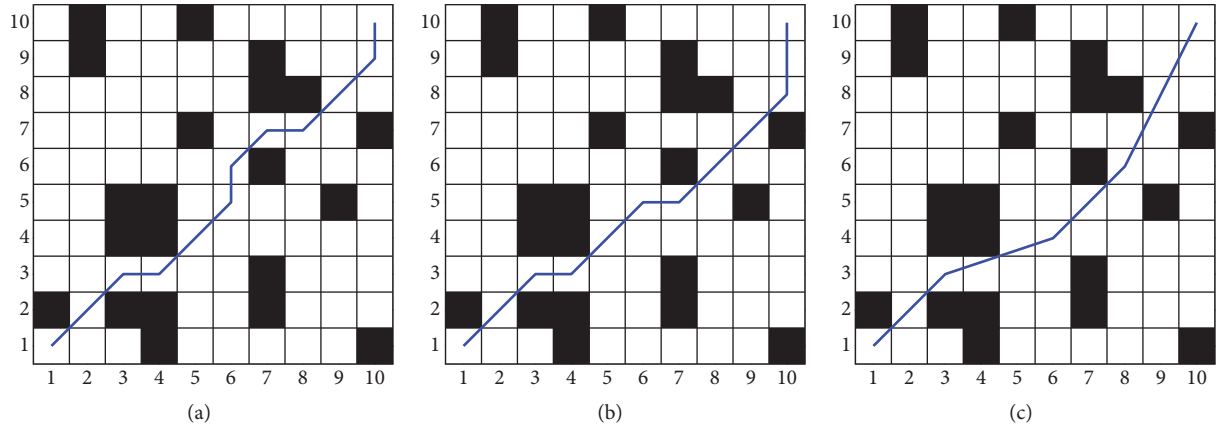


FIGURE 9: Three algorithms running path results.

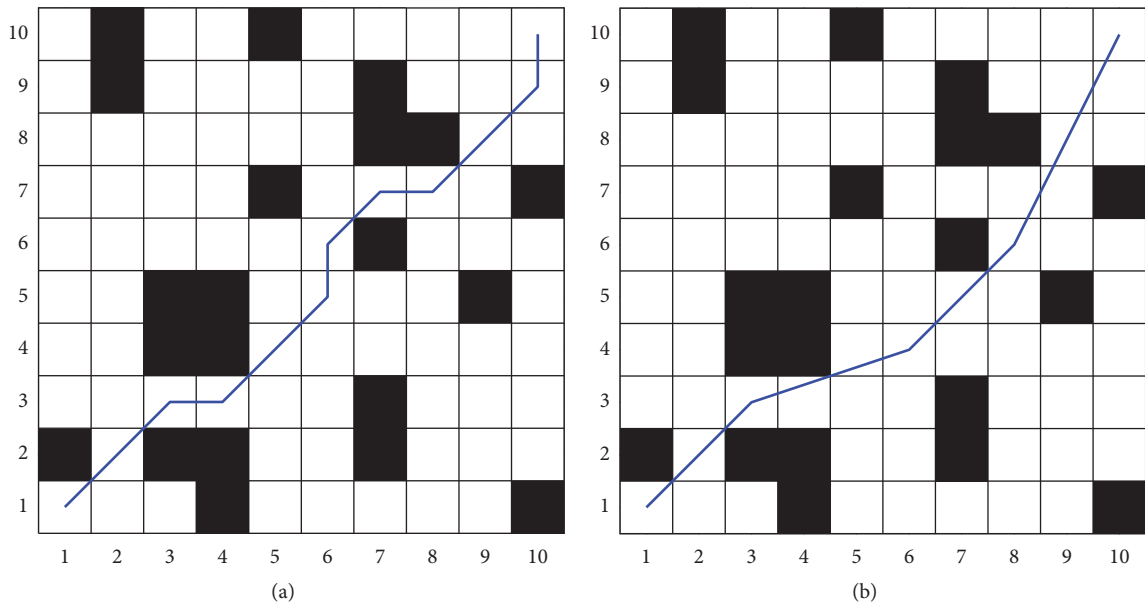


FIGURE 10: The result of using the path length as a cost function.

that the path effects generated by our method using different cost functions are the different in same environment. The experimental results of our experiments are the same because our environment is simple. We use the path generated by our method to actually fly 20 times. Record the energy consumption of each flight. Figure 12 is the result of our experiment. In Figure 12, the x-axis is the number of experiments and the y-axis is the energy consumption of the drone. Each point is the actual energy consumption of each flight. The red line is the average of the actual energy consumption. The blue line is the theoretical value that uses our cost function to calculate energy consumption. The green line is the theoretical energy consumption calculated using length as a function of cost. From the data in Figure 12, we can see that our cost function is close to the actual value. The reason for the deviation of our cost function may be that wind and other factors affect the energy consumption of the drone in the actual flight

environment. In future work, we can continue to study other factors that affect the energy consumption of drones, we continue to refine the design of the cost function.

6. Conclusion and Future Work

This paper analyzes the factors affecting the flight path of UAVs and proposes three cost functions: path security cost, length cost, and smoothness cost. The path security cost is used to determine whether the path is feasible; the length cost and smoothness cost of the path directly affect the cost of the energy consumption of the UAV's flight. Because of the weighting of the two costs, we set the weights for the two costs through real experimental data. In order to improve the effect of generating effective paths, we propose a heuristic evolutionary algorithm that sets several evolutionary operations: substitution operation,

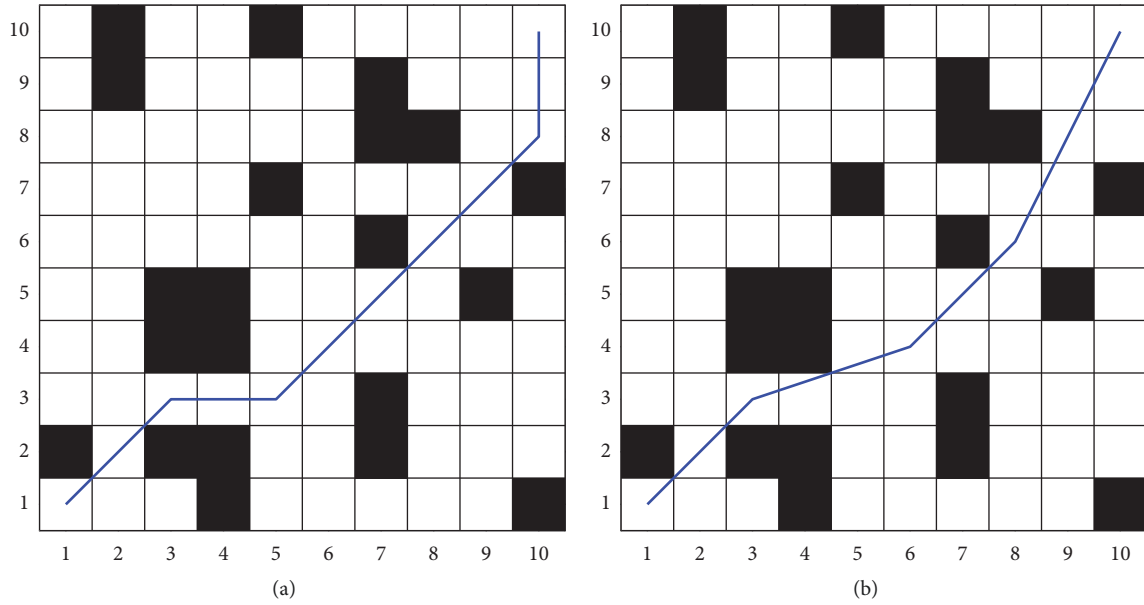


FIGURE 11: The result of using our proposed cost as a cost function.

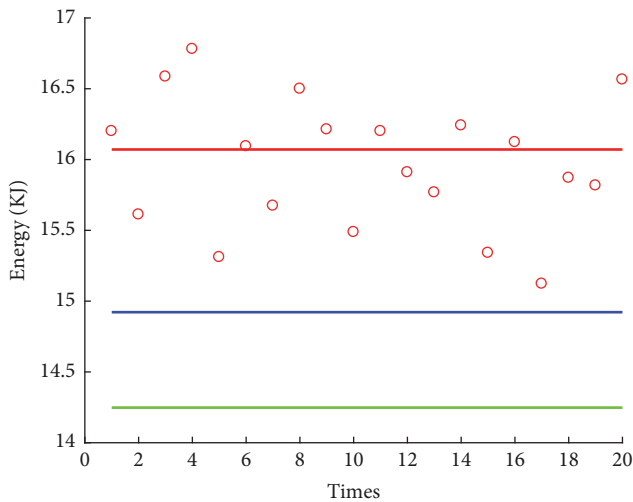


FIGURE 12: The actual energy consumption of the path.

cross operation, mutation operation, length operation, and smoothness operation. Through our combination of real experiments and simulation experiments, we verified that our proposed method is feasible; compared with classical algorithms, our method is better at generating the optimal path.

The three cost functions proposed in this paper influence each other. We must find the best of the three cost functions, that is, a multiobjective optimization problem. The solution proposed in this paper is to convert multiobjective optimization into single-objective optimization by setting weights. Our next work will use a multiobjective optimization algorithm to solve this problem.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the NSFC (61772283, 61672294, U1536206, 61502242, U1405254, and 61602253), BK20150925, R2017L05, PAPD Fund, project funded by China Postdoctoral Science Foundation, Major Program of the National Social Science Fund of China (17ZDA092), Qing Lan Project, Meteorology Soft Sciences Project, and Yueyan Zhi.

References

- [1] J. H. Cui, Y. Y. Zhang, and Z. P. Cai, "Securing Display Path for Security-Sensitive Applications on Mobile Devices Computers," *Materials & Continua*, vol. 55, no. 1, pp. 17–35, 2018.
- [2] H. Cheng, Z. Su, N. Xiong et al., "Energy-Efficient Node Scheduling Algorithms for Wireless Sensor Networks," *Information Sciences*, vol. 329, no. 2, pp. 461–477, 2016.
- [3] H. Cheng, N. Xiong, A. V. Vasilakos, L. Tianruo Yang, G. Chen, and X. Zhuang, "Nodes organization for channel assignment with topology preservation in multi-radio wireless mesh networks," *Ad Hoc Networks*, vol. 10, no. 5, pp. 760–773, 2012.
- [4] A. Vinel, W.-S. E. Chen, N. N. Xiong, S. Rho, N. Chilamkurti, and A. V. Vasilakos, "Enabling wireless communication and networking technologies for the internet of things [Guest editorial]," *IEEE Wireless Communications Magazine*, vol. 23, no. 5, pp. 8–9, 2016.

- [5] B. M. Sathyaraj, L. C. Jain, A. Finn, and S. Drake, "Multiple UAVs path planning algorithms: a comparative study," *Fuzzy Optimization and Decision Making. A Journal of Modeling and Computation Under Uncertainty*, vol. 7, no. 3, pp. 257–267, 2008.
- [6] R. Samar and W. A. Kamal, "Optimal Path Computation for Autonomous Aerial Vehicles," *Cognitive Computation*, vol. 4, no. 4, pp. 515–525, 2012.
- [7] A. Ahmadzadeh, J. Keller, G. Pappas et al., "An optimization-based approach to time-critical cooperative surveillance and coverage with UAVs," *Experimental Robotics. Springer Berlin/Heidelberg*, pp. 491–500, 2008.
- [8] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Efficient complete coverage of a known arbitrary environment with applications to aerial operations," *Autonomous Robots*, vol. 36, no. 4, pp. 365–381, 2014.
- [9] E. Santamaria, F. Segor, I. Tchouchenkov et al., "Path planning for rapid aerial mapping with unmanned aircraft systems," in *Proceedings of the Eighth International Conference on Systems*, 2013.
- [10] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 27–32, May 2001.
- [11] R. Mannadiar and I. Rekleitis, "Optimal coverage of a known arbitrary environment," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation, ICRA 2010*, pp. 5525–5530, USA, May 2010.
- [12] I. Maza and A. Ollero, "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms," *Distributed Autonomous Robotic Systems 6. Springer, Tokyo*, vol. 6, pp. 221–230, 2007.
- [13] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.
- [14] W. Zhou, Z. Yi, and Y. Ruimin, "Mobile robot path planning based on genetic algorithm," *Microcomputer Information*, vol. 24, no. 26, pp. 187–189, 2008.
- [15] N. Xiong, W. Wu, and C. Wu, "An Improved Routing Optimization Algorithm Based on Travelling Salesman Problem for Social Networks," *Sustainability*, vol. 9, no. 6, p. 985, 2017.
- [16] C. Chen, J. Tang, and Z. Jin, "A Path Planning Algorithm for Seeing Eye Robots Based on V-Graph," *Mechanical Science and Technology for Aerospace Engineering*, 2014.
- [17] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [18] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Proceedings of the Congress on Evolutionary Computation (CEC 00)*, vol. 1, pp. 256–263, July 2000.
- [19] A. E.-S. Ezugwu, A. O. Adewumi, and M. E. Frincu, "Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem," *Expert Systems with Applications*, vol. 77, pp. 189–210, 2017.
- [20] D. Adhikari, E. Kim, and H. Reza, "A fuzzy adaptive differential evolution for multi-objective 3D UAV path optimization," in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation, CEC 2017*, pp. 2258–2265, Spain, June 2017.
- [21] O. Castillo, H. Neyoy, J. Soria, M. García, and F. Valdez, "Dynamic Fuzzy Logic Parameter Tuning for ACO and Its Application in the Fuzzy Logic Control of an Autonomous Mobile Robot," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, 2013.
- [22] S. Wang X, X. Wu Z, and X. Wang SXWu Z, "Improved Dijkstra shortest path algorithm and its application," *Computer Science*, vol. 39, no. 5, pp. 223–222, 2012.
- [23] Z. He and L. Zhao, "The comparison of four UAV path planning algorithms based on geometry search algorithm," in *Proceedings of the 9th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2017*, pp. 33–36, China, August 2017.
- [24] X. F Wan, W. Hu, W. F. Fang et al., "Research on path planning of robot based on improved ant colony algorithm," *Computer Engineering and Applications*, vol. 50, no. 18, pp. 63–66, 2014.
- [25] X. Chen, Y. Kong, X. Fang, and Q. Wu, "A fast two-stage ACO algorithm for robotic path planning," *Neural Computing and Applications*, vol. 22, no. 2, pp. 313–319, 2013.
- [26] T. Zhao, X. Pan, and Q. He, "Application of dynamic ant colony algorithm in route planning for UAV," in *Proceedings of the 7th International Conference on Information Science and Technology, ICIST 2017*, pp. 433–437, Viet Nam, April 2017.
- [27] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 2017.
- [28] G. Ji, "A survey of Genetic Algorithm," *Computer Applications and Software*, vol. 21, no. 2, pp. 69–73, 2004.
- [29] S. A. Yildizel and A. U. Öztürk, "A study on the estimation of prefabricated glass fiber reinforced concrete panel strength values with an artificial neural network model," *Computers Materials & Continua*, vol. 52, no. 1, pp. 41–52, 2016.
- [30] W. Wang, S. M. Wei, Y. Q. Yang, Y. F. Jiang, and L. I. DuanLing, "Path planning for a mobile robot using neural networks," *Journal of Beijing University of Technology*, vol. 45, no. 10, pp. 221–225, 2009.
- [31] J. Modares, F. Ghanei, N. Mastronarde, and K. Dantu, "UB-ANC planner: Energy efficient coverage path planning with multiple drones," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation, ICRA 2017*, pp. 6182–6189, Singapore, June 2017.
- [32] H. Zheng, W. Guo, and N. Xiong, "A Kernel-Based Compressive Sensing Approach for Mobile Data Gathering in Wireless Sensor Network Systems," *IEEE Transactions on Systems Man & Cybernetics Systems*, 2017.
- [33] J. Yu, Y. Chen, Y. Mao et al., "A Heuristic Evolutionary Algorithm of UAV Path Planning," in *Proceedings of the International Conference on Cloud Computing and Security (ICCCS 2018)*, 2018, <http://www.icccsconf.org/%E6%8E%A8%E8%8D%90SCI%E6%9C%9F%E5%88%8A%E8%AE%BA%E6%96%87%E5%88%97%E8%A1%A8.pdf>.
- [34] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013.



Hindawi

Submit your manuscripts at
www.hindawi.com

