

## Research Article

# Path Planning of Mobile Robot Based on Improved Multiobjective Genetic Algorithm

Kairong Li <sup>1</sup>, Qianqian Hu <sup>1</sup> and Jinpeng Liu<sup>2</sup>

<sup>1</sup>School of Information Engineering, Yangzhou University, Yangzhou 225127, China

<sup>2</sup>The 29th Research Institute of China Electronics Technology Group Corporation (CETC-29), Chengdu 610036, China

Correspondence should be addressed to Kairong Li; [krl@yzu.edu.cn](mailto:krl@yzu.edu.cn) and Qianqian Hu; [1357824557@qq.com](mailto:1357824557@qq.com)

Received 12 August 2020; Revised 21 September 2020; Accepted 13 April 2021; Published 30 April 2021

Academic Editor: Alessandro Bazzi

Copyright © 2021 Kairong Li 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.

Path planning is the core technology of mobile robot decision-making and control and is also a research hotspot in the field of artificial intelligence. Aiming at the problems of slow response speed, long planning path, unsafe factors, and a large number of turns in the conventional path planning algorithm, an improved multiobjective genetic algorithm (IMGGA) is proposed to solve static global path planning in this paper. The algorithm uses a heuristic median insertion method to establish the initial population, which improves the feasibility of the initial path and generates a multiobjective fitness function based on three indicators: path length, path security, and path energy consumption, to ensure the quality of the planned path. Then, the selection, crossover, and mutation operators are designed by using the layered method, the single-point crossover method, and the eight-neighborhood-domain single-point mutation method, respectively. Finally, the delete operation is added, to further ensure the efficient operation of the mobile robot. Simulation experiments in the grid environment show that the algorithm can improve the defects of the traditional genetic algorithm (GA), such as slow convergence speed and easy to fall into local optimum. Compared with GA, the optimal path length obtained by planning is shortened by 17%.

## 1. Introduction

A mobile robot is a kind of machine device which can perform work automatically [1]. It can not only accept the command of human beings but also act according to the scheme made by artificial intelligence technology [2]. With the development of science and technology, mobile robot has attracted more and more attention. Its application scope is also greatly expanded; from industry, agriculture, medical, service, and other industries to national defense, space exploration, and other dangerous occasions, mobile robots have played an important role [3]. As a key technology for the mobile robot to realize autonomous motion, path planning is one of the most active research directions in the field of mobile robot [4, 5]. The path planning problem of the mobile robot is to make the robot autonomously find a noncollision optimal or suboptimal path from the given starting point to the target point according to certain index requirements (such as path length, path safety, and planning time) within the specified environment with obstacles [6]. According to the degree of

mastering the environment, path planning can be divided into global path planning in known environment and local path planning in unknown environment. According to whether the environmental information changes with time, path planning can be divided into static path planning and dynamic path planning. This paper mainly studies the known static path planning in global environment [7–9]. The general steps include environment modeling, path searching, and path optimization. The planning algorithm used in the path search is the core of the whole mobile robot path planning problem, and the selection of the algorithm determines the quality of the planning path [10]. At present, there are many effective solutions to the problem of mobile robot path planning, but with the continuous development of science and technology, the environment faced by path planning technology will become more complex [11], and the task requirements will become more stringent. This requires that the path planning algorithm should have the ability to quickly respond to the complex environment [12], less energy consumption, and be able to avoid all obstacles. This

is not a problem that can be solved by a single traditional algorithm [13]. Therefore, an improved multiobjective genetic algorithm (IMGGA) is proposed in this paper. The purpose is to improve the shortcomings of the traditional genetic algorithm by optimizing some genetic operations and quickly plan the shortest, collision-free, and less turning safe operation path of the mobile robot in a static grid environment based on multiple planning indexes, so as to solve the path planning problem with more complex environment and task requirements.

In short, the main contributions of this paper are as follows:

- (i) The heuristic median insertion method is designed to generate a feasible initial path and accelerate the algorithm convergence rate
- (ii) The multiobjective fitness function is proposed, which gives different measurement criteria and weights to each indicator according to the planning requirements, so as to ensure that the planned path has the shortest path length, better safety, and smoothness and achieve multiobjective optimization of the algorithm
- (iii) In order to maintain the population diversity in the later stage of the algorithm and avoid premature convergence, the selection operator is designed by the layered method, the crossover operator is designed by the single-point crossover method, and the mutation operator is designed by the eight-neighborhood-domain single-point mutation method
- (iv) The delete operation is added to remove the redundant nodes of the running path to obtain a more optimal path
- (v) On the premise of ensuring that each method is simple and effective, and the optimal path can be generated, the algorithm complexity is reduced and the running speed is accelerated

The rest of this paper is organized as follows. Section 2 reviews the related work on path planning techniques. Section 3 introduces the problem description and environment modeling of path planning. Section 4 describes the path planning method based on improved multiobjective genetic algorithm (IMGGA), including the operating environment model of the mobile robot, the establishment of the initial population, the generation of the multiobjective fitness function, the improved genetic operation, and the redundant point deletion of the optimal path. The simulation results and analysis are given in Section 5. Section 6 summarizes the paper and plans the direction of future work.

## 2. Related Work

In this section, the related work of the existing robot path planning problem is reviewed, and its advantages and disadvantages are discussed.

The commonly used path planning algorithms can be roughly divided into four categories: traditional algorithms, graphics methods, intelligent bionics algorithms, and other algorithms [14]. Among them, the genetic algorithm (GA) is widely used in mobile robot path planning research because of its good scalability, robustness, and fast random search ability [15]. However, the traditional genetic algorithm not only has slow convergence speed but also is easy to fall into premature convergence [16]. Therefore, many researchers have improved the genetic operator to improve the limitations of GA. Zhang et al. [17] have discussed an improved genetic algorithm based on visible space. The main ideas are the concept of visible space, matrix coding, and improved mutation operators. This method is applicable in both static and dynamic environments. However, the smoothness and security of the path are not considered, and the selection operation is very random, which may increase the running time of the algorithm. Lamini et al. [18] have proposed an improved same adjacency crossover operator to solve the robot path planning problem. Considering the variable length of the chromosome, the operator can generate a feasible path with better fitness value and avoid premature convergence. However, the initial population quality of this method still needs to be improved, and the path length does not reach the optimal.

Through the optimization of fitness function, the feasible path can be more in line with the actual requirements. Chen and Chen [19] have used prior knowledge to generate uninterrupted feasible paths and redefine the fitness function, thereby effectively optimizing the genetic algorithm and enabling the algorithm to complete evolution in a shorter time. However, the optimal path generated by this method still has redundant nodes, the operator optimality of the genetic algorithm needs to be improved, and the timeliness and accuracy of the algorithm also need to be enhanced. Cheng et al. [20] regarded path planning as a multiobjective optimization problem and evaluated the performance of the results based on four self-defined fitness objective functions, so as to solve the path planning problem of reconfigurable robot in a complex obstacle environment, but the efficiency of the algorithm still needs to be improved.

With the deepening of research, the effective combination of the path planning algorithm provides the possibility to solve the new cross science problems that a single planning algorithm cannot face. A parallel search algorithm combining the genetic algorithm and artificial potential field method has been proposed by Duan and Chen [21]. This method makes the potential field function jump out of the local minimum point by introducing the filling potential field, and then, the genetic algorithm is built on the improved artificial potential field model to optimize the global path. However, the planned path is too tortuous, and there are a large number of redundant sections. Yi et al. [22] have proposed an improved metaheuristics-adaptive genetic algorithm. The algorithm first uses the random Dijkstra algorithm to create the initial population and then replaces the conventional selection operator with an adaptive operator. Simulation experiments in 2D complex environments prove that the method can effectively avoid the local convergence problem

in path planning. However, this method only considers the length of the path in terms of fitness function, and the method of generating the initial path is too complicated, resulting in the algorithm running time is too long. Sun et al. [23] have proposed an improved adaptive genetic algorithm for robot path planning. Firstly, the idea of simulated annealing is introduced in the selection operation of the genetic algorithm. Then, the self-adjusting strategies of crossover and mutation operators are improved. Finally, the planning indexes such as safe driving speed and turning times are added into the fitness function. Simulation experiments show that the improved algorithm has good efficiency. However, the generation of the initial population needs to be improved, and the index measurement method is too complex, which leads to the increase of the running time of the algorithm.

It is worth noting that although existing studies have improved traditional genetic algorithms to some extent [24, 25], there is still no one method that can fully consider the selection of the initial population, the multiobjective fitness function, and the improvement of genetic operation. Moreover, most of the improved algorithms have high algorithm complexity and low operating efficiency. For this reason, this paper proposes an improved multiobjective genetic algorithm (IMGA), which solves the defects of the traditional genetic algorithm effectively by designing each link of the algorithm. The specific design details will be introduced in the rest of this paper.

### 3. Problem Description and Environment Modeling

In this section, we describe some constraints and environment modeling method for mobile robot path planning.

*3.1. Path Planning Problem Description.* The theoretical complexity of mobile robot path planning is exponentially times of its number of degrees of freedom [26]. Therefore, in the research process of path planning, some specific constraints and assumptions can simplify the solution of the problem and help researchers find out the optimal solution better. In view of this situation, this paper makes the following provisions:

- (1) The mobile robot is regarded as a particle without considering the size of the robot and some factors of kinematics
- (2) The mobile space of the robot is defined as a two-dimensional plane
- (3) All obstacles in the robot's moving space are static obstacles. The shape of the obstacles is not limited, and the size and position of the obstacles are known, and their height is not considered

By simplifying the shape, motion state, and moving space of the robot, the path planning problem is still an NP-hard problem.

*3.2. Environment Modeling.* Using the grid method to model the environment can simply and effectively represent the unknown running space into a clear and intuitive grid map [27]. Therefore, before designing a path planning algorithm for a mobile robot, this paper first uses the grid method to model the running environment of the robot.

Figure 1(a) is a two-dimensional map of the environment without grid processing, which is placed in a rectangular coordinate system. Then, the whole moving space is divided into a  $10 \times 10$  grid diagram, as shown in Figure 1(b). Finally, the obstacles in Figure 1(b) are mapped into corresponding obstacle grids according to the principle that if obstacle is less than a grid, it should be covered with the grid and regarded as a complete obstacle grid. So then, a rasterized environment map of Figure 1(c) is generated. It is obvious that the setting of grid size and number has a great impact on the accuracy of the environmental map. The smaller the unit grid is, the more detailed and accurate the environmental map is.

In Figure 1(c), S, 2, 3, ..., 99 and G are the serial numbers of grid nodes, black is the obstacle grid, white is the free grid, S is the starting point of mobile robot operation, and G is the target point. Since the entire grid map is in a rectangular coordinate system, the serial number of each grid node  $N$  can correspond to a path point coordinate  $(x, y)$ . Assuming that the size of the grid map is  $n \times n$ , the corresponding relationship between the grid serial number and the grid coordinate is

$$\begin{cases} x = \text{mod}(N, n) + 1, \\ y = \text{int}\left(\frac{N}{n}\right) + 1. \end{cases} \quad (1)$$

*3.3. Path Coding.* Path coding is the premise of the genetic algorithm to solve the path planning problem [28]. Through a certain coding method, the feasible solution of the problem can be transformed from its solution space to the search space that the genetic algorithm can handle [29–31]. In order to shorten the encoding length as much as possible, the serial number of the grid node is used for encoding. Any chromosome in the genetic algorithm can find a feasible path from the starting point to the target point on the grid map, and each gene on the chromosome corresponds to the grid number on the feasible path. To ensure the possibility of the path and enhance the operation efficiency of the algorithm, this paper stipulates that the coding sequence number of any feasible path cannot contain the repeated sequence number and the corresponding sequence number of the obstacle grid. As shown in Figure 1, a feasible path can be expressed as S-41-70-G.

## 4. IMGA Algorithm and Analysis

In this section, we describe the improved multiobjective genetic algorithm (IMGA) to plan and select the optimal running path of the mobile robot in the grid static environment.

*4.1. Population Initialization.* The traditional genetic algorithm usually uses a random method to generate the initial population [32]. Although this method is short in time

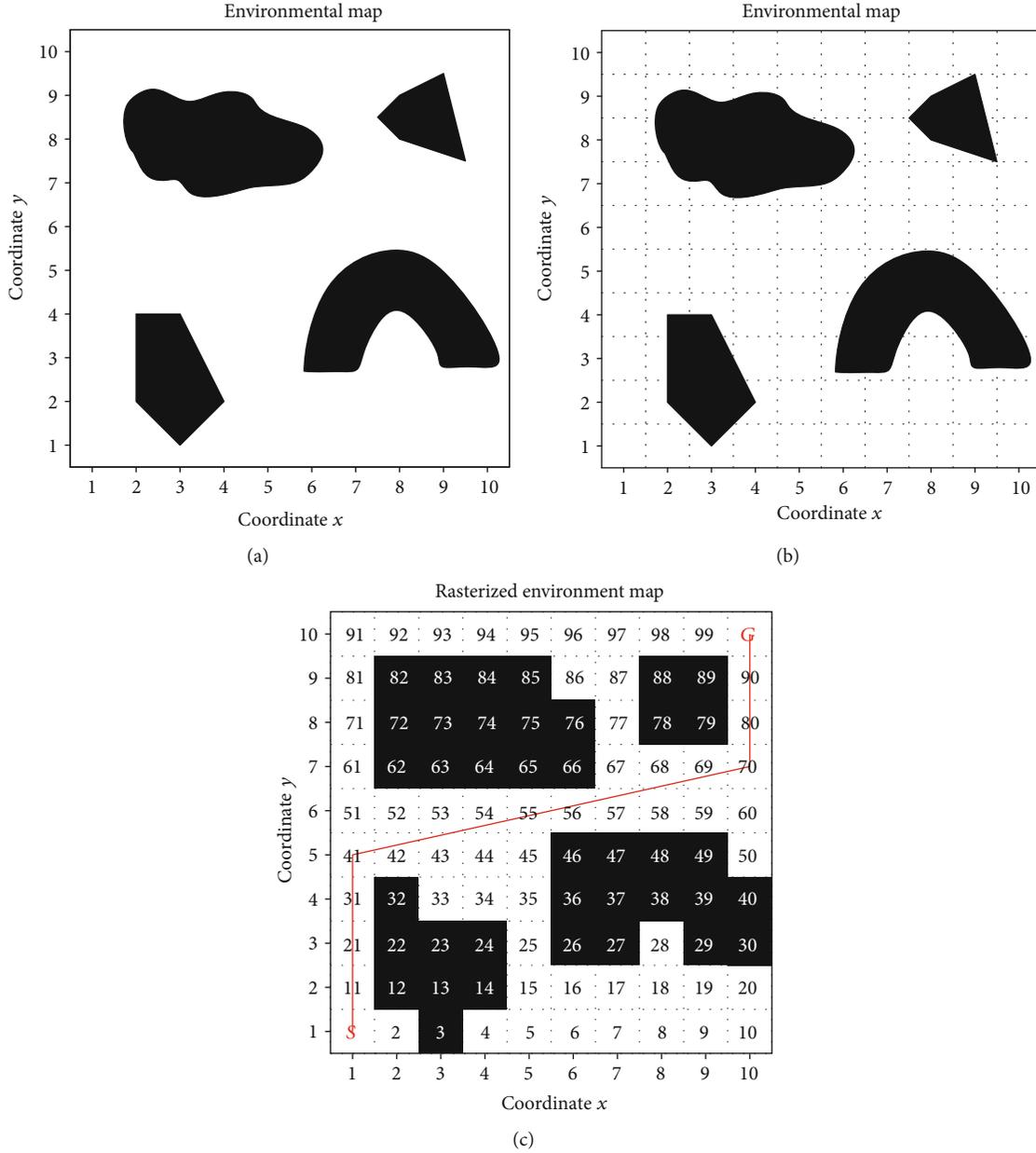


FIGURE 1: Mapping from real environment map to rasterized environment map. (a) Real environment map. (b) Real environment map with grid. (c) Rasterized environment map.

and easy to operate, the proportion of infeasible paths in the generated path is too large, which affects the convergence speed and operating efficiency of the entire algorithm [33]. Aiming at efficiently generating a higher quality initial population and improving the global performance of the algorithm, this paper proposes a heuristic median insertion method to establish the initial population. The specific process of this method is as follows:

- (1) Determine the size of the population  $M$
- (2) Determine the size of the grid map  $n * n$  (representing  $n$  rows and  $n$  columns), the starting point  $S$  of the mobile robot path planning (the starting point  $S$

can be expressed as  $N_1$ ), the target point  $G$  (the target point  $G$  can be expressed as  $N_{n*n}$ ), and the number of obstacle grids  $f$  (the number of free grids is  $n * n - f$ ).

- (3) Generate a path chromosome from the starting point  $S$  to the target point  $G$ , where the starting point  $S$  is always the first gene of the chromosome and the target point  $G$  is always the last gene of the chromosome
  - (i) Randomly generate a grid number  $N_i$  (not belonging to the starting point, ending point, or obstacle grid). At this time, the path of the robot can be expressed as  $S - N_i - G$  (or  $N_1 - N_i - N_{n*n}$ )

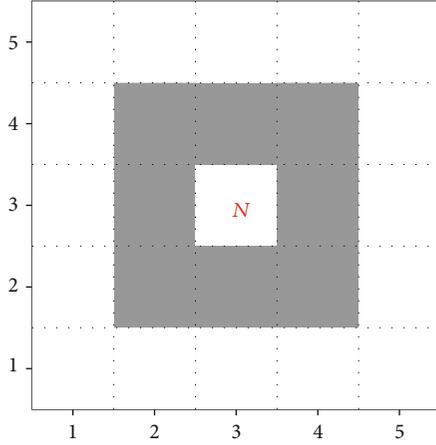


FIGURE 2: Eight neighborhood nodes.

- (ii) Determine whether the adjacent nodes in the path are continuous according to equation (2)

$$\Delta = \max \{ \text{abs}(x_{i+1} - x_i), \text{abs}(y_{i+1} - y_i) \}. \quad (2)$$

In the equation,  $(x_i, y_i)$  and  $(x_{i+1}, y_{i+1})$  represent the rectangular coordinates of two adjacent path points  $N_i$  and  $N_{i+1}$ , respectively. If  $\Delta = 1$ , then  $N_i$  and  $N_{i+1}$  are continuous; otherwise, the two points are discontinuous. At this point, the next insertion point is selected through the median method to fill the discontinuous path. The specific calculation is shown in equation (3):

$$\begin{cases} x_i' = \text{int} \left[ \frac{(x_i + x_{i+1})}{2} \right], \\ y_i' = \text{int} \left[ \frac{(y_i + y_{i+1})}{2} \right], \\ N_i' = x_i' + n \cdot y_i'. \end{cases} \quad (3)$$

In the equation,  $x_i'$  and  $y_i'$  are the coordinates of the candidate grid,  $n$  is the number of rows and columns of the grid, and  $N_i'$  is the number of the candidate grid. If the calculated grid  $N_i'$  is a free grid, it is directly inserted between  $N_i$  and  $N_{i+1}$ ; otherwise, the free grid in the eight neighborhood nodes of  $N_i'$  is randomly selected as the newly inserted node. As shown in Figure 2, the gray area around the  $N$  node is the eight neighborhoods of the point. If there is no free grid in the eight neighborhoods of  $N_i'$ , it indicates that this operation is invalid, and the individual is directly discarded. Repeat the above insertion steps to generate a continuous feasible path.

- (iii) Repeat the above operations until an initial population with  $M$  nonrepeating chromosomes is generated

**4.2. Fitness Function.** After the initial population is established, the algorithm needs to establish a fitness function to

calculate the performance of each individual, so as to determine their pros and cons. Here, the fitness function is equivalent to the objective function of the problem [34]. In order to speed up the convergence of the genetic algorithm while ensuring low complexity of the algorithm, and to find the optimal path that can smoothly avoid obstacles and quickly reach the target point [35], a multiobjective fitness function based on path length, path safety, and path energy consumption is designed in this paper, which is specifically expressed as follows:

$$F(N) = \frac{1}{a \cdot L(N) + b \cdot S(N) + c \cdot E(N)}, \quad (4)$$

where  $L(N)$  is a path length function,  $S(N)$  is a path safety function, and  $E(N)$  is a path energy consumption function.  $a$ ,  $b$ , and  $c$  are the weights of the three indicators. The path length is the sum of Euclidean distances between all adjacent nodes, and the function definition is shown in equation (5).

$$L(N) = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \quad (5)$$

where  $n$  is the number of path nodes, and the path safety function is shown in equation (6).

$$S(N) = \sum_{i=1}^{n-1} S_i, \quad (6)$$

where  $S_i$  is the security penalty value of the node  $i$ . Although it is stipulated in this paper that all available path nodes cannot be obstacle grids, the threat of obstacles to mobile robots still exists. The distance between the robot and the obstacle directly affects its safety. If the distance is too close, it may cause collision or affect the moving speed of the robot. However, it is not better to be as far away from the obstacle as possible. If the robot is too far away from the obstacle, the path length may be too long. Therefore, in this paper, the eight neighborhood domains of a path node are taken as the standard to measure the safe distance between the point and the surrounding obstacles. If there is no obstacle grid in the eight neighborhood domains of a path node, the point is the safe moving point. Otherwise, the penalty value is increased by 5. The path energy consumption function is shown in equation (7).

$$E(N) = \sum_{i=2}^n E_i(l_{i+1}, l_i), \quad (7)$$

where  $E_i$  is the energy consumption penalty value of the node  $i$ . The path energy consumption of a mobile robot mainly refers to the electric energy consumed by the robot in the course of driving. When the robot moves in a straight line, the electric energy consumed by the unit distance is the same. The most power-consuming process is the turning movement occurring when the robot changes the direction of movement. Therefore, the path energy consumption is

actually related to the number of turns and the angle of turning, and it is the least energy consumption only if it keeps moving towards the target point. As shown in Figure 3, if the robot turns at point  $i$ , then point  $i$  is taken as the origin, and the direction of the horizontal and vertical coordinates of point  $i$  is taken as the dividing line to divide the quadrant. When the robot turns and its forward direction is in the first quadrant, the energy penalty  $E_i$  is plus 1; when the robot's forward direction is in the second or fourth quadrant, the energy penalty  $E_i$  is plus 3; when the direction of the robot is in the third quadrant, plus 10. In this way, the number of turns of the robot can be reduced, and the smoothness of the entire planned path can be enhanced; it can also ensure that the robot always advances to the target point during the running process and does not make unnecessary deviation movements.

According to the construction of the fitness function, it can be known that the higher the function value, the better the path.

### 4.3. Genetic Operation

**4.3.1. Selection Operation.** In genetic operation, selection operation is the part that best embodies "survival of the fittest." In order to avoid the nondirectional error caused by the traditional roulette selection method [36], this paper adopts the layered method to design selection operator, which is described as follows: after initializing the population, a total of  $M$  individuals are generated, the fitness value of each individual is calculated according to the fitness function, and the fitness values are arranged in descending order. Then, the population can be divided into three classes, each part is a layer, namely, a layer has  $M/3$  individuals. Copy the aliquot with the higher fitness value in the first layer in duplicate, make a copy with the middle fitness value in the second layer, and do not copy the last fitness with the lower fitness, in order to form the offspring population. And the number of population is still  $M$  (if  $M$  is not divisible by 3, the number of individuals in the last share is  $M/3 + \text{mod}(M/3)$ ). After replicating the first two layers,  $\text{mod}(M/3)$  individuals with high fitness value should be selected from the third layer to join the offspring population in order to ensure that the number of the offspring population is still  $M$ ). This ensures that the best individuals can be passed on to the next generation while maintaining the diversity of the population. The selection method can be seen in Figure 4.

**4.3.2. Crossover Operation.** Crossover operation is the most important genetic operation in the whole algorithm, and its essence is genetic recombination of chromosomes. That is, by exchanging some of the genes of the parent chromosome, new progeny chromosomes are created. This paper adopts the method of single-point crossing. Randomly select one of the two parent individuals with the same sequence number (except the starting point  $S$  and the target point  $G$ ) and then cross at this point. For example: parent individual  $M_1$ : S-2-12-67-69-89-G;  $M_2$ : S-11-22-67-78-80-G. If the same sequence number 67 is selected as the intersection point, the two offsprings generated are  $M_1'$ : S-2-12-67-78-80-G;

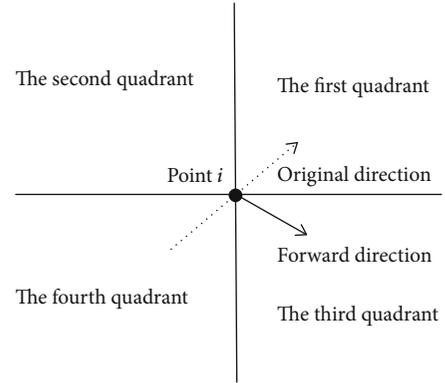


FIGURE 3: The quadrant diagram corresponding to the turning direction of the robot.

$M_2'$ : S-11-22-67-69-89-G. If the same sequence number does not exist in two parent individuals, no crossover operation is performed.

**4.3.3. Mutation Operation.** The mutation operation is to mutate any gene on an individual's chromosome to generate a new chromosome to maintain the diversity of the population. However, traditional single-point mutation, multipoint mutation, and other processing methods are prone to produce infeasible paths, which will affect the operation efficiency of the algorithm. Therefore, this paper uses the method of eight-point neighborhood single-point mutation. First, randomly select a variation point  $N_i$  (except the starting point  $S$  and target point  $G$ ) in the individual of the path to be mutated and then randomly determine a nonobstacle grid  $N_i'$  (except the adjacent nodes  $N_{i-1}$  and  $N_{i+1}$  in the path to be mutated) in the eight neighborhoods of the mutation point (as shown in Figure 2) to replace the original node. Then, according to the initial path generation method,  $N_{i-1}$  to  $N_i'$  and  $N_i'$  to  $N_{i+1}$  are connected into an uninterrupted path. If there is no free grid to choose from in the eight neighborhoods of the mutation points or a feasible uninterrupted path cannot be generated, this mutation fails. That is, jump out of this mutation operation and reselect the next individual to be mutated and its mutation point.

**4.4. Termination Conditions.** The termination condition is a standard to measure whether the genetic algorithm can terminate the operation. The termination condition given in this paper is as follows: the optimal fitness value for a given evolutionary algebraic threshold of 50 or 40 consecutive evolutionary populations remains unchanged or the algorithm runs longer than 5 minutes.

**4.5. Delete Operation.** For the case of redundant nodes in the path, a delete operation is added in this article. The main idea is if a node can be barrier-free connected to its front and rear nodes (nonadjacent path points), then the intermediate nodes between these two points are redundant. Delete these redundant nodes and connect these two points directly, so as to achieve the goal of avoiding unnecessary turning and reducing the path length. To avoid the reduction of path

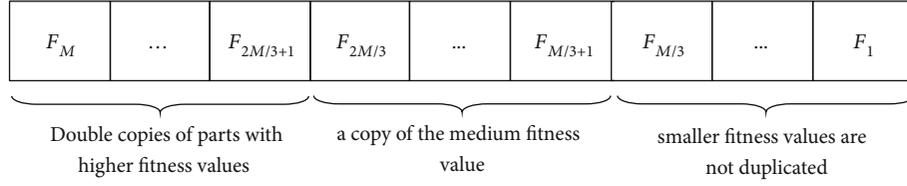


FIGURE 4: Fitness segmentation selection principle diagram.

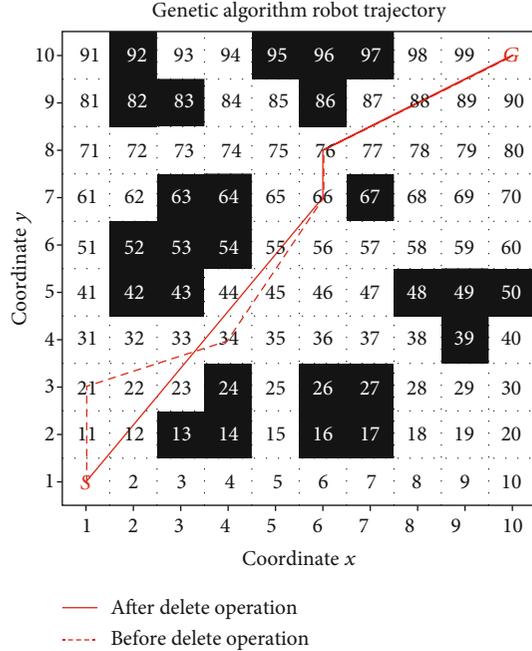


FIGURE 5: Path comparison before and after deletion.

**Input:** Grid map, Map; Start point, S; End point, G  
**Output:** Shortest path point set, min\_path

- 1:  $t \leftarrow 0$ ;
- 2: Evaluate fitness\_value of  $P(t)$ ;
- 3: Best\_fitness  $\leftarrow \max(\text{fitness\_value})$ ;
- 4: **while** not meet termination conditions **do**
- 5: Select operation to  $P(t)$ ;
- 6: Crossover operation to  $P(t)$ ;
- 7: Mutation operation to  $P(t)$ ;
- 8:  $P(t+1) = P(t)$ ;
- 9:  $t = t + 1$ ;
- 10: Evaluate fitness\_value of  $P(t)$ ;
- 11: **if** Best\_fitness < max (fitness\_value) **then**
- 12: Best\_fitness  $\leftarrow \max(\text{fitness\_value})$ ;
- 13: **end if**
- 14: **end while**
- 15: min\_path is  $P(t)$  with Best\_fitness;
- 16: Smooth optimization operation to min\_path;
- 17: **return** min\_path;

ALGORITHM 1: IMGGA Algorithm.

nodes after the delete operation, which affects the feasibility of crossover and mutation operations in genetic operations, this process is only for the optimal paths obtained by each

run. As shown in Figure 5, the original optimal path is S-21-34-66-76-G, and the optimal path after delete operation is S-66-76-G. It is obvious that there are four turns in the original path. After the deletion operation, the redundant nodes 21 and 34 in the section from S point to sequence point 66 are deleted, and only two turns occur in the whole path, and the path length becomes shorter.

**4.6. Algorithm Pseudocode.** The pseudocode of the IMGGA algorithm is shown below. We can see that the time complexity of the whole algorithm is  $O(O_F * (O_{ps} + O_{pc} + O_{pm}))$ , where  $O_F$  represents the complexity of fitness function,  $O_{ps}$  represents the time complexity of selection operation,  $O_{pc}$  represents the complexity of crossover operation, and  $O_{pm}$  represents the time complexity of mutation operation. In other words, the time complexity of IMGGA is determined by both genetic operator and fitness function. Compared with the traditional roulette method, the hierarchical selection method proposed in this paper has less time complexity. Although the fitness function considers multiple indicators, the measurement method of each index is relatively simple. In conclusion, the time complexity of IMGGA is not higher than that of traditional GA.

## 5. Simulation Experiment and Evaluation

In this section, to verify the rationality and optimization degree of the designed improved multiobjective genetic algorithm, MATLAB R2018b software was used to establish a grid model and carry out simulation experiments on the algorithm.

**5.1. Experimental Result.** The simulation experiment is first carried out in the grid environment model of  $10 \times 10$ ,  $15 \times 15$ , and  $20 \times 20$  to verify the universality of the algorithm in different maps. The parameters set in the experiment are as follows: population number: NP = 100; maximum evolutionary algebra: max\_gen = 50; crossover probability: PC = 0.8; mutation probability: PM = 0.2; path length proportion:  $a = 6$ ; path safety proportion:  $b = 1$ ; and path energy proportion:  $c = 3$ . The experimental results are shown in Figures 6(a)–7(a). In Figure 6(a), there are 40 obstacle grids, Figure 6(b) has 90 obstacle grids, and Figure 7(a) has 160 obstacle grids. The dotted line represents the running path before delete operation, and the solid line represents the running path after delete operation. According to the statistical results in Table 1, the delete operation designed by this algorithm can effectively reduce the path length. Among them, the number of turns of the running paths with delete operation in environment 2 and environment 3 is also reduced.

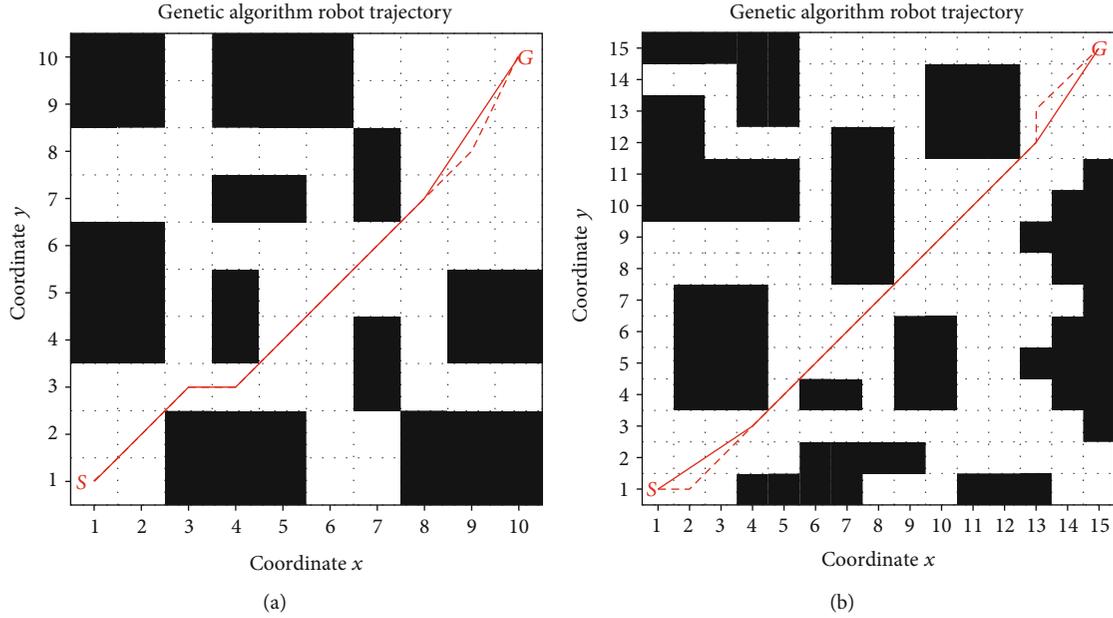


FIGURE 6: Optimal running path. (a) Environment 1. (b) Environment 2.

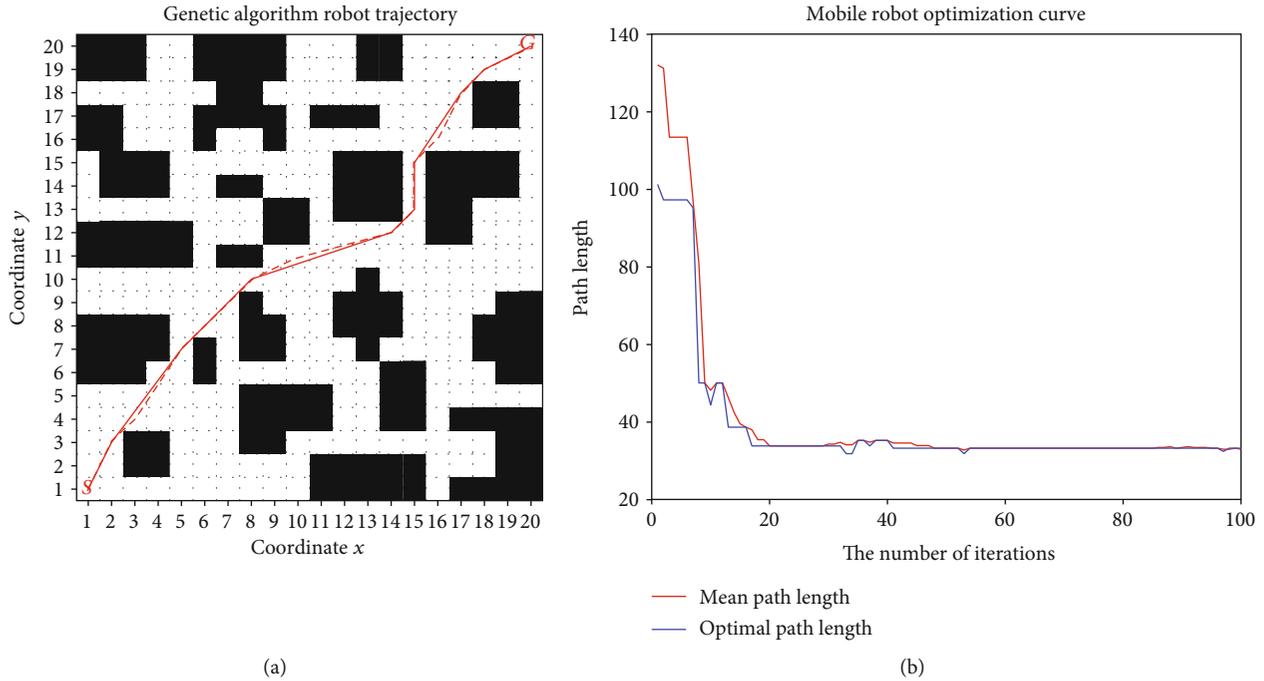


FIGURE 7: Environment 3. (a) Optimal running path. (b) Optimization curve.

Although the number of turns of environment 1 has not changed, it can be clearly seen from Figure 6(a) that the angle of the third turning of the running path with deletion operation is significantly smaller. Therefore, this operation can effectively optimize the path and further ensure the efficient operation of the robot. It is an indispensable link in the operation of the entire algorithm.

Taking environment 3 as an example, the influence of the maximum evolution algebra on the algorithm running

results is analyzed. Ensure that other parameters remain unchanged, set different maximum evolutionary generation values, and run 10 times, respectively. The experimental results are shown in Table 2, where “-” indicates that the algorithm did not obtain the optimal solution after running 10 times. It can be seen that the larger the maximum evolution algebra, the better the search ability and search stability of the algorithm, but the timeliness is poor, and the algorithm runs longer. In order to search the optimal running path of

TABLE 1: Comparison of run path data with or without delete operation.

Operating environment	Path length		Number of turns	
	No deletion operation	With deletion operation	No deletion operation	With deletion operation
Environment 1	13.1356	13.0908	3	3
Environment 2	20.3848	19.9390	3	2
Environment 3	28.5724	28.4733	11	8

TABLE 2: The influence of maximum evolution algebra on the running result of the algorithm.

Maximum evolution algebra	The number of runs to get the optimal solution	The total number of occurrences of the optimal solution in the results of 10 runs	Optimal running time (s)
20	—	0	0.59834
50	5	2	0.71417
100	3	4	0.80262
200	2	5	0.98274

the mobile robot as quickly as possible, this paper selects 50 as the maximum evolution algebra value.

Figure 7(b) shows the evolution curve of the algorithm running in environment 3 of Figure 7(a). It can be seen that in this environment, the algorithm designed in this paper tends to be stable when it evolves to the 10th generation, and the algorithm has a fast convergence speed. Moreover, under the regulation of the genetic operation, the path length did not fall into the local optimal immediately but fluctuated slightly and found the global optimal path successfully. It can be proved that the initial population quality of the algorithm designed in this paper is very high, and the fitness function is set reasonably, which is helpful for the algorithm to converge quickly. In the late stage of evolution, the diversity of the population is guaranteed while the stability of the population is maintained. It avoids the problem that the traditional genetic algorithm easily falls into the local optimal.

**5.2. Algorithm Comparative Analysis.** In order to show the superiority of the proposed algorithm (IMGA) in solving the path planning problem of the mobile robot, IMGA is compared with the improved genetic algorithms in literature [37] and literature [38].

Under the premise of the same control parameters, multiple simulation experiments were carried out in the environment mentioned in literature [37]. The optimal running path of the algorithm in this paper is shown in Figure 8(c). Figure 8(a) is the optimal operation of the basic genetic algorithm (GA), and Figure 8(b) is the optimal operation of the improved genetic algorithm proposed in literature [37]. The optimal path length, turn times, and tangency with obstacles

of the three methods are recorded in Table 3. Through the analysis of the operation results, it can be seen that the path obtained by the GA has a large number of turns and redundant path segments, and there are two places pass through the tiny gap between the obstacles, as shown in the blue circle of Figure 8(a). Although the improved algorithm proposed in literature [37] eliminates some redundant path segments and unnecessary turns, there is still a place that passes through the minimal gap between two obstacles, which is very unfavorable for the actual operation of mobile robots. Considering the path length, path security, and path energy consumption, IMGA can avoid the situation of blue circles in Figures 8(a) and 8(b). The path length is reduced by 17% compared with GA, and there are only three places that are tangent to obstacles. Therefore, under the condition of ensuring the optimal path length, the algorithm IMGA in this paper can effectively improve the operation safety of the mobile robot and reduce the number of turning movements of the robot. The improved effect is better than other improved algorithms.

Then, IMCA and Bezier curve smoothing algorithm (BCA) proposed by literature [38] are compared in a more complex environment under the same parameters. The optimal running path is shown in Figure 9. To eliminate the influence of random factors on the algorithm, the above two algorithms were independently executed 30 times, and the statistical results are recorded in Table 4. BCA first obtains the segmented Bezier curves control points (blue square dots in Figure 9) through genetic operations and then plans the optimal smooth path according to the optimization criteria, which is actually a combination of Bezier curve and GA. According to Table 4, compared with BCA, IMGA can obtain a running path with a smaller path length, and the algorithm running time is only five thousandths of BCA. However, as can be seen from Figure 9, the optimal path planned by IMGA is a broken line. Compared with the BCA algorithm, the broken line is smoother and the path length is shorter. However, there are three (one in the middle is not tangent) times tangent to the barrier grid, and the optimal operation path obtained by BCA planning is a smooth curve that is not tangent to any obstacle. Therefore, IMGA still has space for improvement in path safety and smoothness, but it has great advantages in environments with strong real-time requirements.

## 6. Conclusion

In this paper, an improved multiobjective genetic algorithm is designed to study the path planning problem of the mobile robot. The algorithm uses the heuristic median insertion method during the creation of the initial population, which effectively improves the quality of the initial population and speeds up the convergence rate of the algorithm, and the path length, path safety, and path energy consumption are used as evaluation indicators to establish a multiobjective fitness function, which minimizes the path length while ensuring that the robot always moves toward the target point, avoids unnecessary path energy consumption, and improves the safety of planned paths to a certain extent. Finally, the

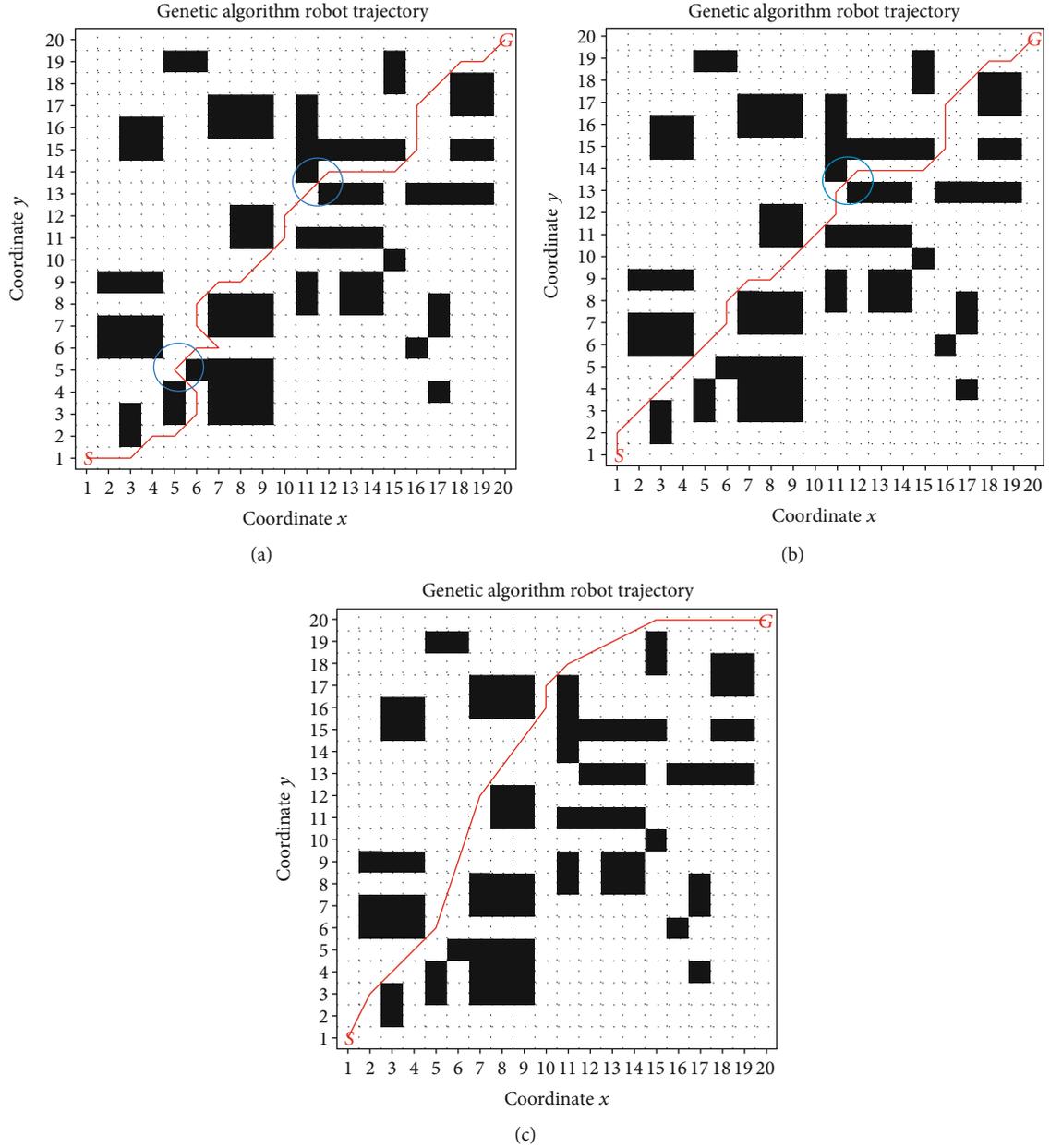


FIGURE 8: Optimal running path. (a) Optimal running path of GA. (b) Optimal running path of literature [37]. (c) Optimal running path of IMGA.

TABLE 3: Path planning data statistics in environment 4.

Path planning algorithm	Optimal path length	Number of turns	Number of tangency
GA	35.7990	20	10
Literature [37]	29.7990	13	8
IMGA	29.6896	7	3

improved genetic operation is used to further prevent the algorithm from falling into local optimum prematurely. Simulation experiments show that the algorithm can run in environments of different scales and complexity. Compared with other improved algorithms, this method has faster conver-

gence speed and shorter path length. Therefore, this algorithm has certain feasibility and superiority in the field of path planning problems for the mobile robot. However, there is still space for improvement in the security and smoothness of the optimal path planned. This is because the algorithm takes the path length as the main optimization objective and pays more attention to the operation efficiency in the planning process and adopts a relatively simple method to measure the three indicators of fitness function.

There are still many meaningful topics to be discussed in this algorithm, and the future research directions are as follows: (1) better design of fitness function to improve the overall quality of the planning path, especially the path safety on the premise of ensuring operation efficiency; (2)

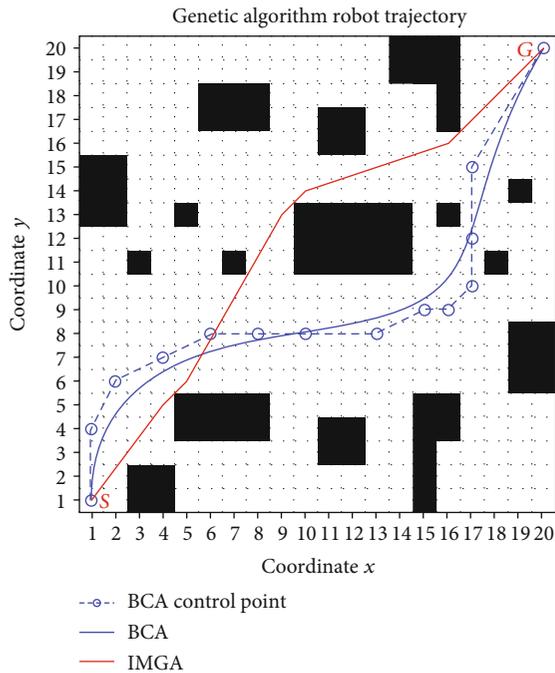


FIGURE 9: Comparison chart of algorithm optimal path.

TABLE 4: Path planning data statistics in environment 5.

Path planning algorithm	Optimal path length	Optimal running time(s)
IMGA	27.8721	0.7016
BCA	29.9416	136.4794

combining the algorithm with Bezier curve or other curve smoothing methods, the polygonal path is planned to be a smoother curve path; (3) increase complexity design, for example, consider the specific model and kinematics factors of mobile robot, design more complex operation environment, and add dynamic obstacles; and (4) apply path planning algorithm to solving scenarios of practical problems, such as express shunting robot, family service robot, and automatic driving vehicle.

### Data Availability

A large number of data are included in the paper, and other relevant information can be obtained from the corresponding authors through e-mail.

### Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

### Acknowledgments

This work was supported by the Science and Technology Project of Emergency Management in Jiangsu Province

(Grant No. YJGLYF202017), in part by the Research Projects in Education Informatization in Jiangsu Province (Grant No. 20180012), in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province (Grant No. KYCX18\_2366), in part by the Yangzhou Science and Technology (Grant Nos. YZ2018209 and YZ2019133), in part by the Yangzhou University Jiangdu High-End Equipment Engineering Technology Research Institute Open Project (Grant No. YDJD201707), and in part by the Open Project in the State Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University (Grant No. #1907).

### References

- [1] A. Bakdi, A. Hentout, H. Boutami, A. Maoudj, O. Hachour, and B. Bouzouia, "Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control," *Robotics and Autonomous Systems*, vol. 89, pp. 95–109, 2017.
- [2] Y. Xue, "Mobile robot path planning with a non-dominated sorting genetic algorithm," *Applied Sciences*, vol. 8, no. 11, p. 2253, 2018.
- [3] H. Zhang, W. Lin, and A. Chen, "Path planning for the mobile robot: a review," *Symmetry*, vol. 10, no. 10, p. 450, 2018.
- [4] J. Chen, S. Xie, H. Li, J. Luo, and K. Feng, "Robot path planning based on adaptive integrating of genetic and ant colony algorithm," *International Journal of Innovative Computing Information and Control*, vol. 11, no. 3, pp. 833–850, 2015.
- [5] H. Lu, Y. Li, S. Mu, D. Wang, and S. Serikawa, "Motor anomaly detection for unmanned aerial vehicles using reinforcement learning," *IEEE Internet of Things Journal*, vol. 1, no. 1, p. 99, 2017.
- [6] C. Ge, Z. Liu, J. Xia, and F. Liming, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 99, p. 1, 2019.
- [7] X. Zhan, "Research on path planning method of humanoid robot based on improved genetic algorithm," *Journal of Physics: Conference Series*, vol. 1237, no. 2, pp. 22–28, 2019.
- [8] N. S. Utami, A. Jazidie, and R. E. A. Kadier, "Path planning for differential drive mobile robot to avoid static obstacles collision using modified crossover genetic algorithm," in *2019 International Seminar on Intelligent Technology and Its Application (ISITIA)*, pp. 282–287, Surabaya, Indonesia, 2019.
- [9] Q. Xiao, J. Li, and C. Xiao, "Research on performance optimization of several frequently-used genetic algorithm selection operators," in *International Conference on Applications and Techniques in Cyber Security and Intelligence*, vol. 580 of *Advances in Intelligent Systems and Computing*, pp. 90–97, Ningbo, China, 2018.
- [10] H. Lu, D. Wang, Y. Li et al., "CONet: a cognitive ocean network," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 90–96, 2019.
- [11] Z. Liu, Y. Xue, C. Zhou, H. Bai, and X. Cui, "Population initialization improvement of robot path planning based on genetic algorithm," *Machine Tool & Hydraulics*, vol. 47, no. 21, pp. 5–8, 2019.
- [12] N. Wang, H. Guo, B. Chen, C. Cui, and X. Zhang, "Design of a rotary dielectric elastomer actuator using a topology optimization method based on pairs of curves," *Smart Materials and Structures*, vol. 27, no. 5, article 055011, 2018.

- [13] B. Song, Z. Wang, L. Zou, L. Xu, and F. E. Alsaadi, "A new approach to smooth global path planning of mobile robots with kinematic constraints," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 107–119, 2019.
- [14] M. Albayrak and N. Allahverdi, "Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1313–1320, 2011.
- [15] L. Fang, W. Susilo, Y. Ren, C. Ge, and J. Wang, "Chosen public key and ciphertext secure proxy re-encryption schemes," *International Journal of Digital Content Technology and Its Applications*, vol. 4, no. 9, pp. 151–160, 2010.
- [16] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and F. Liming, "Secure keyword search and data sharing mechanism for cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 99, pp. 1–1, 2020.
- [17] X. Zhang, Y. Zhao, N. Deng, and K. Guo, "Dynamic path planning algorithm for a mobile robot based on visible space and an improved genetic algorithm," *International Journal of Advanced Robotic Systems*, vol. 13, no. 3, p. 91, 2016.
- [18] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," *Procedia Computer Science*, vol. 127, pp. 180–189, 2018.
- [19] L. Chen and J. R. Chen, "Mobile robot path planning based on improved genetic algorithm," *Software Guide*, vol. 18, no. 4, pp. 24–27, 2019.
- [20] K. P. Cheng, R. E. Mohan, N. H. Khanh Nhan, and A. V. le, "Multi-objective genetic algorithm-based autonomous path planning for hinged-tetro reconfigurable tiling robot," *IEEE Access*, vol. 8, no. 1, pp. 121267–121284, 2020.
- [21] J. Duan and Q. Chen, "Research on path planning algorithm based on improved artificial potential field-genetic algorithm," *Foreign Electronic Measurement Technology*, vol. 38, no. 3, pp. 19–24, 2019.
- [22] X. Yi, W. S. Guo, and L. Zhao, "Robot motion based on adaptive selection operator combined with genetic algorithm," *Application Research of Computers*, vol. 37, no. 6, pp. 1–6, 2020.
- [23] B. Sun, P. Jiang, G. Zhou, and Y. T. Lu, "Application of improved genetic algorithm in path planning of mobile robots," *Computer Engineering and Applications*, vol. 55, no. 17, pp. 162–168, 2019.
- [24] K. Liang, Z. Chen, and X. Yan, "Simulation study on effective path planning for mobile robot," *Modern Electronic Technology*, vol. 41, no. 17, pp. 167–172, 2018.
- [25] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1564–1572, 2012.
- [26] A. Tharwat, M. Elhoseny, A. E. Hassanien, T. Gabel, and A. Kumar, "Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm," *Cluster Computing*, vol. 22, no. S2, pp. 4745–4766, 2019.
- [27] J. C. Mohanta, D. R. Parhi, and S. K. Patel, "Path planning strategy for autonomous mobile robot navigation using Petri-GA optimisation," *Computers & Electrical Engineering*, vol. 37, no. 6, pp. 1058–1070, 2011.
- [28] S. Ghosh, P. K. Panigrahi, and D. R. Parhi, "Analysis of FPA and BA meta-heuristic controllers for optimal path planning of mobile robot in cluttered environment," *IET Science, Measurement & Technology*, vol. 11, no. 7, pp. 817–828, 2017.
- [29] L. Wang, M. Li, J. Cai, and Z. Liu, "Research on mobile robot path planning by using improved genetic algorithm," *Mechanical Science and Technology*, vol. 36, no. 5, pp. 711–716, 2017.
- [30] S. Serikawa and H. Lu, "Underwater image dehazing using joint trilateral filter," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 41–50, 2014.
- [31] H. Lu, Y. Li, M. Chen, H. Kim, and S. Serikawa, "Brain intelligence: go beyond artificial intelligence," *Mobile Networks and Applications*, vol. 23, pp. 368–375, 2018.
- [32] F. Han, J. Jiang, Q. Ling, and B. Su, "A survey on metaheuristic optimization for random single-hidden layer feedforward neural network," *Neurocomputing*, vol. 335, pp. 261–273, 2019.
- [33] C. Ge, W. Susilo, L. Fang, J. Wang, and Y. Shi, "A CCA-secure key-policy attribute-based proxy re-encryption in the adaptive corruption model for dropbox data sharing system," *Designs, Codes and Cryptography*, vol. 86, no. 11, pp. 2587–2603, 2018.
- [34] Y. Li, Z. Huang, and Y. Xie, "Path planning of mobile robot based on improved genetic algorithm," in *3rd International Conference on Electron Device and Mechanical Engineering (ICEDME)*, pp. 691–695, Suzhou, China, 2020.
- [35] J. Ma, Y. Liu, S. Zang, and L. Wang, "Robot path planning based on genetic algorithm fused with continuous Bezier optimization," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 981304, 10 pages, 2020.
- [36] M. Nazarahari, E. Khanmirza, and S. Doostie, "Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm," *Expert Systems with Applications*, vol. 115, pp. 106–120, 2019.
- [37] S. Huang, S. Jiang, and X. Zhu, "Multi-object tracking via discriminative appearance modeling," *Computer Vision and Image Understanding*, vol. 153, pp. 77–87, 2016.
- [38] Q. Ling, Y. Song, F. Han, D. Yang, and D. Huang, "An improved ensemble of random vector functional link networks based on particle swarm optimization with double optimization strategy," *PLoS One*, vol. 11, no. 11, article e0165803, 2016.