Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9590367, 8 pages
https://doi.org/10.1155/2022/9590367

Research Article

An Effective Path Planning of Intelligent Mobile Robot Using Improved Genetic Algorithm

Zhongzhe Chen,[1,2] **Jianzhang Xiao**,[1,2] **and Guifeng Wang**[1,2]

[1] *College of Mechanical and Electrical Engineering, Jinhua Polytechnic, Jinhua, 321007 Zhejiang, China*
[2] *Key Laboratory of Crop Harvesting Equipment Technology of Zhejiang Province, Jinhua Polytechnic, Jinhua, 321007 Zhejiang, China*

Correspondence should be addressed to Zhongzhe Chen; 20151030@jhc.edu.cn

Received 25 January 2022; Revised 22 March 2022; Accepted 4 April 2022; Published 9 May 2022

Academic Editor: Shalli Rani

With the rapid development of the robotics industry, the problem of effective and fast path planning for intelligent mobile robots has always been one of the hot spots in the field of robotics research. Intelligent mobile robot path planning is divided into global path planning and local path planning, and its mathematical modeling and adaptive algorithms are different. Therefore, the research of robot path planning based on improved genetic algorithm is of great significance. This paper mainly studies the robot path planning problem based on improved genetic algorithm. Based on the research of the basic genetic algorithm, the improved genetic algorithm is applied to the mobile four-wheel robot to guide the four-wheel robot to complete path planning and other related tasks. Experiments show that the optimization probability and convergence speed of the genetic algorithm can be improved by improving the genetic algorithm. Studies have shown that evolutionary algebra and population size are inversely proportional to the optimal path length, so it is directly proportional to the search ability. However, as the evolutionary algebra and population size increase, the amount of calculation is also increasing, and the calculation time increases. Comprehensive considerations according to various factors, the best value of population size is 60, the best value of mutation probability is 0.09, the best value of crossover probability is 0.8, and the best value of evolutionary algebra is 150 generations.

1. Introduction

With the development of the times, it is more and more difficult for robots fixed in one place to do the same job to meet the requirements of robots in various industries [1, 2], and the development of navigation technology makes it possible for robots to handle more complex things. Vision-based navigation research will create conditions for mobile robots to move independently from human control and is one of the basic tasks for mobile robots to play a greater role [3, 4]. The problem of path planning is the core problem of navigation. The quality of the feasible path planned by the path planning system is directly related to whether the robot can bypass all obstacles and reach the target point safely and quickly. It is the key link of the navigation system [5–6]. Therefore, mobile robot technology will usher in new development opportunities under the promotion of a large amount of government investment and enterprise demand [7, 8].

In the research of robot path planning based on improved genetic algorithm, many scholars have studied it and achieved good results. For example, the autonomous mobile robot designed by Qi et al. adopts two wheel differential speeds. In order to adjust the movement direction, a laptop is used as the control center, which can be controlled remotely or move autonomously [9]. There are four modular configuration schemes for free selection, which can meet different needs and adapt to different workplaces. The CASIA-1 mobile robot studied by Gao et al. has 32 infrared sensors, which are evenly installed around the fuselage at an angle of 22.5 degrees. One-half of them are used to detect obstacles at close range. One-half is used to detect long-distance obstacles. The robot also has ultrasonic sensors, and the obstacle information obtained by two different types of sensors is fused [10].

This article uses the literature research method to retrieve literature, works, articles, etc. related to keywords such as "genetic algorithm," "path planning," and "intelligent mobile robots," then uses a comparative experiment method to talk about the robots before and after improvements. The effective path ratio is compared.

Through more in-depth research and improvement of genetic algorithm and trying to combine it with other algorithms, we can achieve better results in path planning, which is of great practical significance to expand the application range of mobile intelligent robot.

2. Robot Path Planning Based on Improved Genetic Algorithm

2.1. Principle of Robot Path Planning Based on Genetic Algorithm. In this paper, the genetic algorithm uses the grid method to mark the geographical environment information and maps the environmental information to the grid. Black represents obstacles, and white represents obstacles. Combine the grid method with the array, the array $a[x][y]$ represents the grid, $a[0][0]$ represents the upper left square, and $a[X][Y]$ represents the lower right square; X and Y are the maximum number of horizontal and vertical grids. Import the obstacle information into the array, $a[x][y] = 0$ means that there is no obstacle at the square.

2.1.1. Initialization. Randomly generate a certain number of codes of a certain length as the initial population, which represents the path of the robot from the start point to the end point [11]. The robot path planning of the genetic algorithm is to perform genetic operations on these paths, and then generate the optimal path group, and output the most optimal path [12].

The robot plans the path according to the path code. If it encounters obstacles during walking, it will judge that the current path is invalid and gives up. If the robot can reach the end point, it is judged that the current solution is a feasible solution [13]. Of course, the randomly generated path code is not a feasible solution. Whether most randomly generated path codes are failed path codes, that is, the robot cannot complete the movement task from the starting point to the end point in the direction indicated by it.

However, in the same way, it is not to say that the failed solution generated at the beginning can not become a feasible solution after a series of genetic operations such as crossover, mutation, deletion, and insertion. Similarly, in terms of the fitness function value, the value of the infeasible path is greater than the value of the feasible path; that is, the infeasible solution is better than the feasible solution in the individual evaluation process, indicating that the infeasible solution is more effective than the feasible solution in a certain range. If the infeasible solution is further optimized to make it an excellent feasible solution, the optimal solution obtained in this way is also feasible. Therefore, in the individual evaluation process, different evaluation criteria can be used for feasible and infeasible paths to further improve the degree of superiority of the solution population [14, 15].

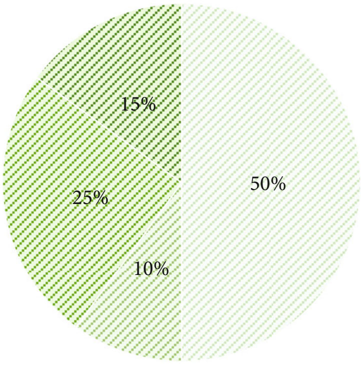

Figure 1: Roulette.

2.1.2. Genetic Operator. Genetic operators include individual evaluation, single-point crossover, multipoint crossover, deletion operators, and so on.

2.1.3. Individual Evaluation. The idea of roulette selection is applied to the process of constructing the fitness function of robot path planning. The idea of roulette selection is to use a pie chart to represent the roulette used for gambling, as shown in Figure 1.

Assuming that the robot path code is a certain block in a specified pie chart, the size of the block in the pie chart is proportional to the size of the fitness function value of the path code. The higher the fitness function value of the path code, the larger the area occupied in the pie. When imitating gambling, put a small ball on the roulette wheel to see where the ball stops, that is, select the path code corresponding to it. The probability that the path code is selected is proportional to its fitness function value.

A fitness function based on a penalty function can also be used, and the fitness function is often set as a penalty function in path planning. For example, let the penalty function be Fit$Q$.

$$\begin{aligned} \mathrm{Fit}Q &= \frac{\mathrm{Fit}q2}{\mathrm{Fit}q1}, \\ \mathrm{Fit}q2 &= \begin{cases} 1\ R = 1, \\ 0\ R = 0, \end{cases} \\ \mathrm{Fit}q1 &= \sum_{i=2}^{N_P} \mathrm{Dis}(Pt_i, Pt_{i-1}) + \mathrm{Dis}\left(ST_p, Pt_i\right) \\ &\quad + \mathrm{Dis}(Pt_{NP}, ET_P) + M \times F_{cst}. \end{aligned} \tag{1}$$

R is defined as the path safety factor. If the robot is already in the danger zone, the safety distance is greater than the linear distance between the robot and the obstacle, then $R = 0$; if the robot is already in the safety zone, the safety distance is less than the linear distance between the robot and the obstacle, then $R = 1$.

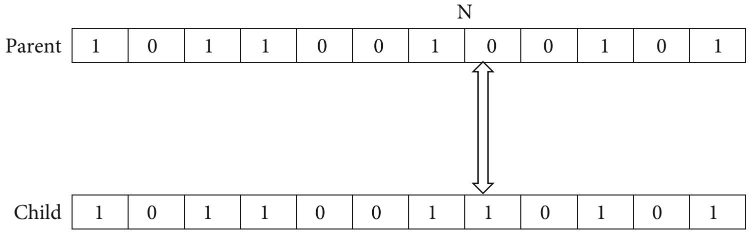

Figure 2: Single point mutation.

N_P represents the number of grids in the path (excluding the start point and end point), Pt_i represents the i-th grid, ST_p and ET_P represent the start point and end point. $\mathrm{Dis}(X, Y)$ represents the straight-line distance between the starting point (X_a, Y_b) and the end point (X_A, Y_B), expressed as follows:

$$\mathrm{Dis}(X, Y) = \sqrt{(X_a - X_A)^2 + (Y_b - Y_B)^2}. \tag{2}$$

M is the penalty coefficient, and F_{cst} is the penalty function.

$$F_{cst} = \sum_{i=1}^{N_{ab}} \left(W(i) + kWL'(i) \right). \tag{3}$$

Define $W(i)$ as the number of paths that pass through the grid. K is a proportional coefficient, which can be a constant.

Definition $L'(i)$ is the shortest distance between the path and the obstacle and the number of grids from the obstacle to the path as a vertical line.

The robot moves from grid $a[X_a][Y_b]$ to $a[X_A][Y_B]$ time T, the average speed of the robot V. This method is generally applicable to square grids.

$$T = \frac{\sqrt{(X_a - X_A)^2 + (Y_b - Y_B)^2}}{V}. \tag{4}$$

In some genetic algorithms, the fitness function value of the robot path planning is modeled according to the distance between the robot and the end point. The binary code of each path is compiled, and then, the result obtained is put into the path test base class to judge the distance between the robot and the end point and finally returns a function value.

2.1.4. Single Point Mutation. Randomly generate a numerical value, and the numerical value is in between, indicating the probability of mutation. When the probability of mutation is less than the set threshold probability, mutation of the code of the corresponding bit is performed. The core idea is shown in Figure 2.

The mutation operation is to reverse the code value at the selected position. In this article, the binary code 0 and 1 values of the selected position are converted; that is, the direction of the robot path in the original code is changed after the code conversion; for example, when the original code, the two-digit code group 01 means that when the robot moves toward the south, the path code changes to 11, and the robot changes to the south and then turns to the west, and the direction of movement changes. The machine will reverse it to 0 by the mutation operation and if the binary code is 1 in the human path coding, and vice versa.

2.1.5. Termination Conditions. The termination of genetic algorithm is divided into two cases. In the first case, if $t = T$, the population has evolved to the maximum evolutionary algebra, and the individual with the current maximum fitness function value is output as the optimal solution, and the calculation is terminated; in the second case, $t < T$ algorithm exits early, the selected individual has met the optimal standard, and the output is the optimal solution.

Constraints can also be introduced, $f_{(n+1)}$ is the fitness of the best individual in the $n + 1$th generation, and δ is an integer.

$$\left| f_{(n+1)} - f_{(n)} \right| < \delta. \tag{5}$$

When $|f_{(n+1)} - f_{(n)}|$ is less than a set constant, the algorithm is judged to end and the algorithm exits.

2.2. Robot Path Planning Based on Improved Genetic Algorithm

2.2.1. Robot Path Planning Principle Based on Improved Genetic Algorithm. In this paper, the VC++ 6.0 program and the use of nwu-rr-i intelligent mobile robot simulation experiments and comparison of different parameter settings under the different effects of path planning, statistics, and analysis of data verify the correctness of the algorithm.

2.2.2. Fitness Function. The traditional grid size must meet the free movement of the robot in the grid, which stipulates that the minimum distance of the grid must be the length of the robot, which increases the size of the grid and reduces the diversity of the grid area, which is not conducive to smallness. In the experiment, the indication of obstacles and the description of specific environment information are added. This paper fully considers the flexibility and variability of grid division and the reality that the robot is not a particle and proposes a safe distance operator. The safe distance is determined by the volume of the robot. Set the ratio between the simulated map and the actual map to $1 : \alpha$, then the safe distance is set to the length/a of the robot. The obstacle is expanded, and the distance of expansion is set to the actual map scale safe distance. The ratio of the safe distance to the grid size is the number of black grids around the obstacle. If the safety distance is less than 1, it will be expanded by 1 grid count, set the grid within the safe distance of the map to black, which means the dangerous area; other points that are not black are the range of safe route selection. In the robot path planning system, the dangerous area should be directly set as an obstacle in advance to simplify the amount of information processed by the algorithm. The setting method of the safe distance operator fully considers the safety of the path and the convenience of the algorithm.

In the simulation, it is assumed that after the obstacles are marked, the robot can pass through a path in a single grid space. The grid size is divided according to pixels. The size of the path display area has been determined. The proportional relationship is calculated, and the pixel size of the grid height and width is input to determine the number of grids. The length of the character string encoded by the robot path is half of the number of grids, and the number of grids is variable.

The improved robot path planning scheme breaks through the constraint that the grid can only be a grid. Researchers can set the length and width of the grid according to their needs. When the grid is not square, if the path of the robot is a connection between two grid vertices, the connection between the two vertices is not completely drawn through the center of each grid, which increases the difficulty of the display, and in order to compare with the previous chapter the algorithm is unified, and the robot still stipulates that the robot will only move in the four directions of due east, due west, due south, and due north. When comparing algorithms, even if the robot will not advance diagonally, the robot will not have acute or obtuse corners, but since neither the algorithm before or after the improvement can achieve diagonal movement, the comparison evaluation factor convergence speed, optimization time, path smoothness, etc. Not affected, the results still have a certain degree of credibility.

2.2.3. Other Genetic Operators. In genetic algorithm, in order to improve the population quality, the algorithm directly copies the optimal individuals in the current population to the next generation. However, the new population can not meet the requirements of crossover and mutation operation, and the algorithm is easy to fall into premature state. Therefore, it is necessary to rejudge the differences within a certain evolutionary algebra, discard the populations that do not meet the differences in the short term, and create a new population. Multipoint crossover is used for crossover operation, and multipoint mutation is used for mutation operation. The positions of crossover points and mutation points are randomly determined, and multiple experiments are required to obtain empirical values.

2.2.4. Delete Operator. Introduce the idea of mathematical modeling in the deletion operator, encode the path as a binary string in the raster map, and each grid corresponds to an array coordinate. Combine the deletion operator and the array $a[x][y]$ to delete. For codes that enter the same array in the same path, the codes after $a[x][y]$ are directly moved forward to shorten the code length. This method effectively reduces redundant and blocked paths and reduces the path length.

The algorithm refers to the concept of the parent node. The node that the robot walks through is regarded as the parent node and saved independently. When the robot finds that the parent node is repeatedly saved during the path planning process, the path is judged as an invalid path and discarded, using recursive calling. Thinking step by step back to the parent node that was repeatedly saved, and then branch from the parent node to other paths, until a feasible path is found, the invalid branch is deleted, and only one valid path is retained. When programming, the grid and array coordinates are related to each other, which greatly facilitates the participation of the delete operator.

The parent node appears in the definition of the tree and belongs to the knowledge of the data structure. The parent node is a component of the tree. Each node of the parent node has only one antecedent. The root node is the root node. The root node is the ancestor of all nodes in the tree except itself. The root node has no parent node. In this algorithm, the starting point of the robot is the root node, and the robot can only move from the starting point. It can also be seen in the virtual map introduced later that the starting point of the path can only be the starting point of the robot.

2.2.5. Principles of Robot Step Length Setting. Set the step size. The step length represents the number of times the robot moves from the start point to the end point, that is, the superimposed sum of the number of grids that the path of each generation of the robot passes through. Taking into account the specific environmental path planning of the NWU-RR-I intelligent mobile robot, set a parameter for the step length, which is expressed by iT Route Count and limited to 3000 steps. If there are too many steps, the population is judged to be an invalid population, discarded, and restarted. The code is as follows: TRACE ("Search path failed\n").

2.3. Robot Path Planning Based on Improved Genetic Algorithm

(1) Activate the start button

(2) Calculate the safety distance, set the start and end points of the robot, and mark the obstacle area

(3) Initialize the population and judge the difference of the population

(4) Calculate the value of the population fitness function and judge whether it meets the optimal standard. If the population meets the optimal standard, the current solution is output, and the current solution is the optimal solution; if the population does not meet the optimal standard, a series of genetic operations are performed on the population, such as crossover, mutation, deletion, and step size judgment. Return to Step (4).

```
class CASRSystem;
interface IASRMotion;
protected:
BOOL BuildSystem(); //library function initialization
void DestroySystem(;//Library function exit
private:
CASRSystem* m_ pRobot; //Define library system pointer
IASRMotion*
m_ pMotion; //Define library interface pointer
enum {SPEED VALUE = 500;//Set the robot's movement speed
```

Code 1.

(5) Output the optimal path

3. Experimental Research on Robot Path Planning Based on Improved Genetic Algorithm

3.1. Path Planning Simulation of Genetic Algorithm

3.1.1. Simulation Scheme

(1) VC++6.0 Is Selected as the Software Platform. VC++6.0 is a visual integrated development environment IDE (integrated development environment) based on the Windows operating system. It supports C++ language, which is easy to use with databases such as SQL and is easy for man-machine interaction. The realization of the interface has a wide range of uses.

The hardware platform uses NWU-RR-I intelligent mobile robot. The NWU-RR-I intelligent mobile robot system library function NWU-RR-ISetupbuild.EXE is compatible with VC++6.0 and can be programmed to control the robot directly in VC. Click to install NWU-RR-ISetupbuild.EXE. You need to add “NWU-RR-ISystem .h” in front of each header file during programming, which means that VC is compatible with all the functions in the NWU-RR-I robot library. The specific syntax is as follows: #include “NWU-RR-ISystem.h”.

Similarly, the control interface function should be added to the CmotionDemoDIg class, and each interface should be initialized. The code is as follows:

(2) Establishment of Simulation Environment. Divide the environment into a series of grids, and mark obstacles in the grid. Obstacles represent 1, and obstacles are represented as 0.

(3) Genetic Algorithm. According to the algorithm coding rules, the path is coded, the population is initialized, and a series of genetic operations are performed on it, and it is judged whether the population meets the termination condition. If the termination condition is met, the optimal path is output, and the result is displayed; if the termination condition is not met, the genetic operation is continued until the termination condition is met.

The NWU-RR-I intelligent mobile robot moves according to the instructions of the simulation results and completes the path planning task from the start point to the end point.

3.1.2. Path Planning Realization of Genetic Algorithm

(1) *Comparison of the Number of Obstacles in Different Situations.* In the same geographic environment, within the same area, the starting point and ending point do not change, and the number of obstacles in the main path area is set differently

(2) *Implementation of Genetic Algorithm Path Planning for Different Starting Points and End Points.* In the same geographical environment, within the same area, the number of obstacles in the main path area is set the same, and the starting point and ending point are changed

3.2. Simulation of Improved Genetic Algorithm and Comparison of Results. The software platform selects VC++6.0 to make a human-computer interaction interface, and the operation idea of manual selection is added to the traditional robot path planning simulation. The differences are mainly in the following two aspects.

3.2.1. Simulation Scheme. The improved simulation program combined with the actual environment can select obstacles multiple times; select the starting point and end point of the robot multiple times; change the grid size multiple times; reproduce the path and so on. The simulation mode of traditional robot path planning is changed, and the correlation between the simulation environment and the actual environment is increased, which is convenient for simulation experiments in different environments. The adjustability of the number of grids facilitates the comparison of different amounts of information in the same environment. This scheme improves the reliability of robot path planning simulation and has high portability and certain reference value.

3.2.2. Algorithm Improvement. Improvements are mainly proposed from fitness function, deletion operator, termination conditions, etc., which increase the convergence speed of the algorithm and improve the safety of robot path planning.

Table 1: The relationship curve between effective path ratio and population size.

Mutation rate = 0.09	Population size	10	20	30	40	50	60	70	80	100	150	200	300
Crossover probability = 0.8	Before improvement	0	20	40	41	42	43	42	43	42	38	40	39
Evolutionary algebra = 150	After improvement	80	77	79	78	82	83	83.5	79	80	80.5	81.5	81

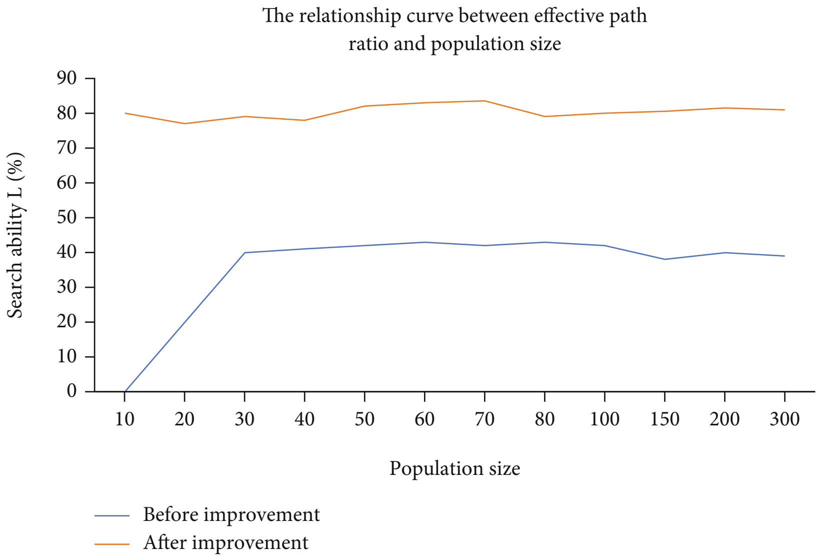

Figure 3: The relationship curve between effective path ratio and population size.

Table 2: The relationship curve between effective path ratio and evolutionary algebra.

Mutation rate = 0.09	Evolutionary algebra	10	20	30	40	50	60	80	100	150	200	500	1000
Crossover probability = 0.8	Before improvement	0	3	4	15	21	23	42	43	62	50	59	70
Population size = 60	After improvement	72	81	80	79	80	81	82	80	79	80	81	80

The population is a binary code generated by a random operator, and its code length changes with the number of grids, so the path of each run is different, but an average can be obtained through various average evolutionary algebras and average path smoothness. The reference value in the sense has a certain universality. In addition, the randomly generated path is not affected by the artificially determined population individuals on the algorithm, and the accuracy of the algorithm can be verified by only multiple measurements and averaging.

4. Experimental Research and Analysis of Robot Path Planning Based on Improved Genetic Algorithm

4.1. Relationship Curve between the Effective Path Ratio and the Population Size before and after the Algorithm Is Improved. Because the path planning algorithm needs to execute a large number of loops, and the C language program is 300 times more efficient than the MATLAB program for loop execution, the algorithm is written in C language, but MATLAB has good graphics processing functions, so MATLAB is used for simulation.

In order to improve computational efficiency, it is necessary to determine the optimal value of genetic parameters. The traditional genetic algorithm equates the search capability L of the mobile robot with the ratio of the effective path at the end of evolution, namely:

$$L = \frac{\text{number of effective paths}}{\text{population size}} (\text{at the end of evolution}). \quad (6)$$

By analyzing Table 1 and Figure 3, the following conclusion can be drawn: In the algorithm of this paper, the genetic parameter that has the greatest impact on the effective path ratio is the mutation probability. 95.64% gradually decreased to 73.42%. It can be seen from the mutation probability graph that when the mutation probability is 0.09, the effective path ratio is 80.65%. From the other three graphs, it can be seen that no matter how the population size, evolutionary algebra, and crossover probability are selected, as long as the mutation probability is taken as 0.09, the effective path ratio is about 80%. The effective path

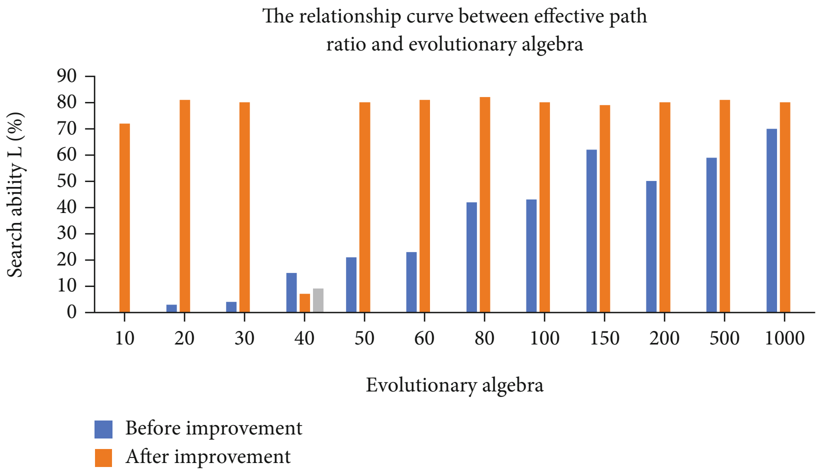

Figure 4: The relationship curve between effective path ratio and evolutionary algebra.

ratio of the improved algorithm has been maintained at a relatively high level, and it is obviously better than the previous algorithm in terms of numerical value and stability, so it proves that the improvement of the genetic algorithm in this paper is successful.

4.2. Relationship Curve between the Effective Path Ratio before and after the Algorithm Improvement and the Evolution Algebra. It can be seen from Table 2 and Figure 4 that the effective path ratio is inversely proportional to the mutation probability but has little to do with population size, evolutionary algebra, and crossover probability. The greater the mutation probability, the more irreparable paths are generated during the mutation process, so the mutation probability is inversely proportional to the effective path ratio, but the mutation probability will increase the diversity of the population, which can avoid the premature solution, so the search is only judged from the effective path ratio. The capacity is inappropriate. Because path planning seeks the shortest path without collision from the starting point to the target point, this paper also proposes a method to measure the search ability; that is, the shortest path length obtained at the end of evolution is used to measure the size of the search ability.

5. Conclusions

With the continuous progress of social science and technology, especially in the field of artificial intelligence and intelligent control, relevant theories continue to be in-depth research, and technological methods are changing with each passing day. With the improvement of social industrial manufacturing automation, robots will play an increasingly important role in industrial production, personal services, and other industries. In this paper, the genetic algorithm is improved by rejudging the difference and discarding precocious population in the process of genetic algorithm iteration. The improved genetic algorithm is used to plan the robot's path. It can be seen from the experimental results that the effective path ratio of the improved algorithm remains at a high level, and it is obviously superior to the previous algorithm in terms of numerical value and stability, which proves that the improvement of the genetic algorithm in this paper is successful. In the future society, robot research will be more in-depth; robot path planning research will make great progress, to show us a higher level of robots.

Data Availability

The data underlying the results presented in the study are available within the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Z. Wang, X. Y. Zheng, C. Lu, and S. P. Zhu, "Structural dynamic probabilistic evaluation using a surrogate model and genetic algorithm," *Proceedings of the Institution of Civil Engineers: Maritime Engineering*, vol. 173, no. 1, pp. 13–27, 2020.

[2] R. Fitriana, P. Moengin, and U. Kusumaningrum, "Improvement route for distribution solutions MDVRP (multi depot vehicle routing problem) using genetic algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 528, no. 1, article 012042, 2019.

[3] G. A. Rovithakis, M. Maniadakis, and M. Zervakis, "A hybrid neural network/genetic algorithm approach to optimizing feature extraction for signal classification," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 695–703, 2004.

[4] 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, no. 1, Article ID 9813040, 2020.

[5] T. Hwu, A. Y. Wang, N. Oros, and J. L. Krichmar, "Adaptive robot path planning using a spiking neuron algorithm with axonal delays," *IEEE Transactions on Cognitive & Developmental Systems*, vol. 10, no. 2, pp. 126–137, 2018.

[6] D. Zhang, X. You, S. Liu, and H. Pan, “Dynamic multi-role adaptive collaborative ant colony optimization for robot path planning,” *Access*, vol. 8, pp. 129958–129974, 2020.

[7] M. Zhao, H. Lu, S. Yang, and F. Guo, “The experience-memory Q-learning algorithm for robot path planning in unknown environment,” *Access*, vol. 8, pp. 47824–47844, 2020.

[8] S. D. Han and J. Yu, “DDM: fast near-optimal multi-robot path planning using diversified-path and optimal sub-problem solution database heuristics,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1350–1357, 2020.

[9] J. Qi, H. Yang, and H. Sun, “MOD-RRT∗: a sampling-based algorithm for robot path planning in dynamic environment,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7244–7251, 2021.

[10] L. Gao, R. Liu, F. Wang et al., “An advanced quantum optimization algorithm for robot path planning,” *Journal of Circuits, Systems and Computers*, vol. 29, no. 8, pp. 2050122–2050143, 2020.

[11] S. Nafea, E. Kadum, and D. Hamza, “Path loss optimization in WIMAX network using genetic algorithm,” *International Journal of Computers, Communications & Control*, vol. 20, no. 1, pp. 24–30, 2020.

[12] E. Joelianto, D. Christian, and A. Samsi, “Swarm control of an unmanned quadrotor model with LQR weighting matrix optimization using genetic algorithm,” *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, vol. 11, no. 1, pp. 1–10, 2020.

[13] S. Quraishi and A. J. Khan, “Genetic algorithm for shortest path optimization in network,” *International Journal of Scientific and Research Publications*, vol. 9, no. 7, article 9188, 2019.

[14] M. R. Islam, P. Protik, S. Das, and P. K. Boni, “Mobile robot path planning with obstacle avoidance using chemical reaction optimization,” *Soft Computing*, vol. 25, no. 8, pp. 6283–6310, 2021.

[15] E. Behmanesh and J. Pannek, “A comparison between memetic algorithm and genetic algorithm for an integrated logistics network with flexible delivery path,” *Operations Research Forum*, vol. 2, no. 3, pp. 1–24, 2021.