Application of Improved Genetic Algorithms in Path Planning

*Di Chen**

China National Accreditation Service for Conformity Assessment, China
chend@cnas.org.cn

Abstract

Genetic algorithm, as a heuristic optimization technique, has achieved remarkable results in various fields, including path planning. Path planning, a fundamental challenge in automation systems and robotics, involves finding the optimal route from a starting point to a destination. This paper focuses on the application of improved genetic algorithms in path planning. Firstly, a brief overview of the fundamental principles of traditional genetic algorithms is provided, including operations such as individual encoding, selection, crossover, and mutation. Subsequently, a thorough exploration is conducted into how improved genetic algorithms can be introduced to address path planning problems. These enhancements encompass optimized population initialization strategies, novel genetic operation methods, and more effective fitness function designs. Furthermore, the discussion extends to how these improvements contribute to accelerating convergence speed, enhancing global search capabilities, and elevating the quality of solution outcomes. A substantial number of objective experiments have demonstrated the effectiveness of our approach.

Keywords: Genetic algorithm, Path planning, Shortest path

1 Introduction

Path planning, as a fundamental challenge in automation systems, robotics, and navigation applications, has consistently garnered extensive research and attention [1]. In numerous real-world applications, such as autonomous driving, robot navigation, and logistics distribution, effective path planning is crucial for enhancing efficiency, reducing costs, and ensuring safety [2-3]. However, due to the complexity of environments, the diversity of problems, and the demand for real-time solutions, finding optimal paths adaptable to various scenarios has remained a challenging task.

With the continuous improvement in computing power and artificial intelligence applications [4-6], optimization algorithms have emerged as potent tools for addressing path planning challenges. Among these, genetic algorithms (GAs) [7-8], based on the principles of biological evolution, have gradually captured the interest of researchers. Genetic algorithms simulate natural selection and genetic mechanisms to seek optimized solutions through an evolutionary process. Their notable attributes such as robust adaptability and robust global search capabilities have unveiled immense potential, particularly in addressing intricate path planning problems.

Specifically, GAs is an optimization technique inspired by natural selection and genetic mechanisms, used to address optimization and search challenges in complex problems. In the field of path planning, genetic algorithms find extensive application in determining the optimal route from a starting point to a destination. The fundamental concept of genetic algorithms involves iteratively refining solutions by simulating the process of biological evolution, aiming to identify improved solutions.

The fundamental workflow of GAs encompasses several key steps: 1) Individual Encoding: Initially, the path planning problem is represented by a set of individuals, often through binary encoding, integer encoding, and similar methods. 2) Initial Population Generation: An initial set of individuals, known as a population, is generated. These individuals are randomly created, forming an initial sample from the solution space. 3) Fitness Evaluation: Each individual's fitness in the problem domain is assessed, reflecting the quality of the path solution. The fitness function is tailored to the problem's characteristics. 4) Selection: Based on fitness, a portion of individuals is selected as "parents." Generally, individuals with higher fitness have a greater chance of selection, simulating the natural selection process. 5) Crossover: Crossover operations occur between selected parent individuals, producing new offspring individuals. This process simulates genetic exchange in biological evolution. 6) Mutation: Offspring individuals are subjected to mutation operations, introducing randomness to enhance population diversity. This step emulates genetic mutation.

However, in path planning, traditional GAs often suffer from inadequate population initialization and fitness evaluation steps, which can lead to their path planning results getting trapped in local optima or even resulting in failure. Therefore, in this paper, we start by addressing various aspects of the traditional genetic algorithm and propose a novel improved genetic algorithm to enhance the ability to find the shortest path, aiming to prevent falling into local optima.

Specifically, we focus on the application of the improved genetic algorithm in the field of path planning. These enhancements may involve optimizing population initialization strategies, introducing novel genetic operation methods, and designing more effective fitness functions. The introduction of these improvements aims to further enhance

*Corresponding Author: Di Chen; E-mail: chend@cnas.org.cn
DOI: https://doi.org/10.70003/160792642024122507013

the performance and efficiency of genetic algorithms in addressing path planning problems. In this paper, we also explore how to model problems considering obstacles, constraints, and multi-objective optimization. Furthermore, we demonstrate how the improved genetic algorithm excels in these intricate scenarios.

The rest of this paper is structured as follows: Section 2 delves into related research. In Section 3, we provide a comprehensive explanation of the proposed virtual try-on framework. Section 4 presents the results of our experiments. Finally, Section 5 analyzes and discusses the limitations of his work. and Section 6 offers concise conclusions.

2 Related Works

In this section, we briefly review previous works on path planning and genetic algorithms.

2.1 Path Planning

In the field of path planning [1, 3, 9-11], numerous studies have been dedicated to developing various methods and techniques to address the challenges of path planning in automation systems, robotics technology [2], and navigation applications. The following are some of the common path planning-related works frequently discussed in papers:

Early path planning methods primarily focused on finding the shortest path in static environments. Classic algorithms such as Dijkstra's algorithm and A* algorithm employ graph search techniques to locate the optimal path from the starting point to the destination. However, these methods may be constrained when dealing with dynamic environments and multi-objective problems.

In practical applications, many studies employ hybrid methods that combine different path planning techniques to overcome various challenges. These methods not only enrich the research in the field of path planning but also provide more powerful tools to address intricate path planning challenges.

To sum up, the field of path planning encompasses various methods and techniques as researchers continuously explore new approaches to meet the path planning demands in different environments and application scenarios. In this paper, our focus will be on the application of improved genetic algorithms in path planning, aiming to enhance the quality and efficiency of solution outcomes.

2.2 Genetic Algorithms

In the field of path planning, Genetic Algorithms (GAs) [12], as an optimization technique based on the principles of biological evolution, have garnered widespread attention and application. GAs simulate natural selection and genetic mechanisms to iteratively explore the solution space of a problem, ultimately identifying optimal path planning solutions.

Early research on genetic algorithms primarily focused on solving classical optimization problems, such as function optimization and parameter tuning. However, with a deeper understanding of genetic algorithms, researchers began applying them to the field of path planning. The introduction of this approach stems from the advantages genetic algorithms offer in dealing with scenarios involving multiple objectives, constraints, and complex environments. Researchers began investigating how to apply genetic algorithms to practical problems like robot navigation, autonomous driving, and UAV path planning.

Although traditional genetic algorithms have achieved some success in path planning, they have also exposed certain issues. For instance, conventional genetic algorithms can get trapped in local optima, resulting in suboptimal path planning outcomes. Therefore, numerous studies have focused on enhancing the performance of genetic algorithms to better suit the characteristics of path planning problems.

In recent years, researchers have proposed various improved variants of genetic algorithms, including elite genetic algorithms, methods combining genetic algorithms with local search, and adaptive strategy-based genetic algorithms. These enhancements aim to boost the algorithms' global search capabilities, convergence speed, and the quality of path solutions. Additionally, some studies have explored the design of more effective fitness functions to accurately assess the merits and drawbacks of path planning solutions. In this paper, we introduce a novel improved genetic algorithm specifically tailored to address path planning problems.

3 Methodology

3.1 Overview

Traditional path-planning algorithms suffer from issues such as high computational complexity and susceptibility to local optima. Therefore, in this chapter, by improving traditional genetic algorithms from the perspective of machine learning, we address problems related to path redundancy and real-time obstacle avoidance. This optimization aims to further enhance the results of shortest-path planning. The execution process is illustrated in Figure 1, and we summarize it into the following five steps:

Step 1: Construct a grid map and use a real-valued permutation encoding method to label the grid cells.

Step 2: Define the start and end points, and initialize the path population using an improved chaotic mapping.

Step 3: Compute fitness values, perform selection, crossover, and mutation operations.

Step 4: Calculate fitness values, determine if the iteration limit has been reached. If so, proceed to Step 5; otherwise, return to Step 3 and continue the genetic operations until the iteration limit is met.

Step 5: Output the optimal path from the start point to the end point.

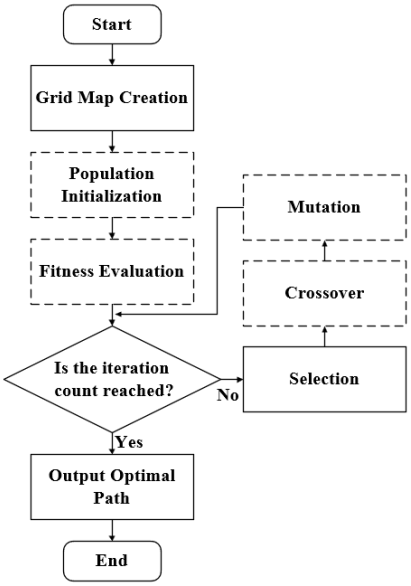


Figure 1. The algorithm flow chart of our improved genetic algorithms (The dashed box represents the part we improved.)

3.2 Grid Map Modeling

The grid map method is a common approach for map modeling, which employs a two-dimensional grid map to represent the environment. In the grid map method, the environment is divided into numerous grid cells, with each cell representing a point within the environment, possessing specific attributes and states. For instance, points can be labeled as passable or impassable, depicted with distinct colors to represent varying terrains, or utilized to depict obstacles and boundaries within the environment.

After the completion of map construction, it is necessary to establish a coordinate system on the map. The purpose of establishing a coordinate system is to allow the Automated Guided Vehicle (AGV) to determine its position and orientation on the map. As shown in Figure 2, we label obstacle grids with the number "1" and accessible grids with the number "0." This can be expressed using Equation (1).

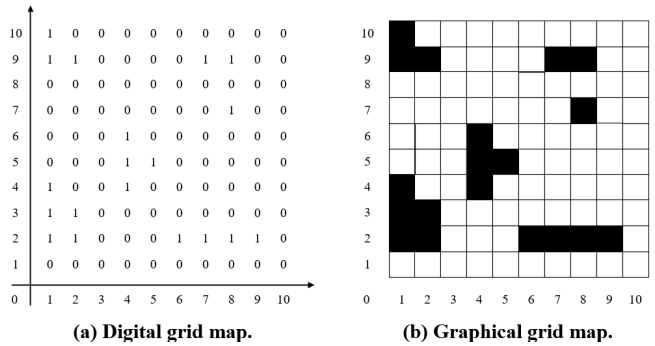


Figure 2. Map modeling example

In order to satisfy the multi-angle visual analysis of data by managers, this tool provides a variety of perspective functions, including heat map, correlation matrix, scatter matrix, histograms, density plots and box-plots. All of these functions exhibit the data from different perspectives and can help the managers to analyze data easily. It is worth noting that before using these functions, the dimension to display the data must be selected as a scenario for analysis. Figure 2 is an illustration of the buttons for data visualization.

$$G = \{0,1\} \tag{1}$$

In our work, a 20×20 grid map is constructed using the Matlab simulation software. The AGV is treated as a moving point, and each grid is assigned a number (represented by n). The starting grid for the AGV is set as grid 0, located at the bottom left corner (0,0), and the destination grid is set as grid 399, situated at the top right corner (20,20). Black blocks in the grid map represent obstacle grids, indicating impassable areas, while white blocks represent path grids, signifying accessible areas. The red line in the grid map indicates the optimal path obtained using a genetic algorithm. In the text, (x, y) represents the position of a certain grid, and the value range for the index n is from 0 to 399. The relationship between them can be expressed using the following formula:

$$x = \mathrm{mod}(n, 20) + 1 \tag{2}$$

$$y = fix(\frac{n}{20}) + 1 \tag{3}$$

where *mod* represents the modulo operation, and *fix* represents the floor division operation.

In this work, all coordinates of the grid map are treated as a complete individual (chromosome), utilizing a real-valued permutation encoding method based on grid indices. This encoding method offers the advantages of a straightforward decoding process and space efficiency, while also being easily comprehensible. Through this approach, a more efficient implementation of the path planning algorithm can be achieved.

3.3 Initialization of Populations

Chaotic mapping [13-15] is commonly utilized in genetic algorithms for generating initial populations, leveraging its chaotic nature to achieve randomness, thoroughness, and regularity. The Sine chaotic map, for instance, is a type of nonlinear dynamic system that can generate high-quality sequences of random numbers. Its primary characteristics encompass strong chaotic behavior, favorable periodicity, and excellent continuity. These attributes collectively promote a more uniform distribution of random number sequences, thereby aiding the algorithm in better exploring the solution space. The traditional sine chaotic mapping is defined as follows:

$$y_{n+1} = \eta \sin(\pi y_n) \tag{4}$$

where η represents a control parameter with a range of 0.87 to 0.93. Although the sine chaotic mapping is straightforward in implementation, as shown in Figure 3, it exhibits chaotic behavior when the sine chaotic mapping falls within the

intervals [0,0.05] and [0.095,1]. In these intervals, the probability of occurrence is significantly higher compared to other intervals, resulting in an uneven distribution and periodicity of the chaotic mapping. This phenomenon significantly impacts the search efficiency and convergence speed of the genetic algorithm. To achieve a more uniform distribution, we use an improved sine chaotic mapping, as demonstrated by the following equation:

$$\begin{aligned} y_{i+1} &= \sin(\delta\pi y_i) \\ \upsilon_{i+1} &= \sin(\delta\pi\upsilon_i) \\ y_{n+1} &= y_{i+1} + \upsilon_{i+1} \bmod 1 \end{aligned} \tag{5}$$

where *mod* represents the modulo operation, υ_i and y_i are random numbers.

In the traditional GAs, random generation of initial populations can lead to the creation of infeasible solutions due to obstacles present in the grid. To address this issue, we introduce the concept of the Vector Field Histogram (VFH) [16-19] obstacle avoidance algorithm to further optimize the method of generating initial populations. The VFH algorithm divides the AGV's operational space into grids and assigns a probability value to each grid to represent the likelihood of obstacles. During the path search process, the AGV avoids grids with high probability values while continuing the search.

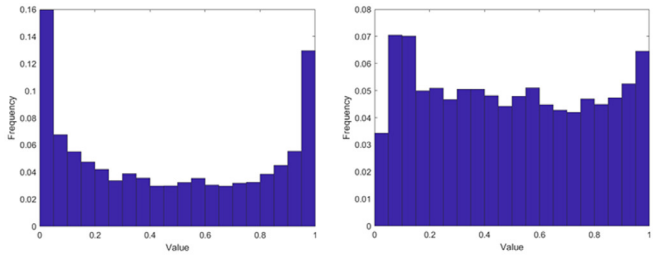


Figure 3. Comparison between the effects of traditional sine chaotic mapping and our improved sine chaotic mapping

3.4 Design of the Fitness Function

In traditional genetic algorithms, the design of the fitness function is crucial to the algorithm's performance. The primary role of the fitness function is to assess the quality of chromosomes (paths) to guide the selection of better chromosomes for crossover and mutation in the next generation. Specifically in the context of path planning problems, fitness can be defined as the quality of the path.

In traditional genetic algorithms, typically only the path length is used as the fitness function, representing the assessment of path quality. The fitness function is formulated as follows:

$$\begin{aligned} f_T(p) &= \frac{1}{f_L(p)} \\ f_L(p) &= \sum_{i=1}^{m-1} \sqrt{(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2} \end{aligned} \tag{6}$$

where p represents an individual path, i represents the coordinates of path point i, and m represents the number of nodes in the path. $f_L(p)$ represents the path length fitness function, and $f_T(p)$ represents the total fitness function as shown in Figure 4. However, due to the uncertainty and numerous turning points in the obstacle-filled warehousing environment, the feasibility index of the path is low, resulting in reduced safety. To minimize excessively small turning angles and enhance path efficiency, this paper introduces a path smoothness evaluation measure $f_S(p)$, which penalizes paths with excessively small angles and aims to reduce the probability of turns. The improved total fitness function $f_T(p)$ is formulated as follows:

$$f_T(p) = \kappa_l \frac{1}{f_L(p)} + \kappa_s \frac{1}{f_S(p)} \tag{7}$$

where κ_l and κ_s are hyperparameters that balance the fitness components.

$$f_S(p) = \sum_{i=1}^{m-2} \begin{cases} S_{i-1} + \alpha, & 91 < \theta \le 179, \\ S_{i-1} + \beta, & 47 < \theta \le 91, \\ S_{i-1} + \gamma, & \theta \le 47. \end{cases} \tag{8}$$

where

$$\theta = \arccos\left(\frac{c+b-a}{2\sqrt{bc}}\right) \tag{9}$$

where a, b, c are Euclidean distances between pairs of grids among the three. The above formula indicates that when the turning angle is within the range of $91 < \theta < 171$, the AGV can execute a safe turn. For turning angles of $47 < \theta \le 91$, the AGV can analyze the surrounding obstacles to perform a turn. When the turning angle $\theta \le 47$, the AGV can turn safely. This can be achieved by adjusting the penalty weights α, β, and γ to constrain the turning angles.

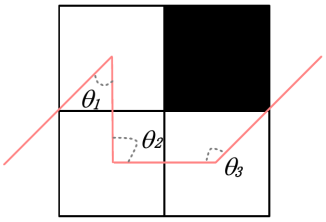


Figure 4. Three different path angles

3.5 Selection

The selection operation [20] is used to determine which individuals will undergo crossover and how many offspring individuals will be generated. During the selection process, the overall population size should remain constant. Fitness values of each chromosome in the population are computed, and the chromosomes are sorted in ascending order based on their fitness. The individual with the lowest fitness value is directly retained in the next generation. However, the lower the fitness value of an individual, the higher the probability

of its selection. Duplicate chromosomes are not preserved.

In this paper, we employ roulette wheel selection. The specific method involves converting each individual's fitness value into a probability, and then selecting individuals based on the magnitude of their probabilities. First, the selection probabilities of each individual are computed, and these probabilities are normalized so that their sum equals 1. Next, a random number within the range of [0,1] is generated and mapped onto the roulette wheel. Individuals for the next generation are selected based on their position on the wheel. Since in roulette wheel selection, individuals with higher fitness values have a higher chance of being chosen, it allows excellent individuals to be more easily inherited in the next generation.

3.6 Crossover

In genetic algorithms, crossover [21] is an operation that generates new individuals. The crossover operation involves pairing the genes of two parent individuals and randomly selecting a segment of genes to exchange, resulting in the creation of two new offspring individuals as shown in Figure 5. In the context of path planning problems, parent individuals can be represented as paths, and the crossover operation involves randomly selecting a segment from two paths for exchange, generating two new paths as offspring.

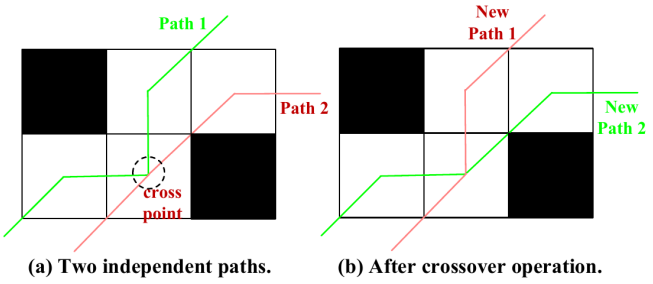


Figure 5. Crossover operation

It should be noted that the magnitude of the crossover probability (p_c) directly affects the algorithm's search capability and optimization performance. A high crossover probability (p_c) can lead to premature convergence of the population, making it prone to getting trapped in local optima and unable to escape, thereby affecting the algorithm's global search capability and optimization performance. Conversely, a very low crossover probability (p_c) can result in a lack of diversity in the algorithm, making it difficult to find better solutions. In order to achieve beneficial evolution, enhance population diversity, and promote exploration, this paper introduces a nonlinear crossover operation, which is formulated as follows:

$$p_c = -\frac{\sqrt{i}}{(1+i)^2} + p_d + \varepsilon \tag{10}$$

Where (p_d) represents the predefined crossover probability threshold and (i) denotes the current iteration index.

3.7 Mutation

Mutation operation [22-23] is a fundamental step in genetic algorithms, aiming to introduce new gene combinations into the population by randomly altering individual genes during the evolutionary process. This process enhances population diversity, preventing the algorithm from prematurely converging to local optima. Mutation typically occurs after crossover operations, with the primary goal of refining the newly generated individuals by modifying certain genes for improved adaptability. In the context of path planning, mutation involves fine-tuning gene sequences to generate novel feasible solutions.

It's important to note that mutation should not occur too frequently, as excessive mutation can disrupt favorable gene combinations, affecting the convergence speed and precision of the algorithm. The mutation rate, denoted as p_m, is typically kept between 0.1% and 1%. A high mutation probability p_m can lead to excessive randomness, resulting in loss of convergence and stability, making it difficult for the algorithm to find suitable solutions. As illustrated in Figure 6, a high linear mutation probability disperses the population widely, making it challenging to converge near the global optimum. Conversely, a low linear mutation probability can trap the algorithm in local optima, preventing it from escaping and conducting broader searches due to reduced population diversity.

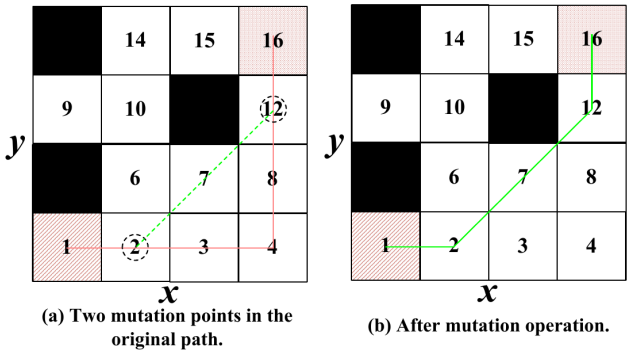


Figure 6. Mutation operation

In this regard, this study introduces a linear mutation operation to enhance the exploratory capability of the genetic algorithm, prevent it from getting stuck in local optima, and improve the diversity of the population during later iterations. This operation introduces a degree of randomness and diversity, thereby expanding the search space and enhancing population diversity. The formula for the linear mutation operation is provided below:

$$p_m = p_{\min}\left(1 - \sqrt{\frac{i}{G}}\right) + \varepsilon \tag{11}$$

where, p_{min} represents the minimum mutation probability, i denotes the current iteration index, G represents the maximum generations, and ε represents the error term (or noise term).

4 Experiments

4.1 Experiment Settings

To validate the effectiveness of the algorithms proposed in the paper, experiments were conducted on the MATLAB simulation platform for both traditional genetic algorithms and improved genetic algorithms.

The parameter settings are as follows: the starting position is labeled as 0, and the ending position is labeled as 399. c

4.2 Visualization of Path Planning Results

Results in 20×20 map. The corresponding analysis of simulation data is presented in Table 1 and Figure 7. Classic genetic algorithms, when compared to the improved genetic algorithms, yield longer path lengths, longer execution times, and slower convergence rates in the pursuit of finding optimal paths. The algorithm tends to gradually converge and stabilize after approximately 5 iterations. On the other hand, the improved genetic algorithm proposed in this paper achieves convergence and stability after around 8 iterations. Furthermore, in traditional genetic algorithms, a notable observation is the existence of multiple 90° turning points and even turning points smaller than 47°. These conditions significantly undermine the safety and feasibility of the path. And our method does not suffer from these problems.

Results in 30×30 map. In order to further demonstrate the superiority of the algorithm presented in this paper and to eliminate the influence of random factors, as shown in Figure 8 and Table 2, simulations were conducted on a more complex random grid map. The size of this map is 30×30, with a more intricate and randomized distribution of obstacles. Consequently, finding the optimal path in this environment poses a greater challenge.

As shown in the figure, the horizontal axis denotes time, the longitudinal axis denotes root categories, and the vertical axis denotes the total check-in times of a user in corresponding time and root category.

Table 1. Simulation ablation experiment of algorithm in 20×20 map

Methods	Path length	Time (s)	Iterations
T GAs	32.5563	0.3774	50
T Gas + A	32.3848	0.3681	50
T Gas + A + B	31.2132	0.2371	50
T Gas + A + B + C	31.2132	0.2366	50
Ours	**29.2132**	0.2240	**50**

Table 2. Simulation ablation experiment of algorithm in 30×30 map

Methods	Path length	Time (s)	Iterations
T GAs	45.7696	0.6869	50
Ours	**41.1838**	**0.5291**	**50**

4.3 Ablation Study

Figure 9 and Table 1 present the results of the corresponding ablation experiments as depicted in the previous figure, including numerical analyses of path length, execution time, and iteration count. Combined with the experimental graphs, it can be observed that when using only the classical genetic algorithm for path planning, the algorithm generates overly long and complex paths with slow convergence. The algorithm gradually converges and stabilizes after approximately 35 iterations. Upon incorporating the improved chaotic mapping A, the algorithm exhibits convergence and stabilization after around 20 iterations. Subsequently, upon integrating the enhanced smoothness fitness function B, the algorithm demonstrates convergence and stabilization after roughly 15 iterations. Furthermore, with the incorporation of the improved nonlinear crossover operation C, the algorithm achieves convergence and stabilization after about 13 iterations. Lastly, upon integrating the enhanced nonlinear mutation operation, the algorithm converges and stabilizes gradually after approximately 10 iterations.

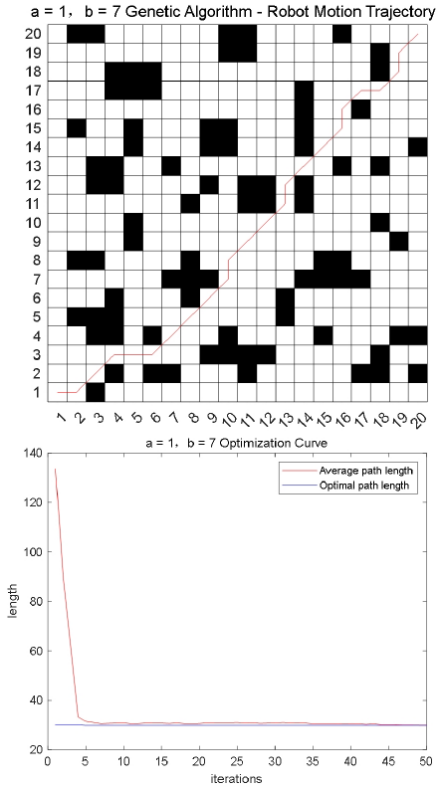
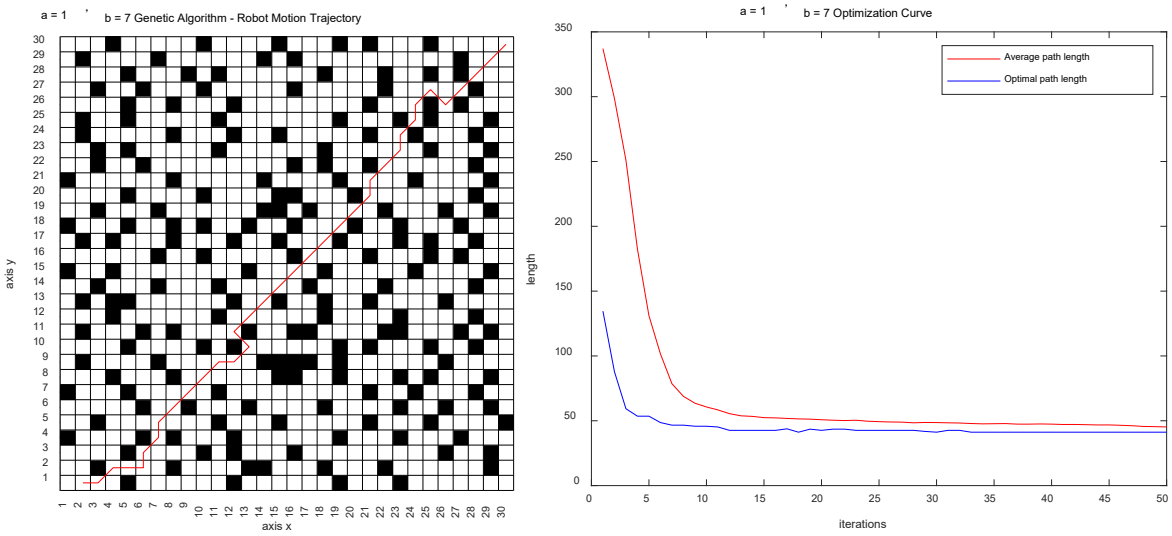


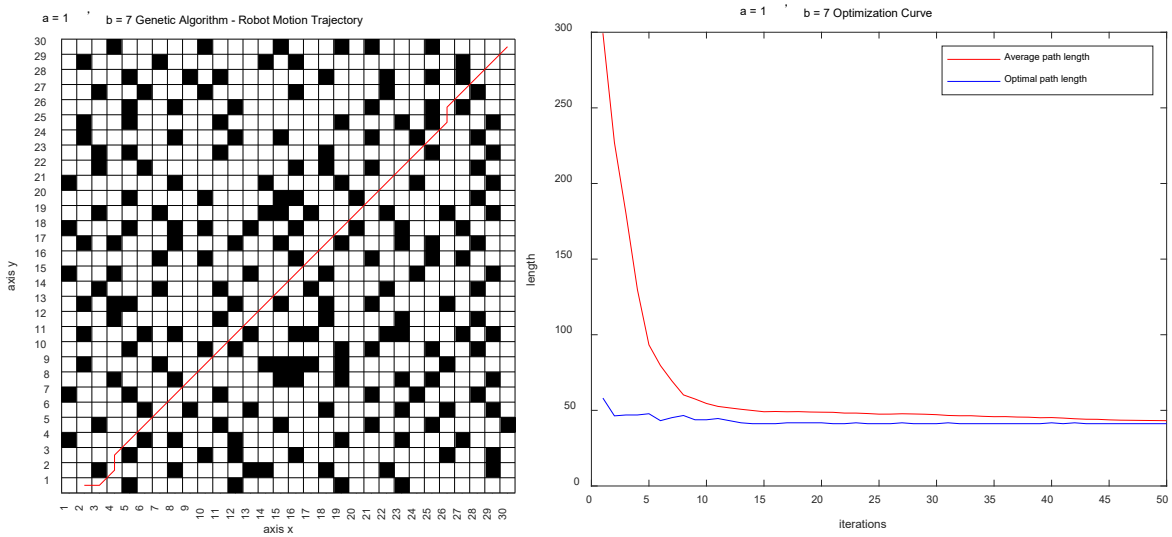
Figure 7. Simulation ablation experiment of algorithm in 20×20 map

5 Discussion

The improved genetic algorithm has achieved significant advancements in the field of path planning. However, it still faces certain limitations while also showcasing several potential prospects. Firstly, despite the notable progress of the improved genetic algorithm in optimizing path planning problems, challenges persist when dealing with complex environments and multiple constraints. Real-world scenarios involve numerous obstacles, dynamic changes, and multi-objective optimizations, potentially causing difficulties in converging to ideal solutions. Therefore, future research could further explore how to enhance the improved genetic algorithm to address the more diverse path planning requirements of real-world situations.

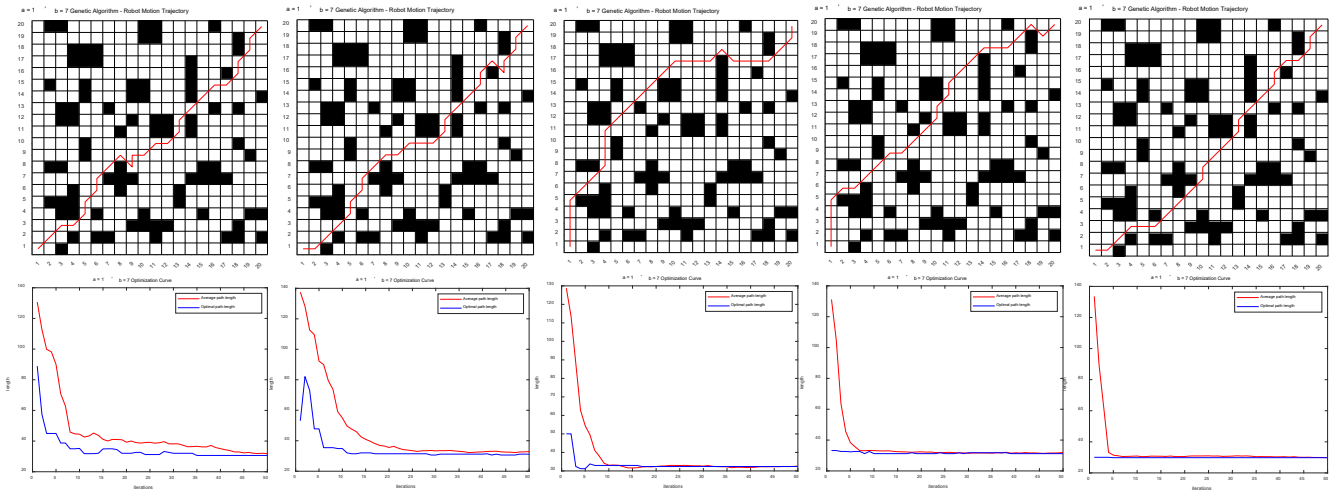


(a) Path planning results based on traditional genetic algorithms.



(b) Path planning results based on improved genetic algorithms.

Figure 8. Comprehensive demonstration of data visualization



(a) Traditional GAs. **(b) GAs + A.** **(b) GAs + A + B.** **(b) GAs + A + B + C.** **Ours**

Figure 9. Ablation study

Secondly, the performance of the improved genetic algorithm heavily relies on parameter configuration and adjustment. Selecting appropriate parameter values is crucial for the algorithm's efficiency and convergence. However, parameter tuning often requires experience and experimentation, potentially affecting the algorithm's stability and reproducibility. Future research might consider adaptive parameter adjustment methods, enabling the algorithm to automatically adapt parameters in various problems and environments, thereby enhancing its adaptability and generalization capability.

Furthermore, path planning problems often involve large search spaces, leading to potential challenges in terms of computational complexity. Although the improved genetic algorithm has optimized the search process to some extent, handling large-scale problems may still pose challenges in terms of computational resources. Future endeavors could explore the integration of other optimization techniques, such as deep learning and reinforcement learning, to further enhance the efficiency and scalability of path planning algorithms.

Lastly, when solving path planning problems, the improved genetic algorithm typically operates on static maps. However, in practical applications, environments can be subject to dynamic changes, such as traffic flow, obstacle positions, and other variables that evolve over time. Therefore, exploring how to integrate the improved genetic algorithm with real-time perception and adaptive path planning to address the dynamic requirements of changing environments is another promising avenue for future research.

6 Conclusion

This study delved into the application of improved genetic algorithms in the domain of path planning, a critical aspect of automation systems and robotics. The investigation began with an elucidation of the foundational principles underpinning traditional genetic algorithms, encompassing individual encoding, selection, crossover, and mutation. Building upon this groundwork, the research advanced towards the integration of enhanced genetic algorithms to tackle the challenges posed by path planning. The innovations introduced in this research primarily revolved around the optimization of population initialization strategies, the introduction of novel genetic operators, and the refinement of fitness function designs. These enhancements collectively aimed at expediting convergence speed, augmenting global search capabilities, and ultimately elevating the quality of solutions generated. The empirical evidence garnered from a comprehensive set of objective experiments strongly attested to the efficacy of the proposed approach. The results from the experiments underscored the tangible benefits of the improved genetic algorithms in addressing complex path planning problems. The optimized population initialization techniques contributed to the rapid initialization of diverse solutions, while the novel genetic operators facilitated efficient exploration of the solution space. Additionally, the carefully designed fitness functions guided the evolutionary process towards more desirable solutions. These synergistic improvements culminated in superior performance in terms of solution quality and convergence speed, as compared to traditional genetic algorithms.

References

[1] A. Gasparetto, P. Boscariol, A. Lanzutti, R. Vidoni, Path planning and trajectory planning algorithms: A general overview, in: G. Carbone, F. Gomez-Bravo (Eds), *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, 2015, pp. 3-27.

[2] J. Zhang, Q. Xiao, L. Li, Solution space exploration of low-thrust minimum-time trajectory optimization by combining two homotopies, *Automatica*, Vol. 148, Article No. 110798, February, 2023.

[3] L. Li, C. Colombo, L. Gkolias, J. Zhang, Low-thrust station-keeping towards exploiting the inclined geosynchronous dynamics, *Advances in Space Research*, Vol. 72, No. 5, pp. 1570-1582, September, 2023.

[4] F. Yu, Z. Chen, M. Jiang, Z. Tian, T. Peng, X. Hu, Smart Clothing System With Multiple Sensors Based on Digital Twin Technology, *IEEE Internet of Things Journal*, Vol. 10, No. 7, pp. 6377-6387, April, 2023.

[5] C. Du, F. Yu, M. Jiang, A. Hua, X. Wei, T. Peng, X. Hu, Vton-scfa: A virtual try-on network based on the semantic constraints and flow alignment, *IEEE Transactions on Multimedia*, Vol. 25, pp. 777-791, February, 2022.

[6] F. Yu, A. Hua, C. Du, M. Jiang, X. Wei, T. Peng, L. Xu, X. Hu, VTON-MP: Multi-Pose Virtual Try-On via Appearance Flow and Feature Filtering, *IEEE Transactions on Consumer Electronics*, Vol. 69, No. 4, pp. 1101-1113, November, 2023.

[7] R. Sharma, S. N. Singh, Towards Accurate Heart Disease Prediction System: An Enhanced Machine Learning Approach, *International Journal of Performability Engineering*, Vol. 18, No. 2, pp. 136-148, February, 2022.

[8] J. Wang, Z. Zhai, Y. Lan, X. Zhai, L. Zhao, Reliability Analysis and Optimization of Forage Crushers Based on Bayesian Network, *International Journal of Performability Engineering*, Vol. 19, No. 10, pp. 700-709, October, 2023.

[9] S. A. Bortoff, Path planning for UAVs, *Proceedings of the 2000 american control conference*, Chicago, IL, USA, 2000, pp. 364-368.

[10] O. Souissi, R. Benatitallah, D. Duvivier, A. Artiba, N. Belanger, P. Feyzeau, Path planning: A 2013 survey, *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*, Agdal, Morocco, 2013, pp. 1-8.

[11] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, Y. Xia, Survey of robot 3D path planning algorithms, *Journal of Control Science and Engineering*, Vol. 2016, Article No. 7426913, July, 2016.

[12] K. De Jong, Learning with genetic algorithms: An overview, *Machine learning*, Vol. 3, No. 2-3, pp. 121-

138, October, 1988.
[13] E. V. Altay, B. Alatas, Bird swarm algorithms with chaotic mapping, *Artificial Intelligence Review*, Vol. 53, No. 2, pp. 1373-1414, February, 2020.
[14] X. Wu, Z. Guan, A novel digital watermark algorithm based on chaotic maps, *Physics Letters A*, Vol. 365, No. 5-6, pp. 403-406, June, 2007.
[15] C. Li, B. Feng, S. Li, J. Kurths, G. Chen, Dynamic analysis of digital chaotic maps via state-mapping networks, *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 66, No. 6, pp. 2322-2335, June, 2019.
[16] J. Borenstein, Y. Koren, The vector field histogram-fast obstacle avoidance for mobile robots, *IEEE transactions on robotics and automation*, Vol. 7, No. 3, pp. 278-288, June, 1991.
[17] B. Kazem, A. Hamad, M. Mozael, Modified vector field histogram with a neural network learning model for mobile robot path planning and obstacle avoidance, *International Journal of Advancements in Computing Technology*, Vol. 2, No. 5, pp. 166-173, December, 2010.
[18] W. Yim, J. Park, Analysis of mobile robot navigation using vector field histogram according to the number of sectors, the robot speed and the width of the path, *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, Gyeonggi-do, Korea (South), 2014, pp. 1037-1040.
[19] J. Kumar, R. Kaleeswari, Implementation of vector field histogram based obstacle avoidance wheeled robot, *2016 Online international conference on green engineering and technologies (IC-GET)*, Coimbatore, India, 2016, pp. 1-6.
[20] J. Zhong, X. Hu, J. Zhang, M. Guo, Comparison of performance between different selection strategies on simple genetic algorithms, *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06)*, Vienna, 2005, pp. 1115-1121.
[21] G. Syswerda, Uniform crossover in genetic algorithms, *Proceedings of the 3rd International Conference on Genetic Algorithms (ICGA)*, San Francisco, CA, United States, 1989, pp. 2-9.
[22] H.-Y. Fan, J. Lampinen, A trigonometric mutation operation to differential evolution, *Journal of global optimization*, Vol. 27, No. 1, pp. 105-129, September, 2003.
[23] M. Albayrak, 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, March, 2011.

Biography

Di Chen received the M.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2007. He works at China National Accreditation Service for Conformity Assessment now. His current research interests include high voltage electrical testing technology, quality control and measurement uncertainty evaluation.