



Article

Omnidirectional AGV Path Planning Based on Improved Genetic Algorithm

Qinyu Niu, Yao Fu * and Xinwei Dong

College of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an 710054, China;
21205224070@stu.xust.edu.cn (Q.N.); 21205016005@stu.xust.edu.cn (X.D.)

* Correspondence: fuyao@stu.xust.edu.cn; Tel.: +86-18729091197

Abstract: To address the issues with traditional genetic algorithm (GA) path planning, which often results in redundant path nodes and local optima, we propose an Improved Genetic Algorithm that incorporates an ant colony algorithm (ACO). Firstly, a new population initialization method is proposed. This method adopts a higher-quality random point generation strategy to generate random points centrally near the start and end of connecting lines. It combines the improved ACO algorithm to connect these random points quickly, thus greatly improving the quality of the initial population. Secondly, path smoothness constraints are proposed in the adaptive function. These constraints reduce the large-angle turns and non-essential turns, improving the smoothness of the generated path. The algorithm integrates the roulette and tournament methods in the selection stage to enhance the searching ability and prevent premature convergence. Additionally, the crossover stage introduces the edit distance and a two-layer crossover operation based on it to avoid ineffective crossover and improve convergence speed. In the mutation stage, we propose a new mutation method and introduce a three-stage mutation operation based on the idea of simulated annealing. This makes the mutation operation more effective and efficient. The three-stage mutation operation ensures that the mutated paths also have high weights, increases the diversity of the population, and avoids local optimality. Additionally, we added a deletion operation to eliminate redundant nodes in the paths and shorten them. The simulation software and experimental platform of ROS (Robot Operating System) demonstrate that the improved algorithm has better path search quality and faster convergence speed. This effectively prevents the algorithm from maturing prematurely and proves its effectiveness in solving the path planning problem of AGV (automated guided vehicle).



Citation: Niu, Q.; Fu, Y.; Dong, X. Omnidirectional AGV Path Planning Based on Improved Genetic Algorithm. *World Electr. Veh. J.* **2024**, *15*, 166. <https://doi.org/10.3390/wevj15040166>

Academic Editor: Joeri Van Mierlo

Received: 15 March 2024

Revised: 31 March 2024

Accepted: 11 April 2024

Published: 16 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In scenarios such as intelligent manufacturing factories and logistics and transportation systems, automated guided vehicles (AGVs) have been widely used as the main tools for intelligent material transportation [1]. AGV path planning technology refers to AGVs that autonomously generate safe and feasible paths based on evaluation criteria such as travel time, path length, and turning frequency, combined with their own sensors' perception of the environment [2,3].

The use of traditional genetic algorithms in path planning typically involves identifying the most efficient route from a starting point to a destination. The fundamental concept of traditional genetic algorithm path planning is to search for the optimal path by simulating the process of genetic evolution in nature through population evolution and fitness evaluation. The general steps and characteristics of traditional genetic algorithm path planning are as follows: Firstly, the path planning problem must be transformed into an optimization problem that can be handled by a genetic algorithm. This requires defining the objective function of the problem, constraints, and the solution space of the problem. Secondly, the population must be initialized. A set of initial paths, or chromosomes, is randomly generated based on the problem's characteristics.

The fitness value of each path is then calculated to measure its superiority, typically based on factors such as path length and obstacle avoidance ability. Finally, a selection operation is performed. The path's fitness value determines the probability of selecting it as the parent for generating the next generation of the population. The crossover operation generates new paths by crossing over chromosomes to increase population diversity and explore the search space. The mutation operation is also performed as follows: Mutate the chromosomes of the paths with a certain probability to introduce new path information and increase the diversity of the population, preventing it from falling into local optimal solutions. Repeat the selection, crossover, and mutation operations until the termination conditions are met. The traditional genetic algorithm is advantageous for coping with complex path-planning problems and has better global search capability and robustness. The language used should be clear, objective, and value-neutral, avoiding biased, emotional, figurative, or ornamental language. The text should adhere to style guides, use consistent citation, and follow a consistent footnote style and formatting features. Additionally, the language should be formal, avoiding contractions, colloquial words, informal expressions, and unnecessary jargon. Finally, the text should be free from grammatical errors, spelling mistakes, and punctuation errors. However, it also has drawbacks, such as low search efficiency and a tendency to fall into local optimal solutions. It is important to note that subjective evaluations should be excluded unless clearly marked as such. Therefore, when faced with different path-planning problems, it is necessary to select the appropriate optimization algorithm or enhance it by combining it with other heuristic algorithms based on the specific situation.

As of now, a considerable number of researchers worldwide have conducted extensive studies in this area. For example, Tian et al. [4] proposed an approach based on improved adaptive genetic algorithms to address the problem of infeasible paths generated by genetic algorithms by ensuring feasible paths for population individuals after genetic operations using prior knowledge. Li et al. [5], recognizing that the performance of basic genetic algorithms depends on factors such as the initial population quality, introduced an artificial bee colony algorithm to initialize the population and improve the population diversity. They then designed an adaptive strategy adjustment for crossover and mutation operators based on trigonometric functions to improve the convergence speed of the algorithm. Qiao et al. [6] corrected non-smooth paths generated by genetic algorithm planning using an enhanced artificial potential field method, optimizing paths by adding nodes for smoother trajectories. Liu et al. [7] addressed the problems of traditional mutation operators struggling to produce high-quality solutions and premature algorithm convergence by presenting a path fine-tuning algorithm and incorporating simulated annealing operations in each generation for local optimization. Xu et al. [8] proposed a combination of immune operators to improve solution quality. Andranik S. Akopov et al. [9] introduces a novel approach using parallel genetic algorithms for simulation-based optimization, addressing large-scale optimization problems characterized by wide feasible ranges and multiple local extrema. This methodology is successfully applied to minimize the potential number of traffic accidents in an Artificial Multi-Connected Road Network (AMCRN). R Zhou et al. [10] proposed a genetic algorithm-based entry event and car following event framework to solve the problem of a lack of scenarios and low efficiency in virtual testing of CAV (Connected automated vehicles), which significantly improves accident scenarios and challenging scenarios and is more efficient. CHAYMAA et al. [11] proposed a mobile robot path planning method based on improved crossover genetic algorithm in terms of efficiency and applicability of scene generation, which was improved in terms of distance and safety to find a feasible and efficient path between two points. Chen et al. [12] proposed a genetic algorithm fused with the fireworks algorithm to compensate for the premature convergence disadvantage of genetic algorithms. Li et al. [13] used the A-star algorithm as heuristic information in population initialization and added deletion sub-operators, etc., which reduced the number of iterations, but the computation time was long when faced with complex paths.

The study aims to address the limitations of traditional genetic algorithms (GA) in path planning. The proposed Improved Genetic Algorithm (IGA) incorporates the ant colony algorithm (ACO) to solve issues such as poor initialized path quality, premature convergence,

long turn times and angles, easy deadlock, regional extremes, and too many redundant path nodes. First, a new population initialization method is proposed to improve the quality of random point generation by generating points near the start and end point connecting lines. This results in higher-quality random points and shorter generated paths. Additionally, an improved ant colony algorithm is used to efficiently connect these random points, resulting in a higher-quality population under this initialization strategy. During the selection phase, a combination of tournament and roulette wheel selection is utilized to prevent premature convergence of the population. Additionally, an elite retention strategy is employed to prevent the degradation of the algorithm. In the crossover phase, the algorithm introduces the concept of edit distance and proposes a two-layer crossover operation based on edit distance to prevent invalid crossover and expedite the evolution of the algorithm. During the mutation stage, a new method that incorporates the concept of simulated annealing is proposed to introduce novel mutations. Additionally, a three-stage mutation method is suggested to enhance population diversity and accelerate convergence to the optimal solution. To eliminate redundant nodes in the paths and expedite the algorithm's convergence, a deletion operator is also included to remove non-essential paths. The fitness function introduces the constraint of path smoothness to guide the algorithm towards smoother paths. To balance population diversity and the algorithm's ability to determine optimal individuals in later stages, we use adaptive crossover and mutation probability adjustment strategies. This ensures greater species diversity in the early stages and the faster determination of optimal individuals in later stages. The simulation results indicate that the improved algorithm is more effective than the traditional genetic algorithm. It can avoid local extremes better, converge faster, and generate higher quality and shorter length paths. Practical experiments confirmed the advantages of the improved algorithm in terms of algorithm convergence speed, path smoothness, and path length.

2. Establishment of Environmental Model

In this paper, the grid method is used to model the working environment of an automated guided vehicle. Obstacles are represented by black grid cells, free space is represented by white grid cells, and the grid width is unit length. As shown in Figure 1 below, the start point is denoted by S, and the end point is denoted by E. The following assumptions are made in the modeling process: The actual size of the AGV is not considered as a point mass [14]. A Cartesian coordinate system is established with the lower left corner of the grid map as the origin. The grids are numbered sequentially from 0, bottom to top and left to right. The formula for converting the grid number n into real coordinates (X_n, Y_n) in an $m \times m$ grid environment is as follows:

$$\begin{cases} X_n = \text{mod}(n, m) + 1 \\ Y_n = \text{fix}\left(\frac{n}{m}\right) + 1 \end{cases} \quad (1)$$

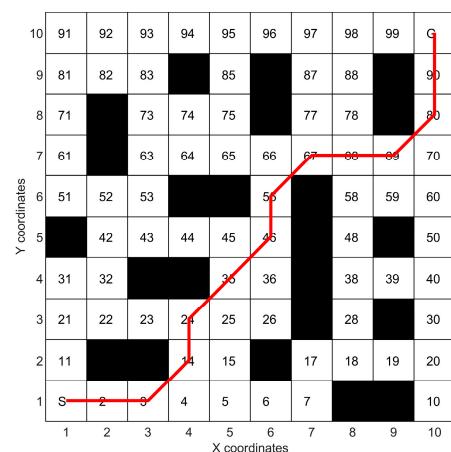


Figure 1. Genetic algorithm robot motion trajectory.

In Equation (1), m is the number of grids in each row or column, mod is the remainder operation, and fix is the rounding operation.

3. Genetic Algorithm Improvement and Implementation

3.1. Encoding Method

This article uses grid serial numbers to encode chromosomes; that is, each path of the AGV is composed of the grid serial numbers it passes through in the process of moving from S to G. The red path in Figure 1 can be expressed as {S, 10, 21, 31, 42, 43, 44, 55, 66, 77, 78, 89, G}.

3.2. Population Initialization

In the existing literature, population initialization methods usually use random generation [15]. This approach tends to generate a large number of infeasible paths in the initialized population, which affects the rate of population evolution and the quality of individual chromosomes [15]. On this basis, this paper proposes a heuristic strategy based on the reference [16], which integrates an improved ACO algorithm for generating initial populations, as shown in Figure 2. The specific steps are summarized below:

- (1) Determine the population size M by connecting the points S and G with a straight line, using S as the starting point and G as the destination point.
- (2) Divide the line SG vertically by the dotted line L_i into $n - 2$ parts.
- (3) The point P_i is randomly generated on the dashed line $L_i\{i = 1, 2, \dots, n - 2\}$ on the dashed line $L_i\{i = 1, 2, \dots, n - 2\}$, the point P_i is randomly generated, and the coordinates of P_i are calculated as shown in Equation (2). If the generated P_i is not in free space, a point in the neighborhood of P_i in free space is regenerated to replace it.

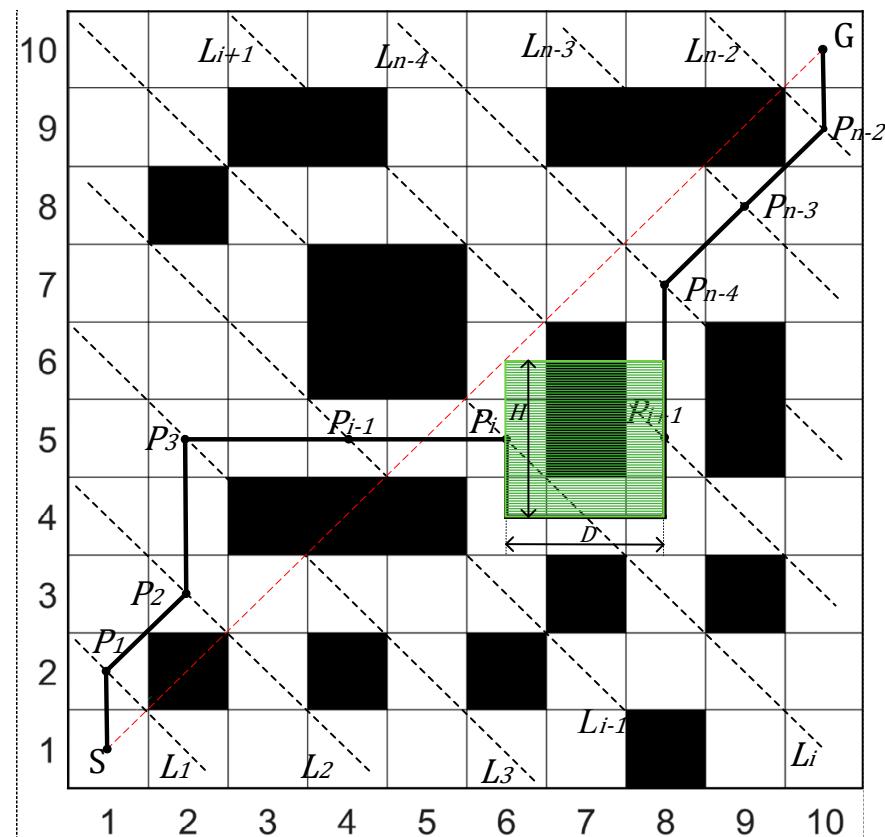


Figure 2. Population initialization.

$$\begin{aligned} X_i &= X_1 + \frac{i-1}{n-1}(X_n - X_1) + \frac{m_{rand}d_{grid}}{\sqrt{(X_1-X_n)^2+(Y_1-Y_n)^2}}(Y_n - Y_1) \\ Y_i &= Y_1 + \frac{i-1}{n-1}(Y_n - Y_1) + \frac{m_{rand}d_{grid}}{\sqrt{(X_1-X_n)^2+(Y_1-Y_n)^2}}(X_n - X_1) \end{aligned} \quad (2)$$

In Equation (2), n represents the length of the chromosome, d_{grid} is the grid size, (x_i, y_i) denotes the coordinates of the randomly generated point P_i , (x_1, y_1) and (x_n, y_n) are the starting point S and the target point G of the AGV movement, respectively, and m_{rand} is a random number selected from $\{-m, 1 - m, \dots, m - 1, m\}$.

- (4) Connect the randomly generated points into a continuous path. Starting from point S, if there is no obstacle between neighboring nodes, connect them with a straight line and go to step six. If there are obstacles, go to the next step.
- (5) Since the ACO algorithm is characterized by fast search speed and optimization on simple paths, this paper uses the ACO algorithm between two points with obstacles [17]. The maximum value of horizontal distance D and vertical distance H is taken as a square to determine the area to be planned S. Assuming that the starting point is A and the ending point is B, the centroid coordinates of the area S are $(\frac{X_A+X_B}{2}, \frac{Y_A+Y_B}{2})$. If the path generation fails because the area is too small and there is no path, the area of area S is increased by one unit in all directions, and then replanning is performed. In order to increase the initialization speed of the algorithm, the area of region S can only be increased twice. After two times, the area is set to fail to generate a feasible path, and random points A and B are regenerated for path planning. Consider P_i and P_{i+1} at the two ends of the obstacle as the start and end points, respectively. There are two key factors for the ants to move from the current position i to the next position j: probabilistic selection and pheromone updating. To further improve the convergence of the ACO algorithm and the quality of the initial population, the weighting factor $r(r \in (0, 1))$ in the state transition probability formula is

$$P_{ij}^k = \begin{cases} r \frac{\tau_{ij}^\gamma \times \eta_{ij}^\varepsilon}{\sum\limits_{k \in allowed_k} \tau_{ij}^\gamma \times \eta_{ij}^\varepsilon}, & j \in allowed_k \\ 0, & otherwise \end{cases} \quad (3)$$

$$\eta_{ij} = \frac{1}{\left\{ (x_i - x_j)^2 + (y_i - y_j)^2 \right\}^{\frac{1}{2}}} \quad (4)$$

where τ_{ij} denotes the pheromone concentration, η_{ij} denotes the pheromone inspiration factor, γ and ε are τ_{ij} and η_{ij} the weighting coefficients, respectively, and $allowed_k$ are the nodes that the ants can pass through when moving.

In terms of pheromone updating, this paper takes the path length as the criterion for better convergence. If the path is smaller than the average path, it is rewarded, and vice versa. The specific equations are as follows:

$$\tau_{ij}(t+1) = (1 - \zeta)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (5)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{3\zeta}{L_k} + \varphi \frac{L_k - L_{best}}{L_{ave} - L_{best}}, & L_k < L_{best} \\ \frac{2\zeta}{L_k} + \varphi \frac{L_{best} - L_k}{L_{ave} - L_{best}}, & L_{best} < L_k < L_{ave} \\ \frac{\zeta}{L_k}, & else \end{cases} \quad (6)$$

where ζ denotes the pheromone volatilization system; $\sum_{k=1}^m \Delta\tau_{ij}^k(t)$ denotes the pheromone left between paths (i, j) in this iteration; $\Delta\tau_{ij}^k(t)$ denotes the pheromone's left by the ants on the path (i, j) at the moment t; L_k denotes the path length of the ants; ζ denotes the pheromone strength, which is generally a constant; L_{best} is the optimal path length; L_{ave} is the average length of the paths in this iteration; φ and ζ is the weighting factor.

- (6) Determine whether goal point G has been reached. If it has, go to the next step; if it has not, return to step 4.
- (7) If there are duplicate paths in the generated individuals, delete them.
- (8) If the population size equals M, stop; otherwise, repeat steps 3 to 7.

3.3. Adaptation Function

In genetic algorithms, the fitness function is used to evaluate the quality of chromosomes under predetermined conditions to determine the evolutionary direction of the population. Considering the stability and economy of AGVs in practical applications, the evaluation criteria for generating paths are path length and path smoothness. Therefore, the objective function of the feasible path can be expressed as follows:

$$F_{fit}(W^*) = \min \{F_{fit1}(W), F_{fit2}(W)\} \quad (7)$$

In Equation (7), W^* denotes the planned path and $F_{fit1}(W)$ and $F_{fit2}(W)$ are given by Equations (8) and (9), respectively.

- (1) Path length:

$$F_{fit1}(W) = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (8)$$

In Equation (5), n is the number of nodes that generate the path, and x_i and y_i are the abscissa and ordinate of the i th node, respectively.

- (2) Path smoothness:

$$F_{fit2}(W) = \sum_{i=2}^{n-1} fit2 \quad (9)$$

$$fit2 = \begin{cases} \frac{\arccos \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} + 1}{10}, & -1 \leq \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \leq -0.7 \\ \frac{\arccos \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} + 10}{10}, & -0.7 < \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \leq 0 \\ \frac{\arccos \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} + 100}{10}, & 0 < \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \leq 1 \end{cases} \quad (10)$$

In Equation (10), vector $\vec{a} = (x_i - x_{i-1}, y_i - y_{i-1})$, vector $\vec{b} = (x_i - x_{i+1}, y_i - y_{i+1})$.

From Equation (7), it can be seen that the evolutionary direction of the paths planned by the IGA is towards shorter paths and shorter number of turns and turning angles. Therefore, the fitness function is shown in Equation (11), and the fitness function value of the optimal path is the smallest.

$$F_{fit}(W) = \alpha \cdot F_{fit1}(W) + \beta \cdot F_{fit2}(W) \quad (11)$$

In Equation (11), α and β are the weights of each part.

3.4. Improved Genetic Operations

3.4.1. Improve Selection Operations

Traditional genetic algorithms usually use the roulette wheel selection method in the selection phase, which may lead to premature convergence and local optimization of the algorithm [18]. To address this phenomenon, this paper combines the improved roulette

wheel selection method with the tournament selection method. To prevent population degradation, this paper also employs an elite preservation strategy.

The enhanced roulette wheel selection method follows a specific procedure. Individuals are sorted according to their fitness values from high to low and proportionally categorized into "high fitness individuals", "medium fitness individuals", and "low fitness individuals". If the roulette wheel selects individuals with "high fitness individuals" or "medium fitness individuals", they are selected accordingly. In contrast to the common method in the literature, where "low fitness individuals" are directly discarded during selection and the corresponding "high fitness individuals" are selected, a 2:1 ratio is used in this paper, i.e., if "low fitness individuals" are selected, they are reselected from the "high fitness individuals" and "medium fitness individuals". This ensures species diversity and effectively avoids local optimization. In addition, tournament selection is introduced, where a number of individuals are selected directly using a tournament selection method. This further preserves species diversity while the algorithm converges. Finally, an elite preservation strategy was introduced to prevent species degradation.

3.4.2. Double Crossover Based on Edit Distance

As populations evolve, identical chromosomes become more common, and crossover operations on identical chromosomes are ineffective and slow down the evolution of populations. Therefore, a screening process based on edit distance is applied before performing crossover operations [19]. If the edit distance between two chromosomes is too small, the crossover operation is not performed.

After the screening process, the chromosomes are sorted in ascending order based on their fitness values for single-point crossover. Specifically, two parent chromosomes are randomly selected from the population, and common genes (excluding S-points and G-points) are identified for crossover. If multiple common genes are found, any one of them is selected for crossover.

If no common genes can be found, the decision is made based on the sorting results. If the fitness value of the chromosome is in the top 60%, no crossover is performed to avoid destroying the best chromosomes and to prevent population degradation. If the fitness value of the chromosome is in the bottom 40%, a gene is randomly selected from each of the two parent chromosomes and a new "binding chromosome" is initialized between these two genes. The bonded chromosome is then used as an intermediary for the crossover operation. This enhanced crossover operation can result in pathway duplication. If a bonding chromosome is present during the crossover process, the duplicated pathway will be removed after the crossover operation. Figure 3 below shows the flowchart of the crossover operation based on edit distance.

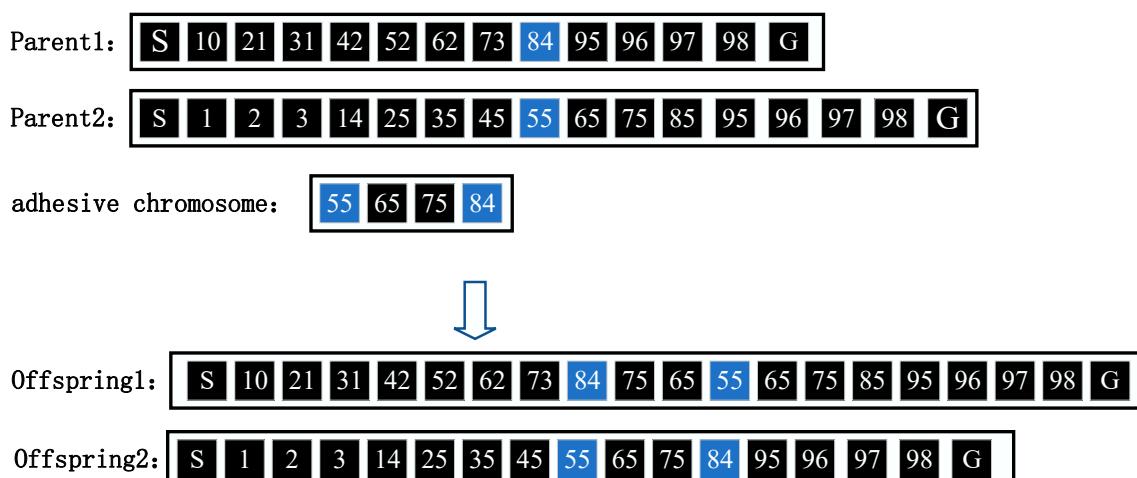


Figure 3. Flowchart of cross-operations based on edit distance.

3.4.3. Simulated Annealing Three-Stage Mutation

The main purpose of the mutation operation is to improve the local searching ability of the algorithm so that the algorithm can rid the local optimum, ensure global exploration, maintain the diversity of the population, and prevent premature convergence [20]. The traditional single-point random mutation operation often suffers from the problems of path blocking and weak local searching ability. In view of this, this paper proposes a three-step mutation strategy by combining the concept of simulated annealing on the basis of the literature [21]. Figure 4 below illustrates the improved mutation operation. Figure 5 illustrates the Improved Mutation Operation flowchart. The specific steps are as follows.

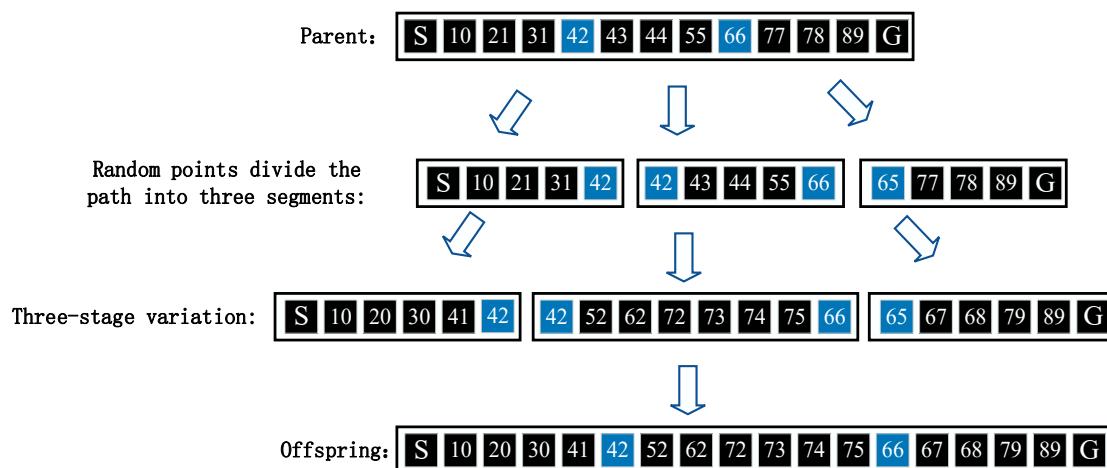


Figure 4. Improved variant operation.

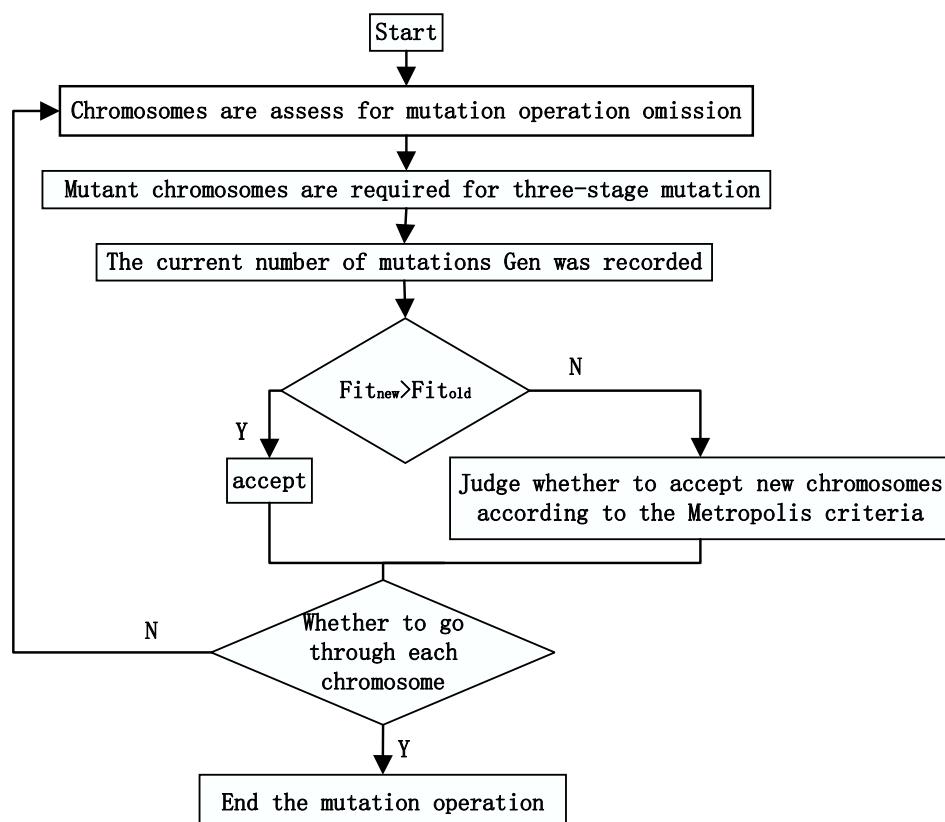


Figure 5. Improved variant operation flowchart.

- (1) Use P_m to determine whether the current chromosome should be mutated. If $P_m >$ random number r, then mutation occurs.
- (2) Randomly select two nodes P_a and P_b (excluding S and G points) from the parent individual W_F to be mutated.
- (3) These two random points divide the parent individual W_F into three segments. Reinitializing these three segments results in a mutated chromosome.

After a trisection mutation, the newly acquired chromosome must be evaluated for acceptance based on the Metropolis guidelines. If the fitness value of the new chromosome is lower than that of the old chromosome, the new chromosome is accepted. Otherwise, the acceptance of the new chromosome depends on the Metropolis policy defined below.

$$P = \begin{cases} 1, & \text{Fit}_{\text{new}} > \text{Fit}_{\text{old}} \\ \exp(-\frac{\text{Fit}_{\text{new}} - \text{Fit}_{\text{old}}}{T}), & \text{Fit}_{\text{new}} \leq \text{Fit}_{\text{old}} \end{cases} \quad (12)$$

In Equation (12), Fit_{new} represents the fitness value of the new chromosome, Fit_{old} represents the fitness value of the old chromosome, and T denotes the current temperature. The calculation formula for T is given by

$$T = T_0 \times Q^{\text{Gen}} \quad (13)$$

In Equation (13), T_0 represents the initial temperature, Q is the cooling rate, and Gen stands for the number of mutation occurrences.

3.4.4. Deletion Operator

After the mutation process, the paths generated may still require several iterations to achieve smoothing [22]. Deletion operations are necessary to refine the genetic algorithm's chromosome [23], removing redundant nodes, speeding up population convergence, and improving algorithm efficiency. In the following example, the red dashed line represents the replanned path after the deletion operation.

- (1) First, determine if there is an obstacle between the starting point S and P_{11} using the method described in [23]. If there is no obstacle (see Figure 6), connect the redundant nodes and delete them.
- (2) Next, evaluate the connectivity between S and P_{17} . As shown in the figure, there is an obstacle between these two points, so P_{17} is retained.
- (3) The process is repeated using P_{17} as a new reference point and applying the rules from the first two steps, until the evaluation reaches the final node.

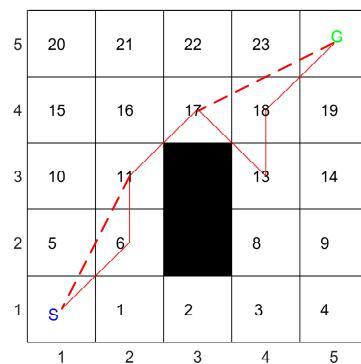


Figure 6. Delete the road map before and after the operation is processed.

3.5. Adaptive Adjustment of Crossover and Mutation

The genetic algorithm's convergence and optimal solution quality are affected by the crossover probability (P) and mutation probability (P_m). A small P can make it difficult to generate new individuals and result in a slow search process, while a large P can disrupt

the genetic model. A small P_m can impede population renewal and hinder the generation of new individuals, while a large P_m can lead to a lack of direction in the search process and the loss of good individuals [23]. An optimal balance between P_m and P is crucial for the success of the algorithm. To address this issue, an adaptive adjustment strategy is adopted in this paper [24]. This strategy increases P and P_m in the early stages to enhance the algorithm's searching ability and decreases them in the later stages to determine the optimal individuals. The formula used for this strategy is as follows:

$$P_c = \begin{cases} \frac{P_{c1}+P_{c2}}{2} + \frac{P_{c1}-P_{c2}}{2} \sin\left(\frac{f_1-f_{avg}}{f_{max}-f_{avg}} \cdot \frac{\pi}{2}\right), & f_1 \geq f_{avg} \\ P_c, & f_1 < f_{avg} \end{cases} \quad (14)$$

$$P_m = \begin{cases} \frac{P_{m1}+P_{m2}}{2} + \frac{P_{m1}-P_{m2}}{2} \sin\left(\frac{f_2-f_{avg}}{f_{max}-f_{avg}} \cdot \frac{\pi}{2}\right), & f_2 \geq f_{avg} \\ P_m, & f_2 < f_{avg} \end{cases} \quad (15)$$

In Equations (14) and (15), P_{c1} and P_{c2} represent the maximum and minimum values of the crossover probability, respectively, while P_{m1} and P_{m2} represent the maximum and minimum values of the mutation probability, respectively. f_{max} denotes the maximum fitness value in the population, f_{avg} represents the average fitness value in the population, f_1 is the fitness value of the individual with higher fitness during crossover and f_2 is the fitness value of the individual involved in the mutation operation.

3.6. Improved Flowchart

Figure 7 below shows a flowchart of the improved genetic algorithm of this paper.

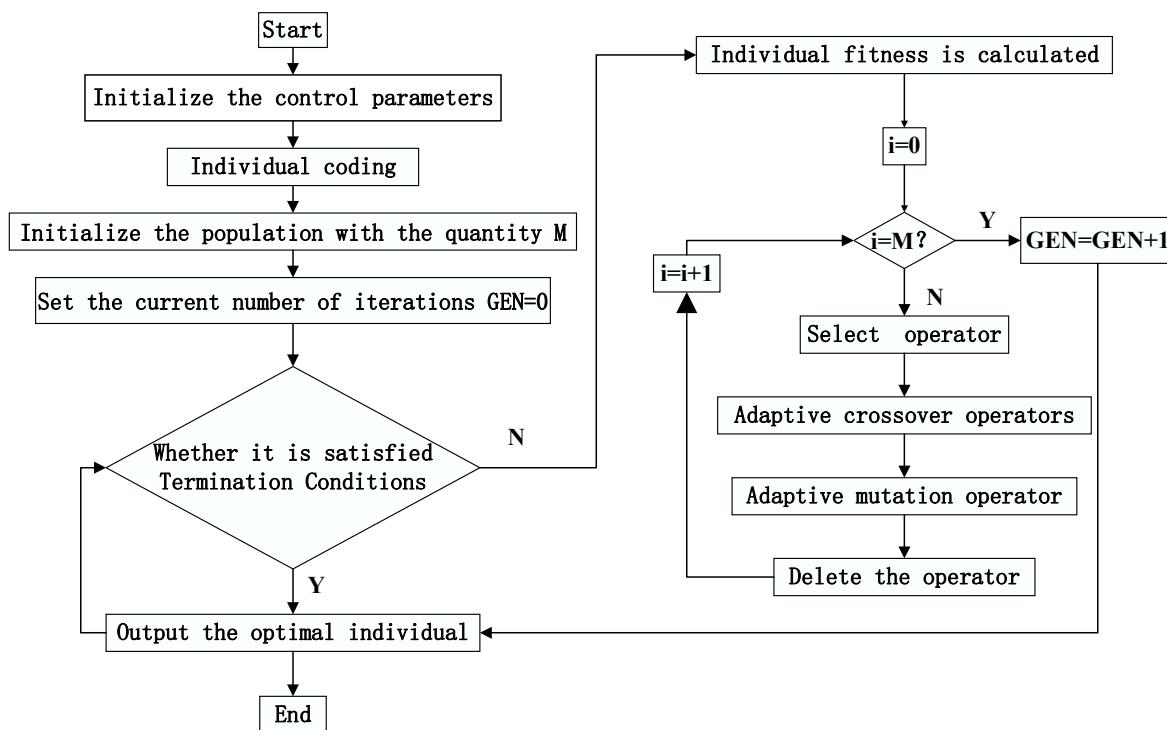


Figure 7. Flowchart for improving the algorithm.

4. Simulation and Results Analysis

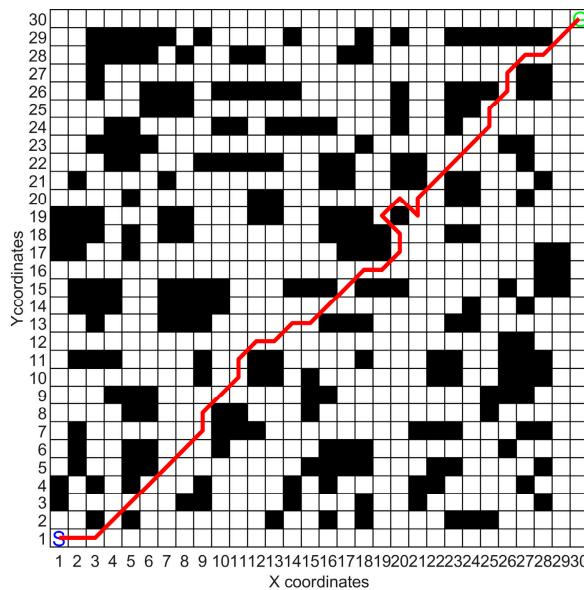
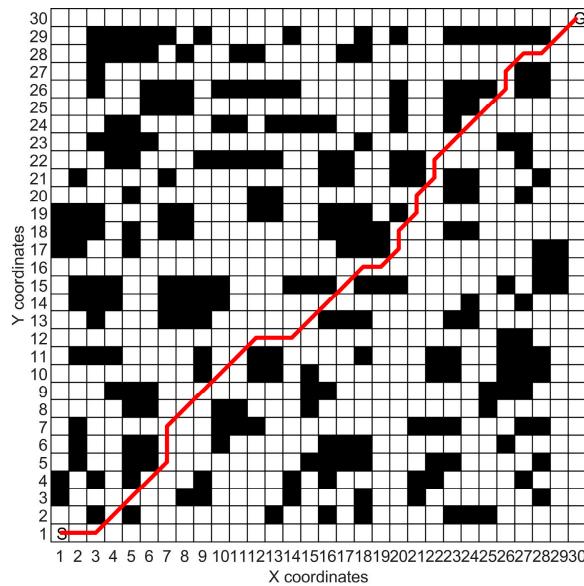
4.1. Simulation Experiment

MATLAB 2020b was used to conduct path-planning simulation experiments. A comparison and analysis of the traditional genetic algorithm [25] and the Improved Genetic Algorithm proposed in this paper was performed on different scaled grid maps. Table 1 shows the specific parameters.

Table 1. Simulation parameters.

Parameter Name	Parameter Value	Parameter Name	Parameter Value
Number of populations	200	Grid size	1 m × 1 m
Maximum crossover probability	0.9	Minimum crossover probability	0.6
Maximum probability of variation	0.2	Minimum probability of variation	0.005
Maximum number of iterations	100	Cooling rate	0.9
Starting temperature	300	Termination temperature	0

The starting and ending points are labeled as S and G, respectively. Here is a comparison of the three path planning results on a 30×30 grid map (see Figures 8–21):

**Figure 8.** GA algorithm planning path under 30×30 map.**Figure 9.** Ref. [2] algorithm planning path under 30×30 map.

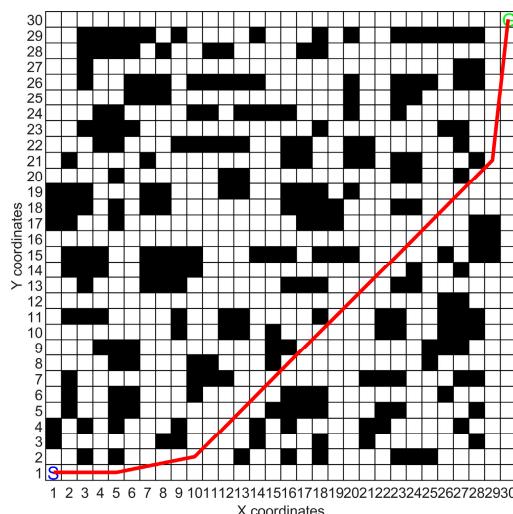


Figure 10. IGA algorithm path planning under 30×30 map.

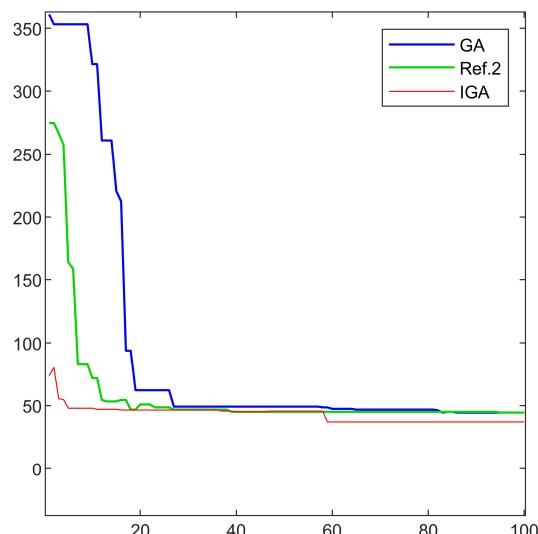


Figure 11. Comparison of path length iterations under 30×30 map.

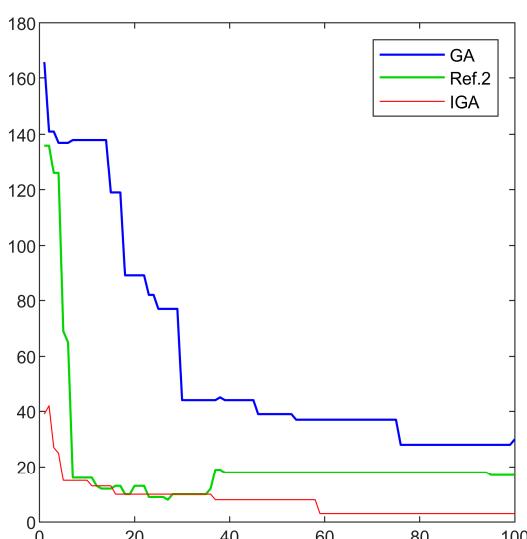


Figure 12. Iterative comparison chart of the number of turns under 30×30 map.

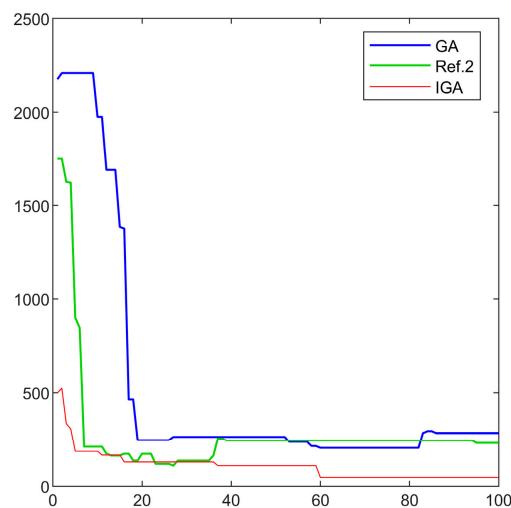


Figure 13. Smoothness iteration comparison chart under 30×30 map.

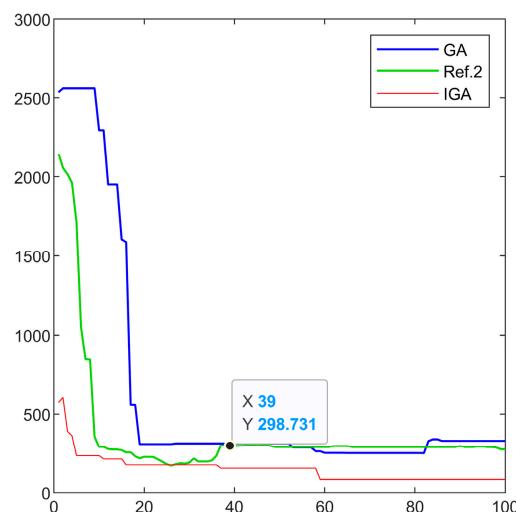


Figure 14. Fitness iteration comparison chart under 30×30 map.

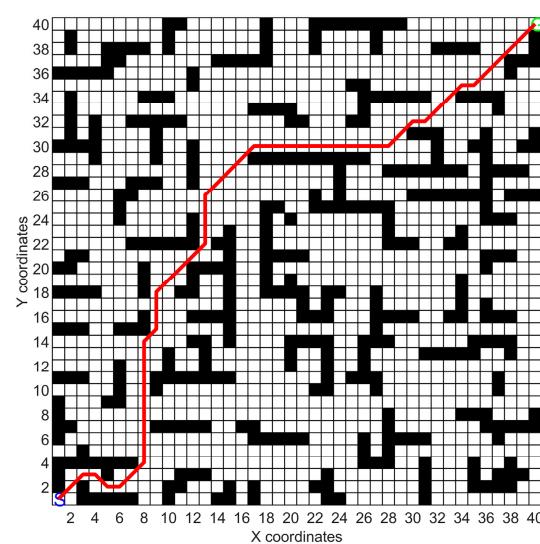


Figure 15. The GA algorithm plans the path under 40×40 map.

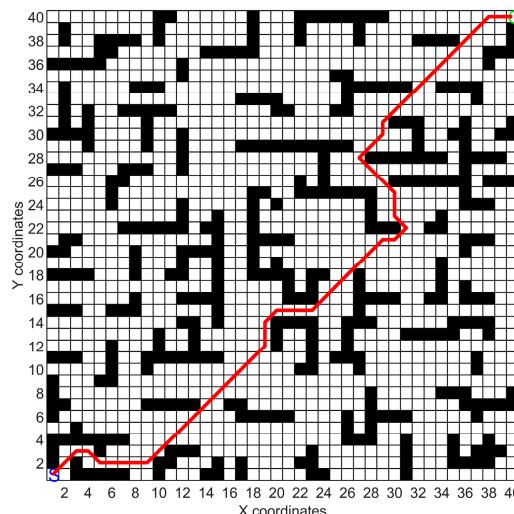


Figure 16. Ref. [2] algorithm planning path under 40×40 map.

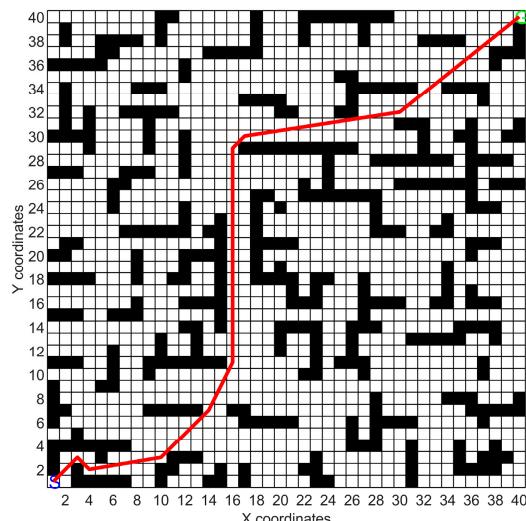


Figure 17. IGA algorithm path planning under 40×40 map.

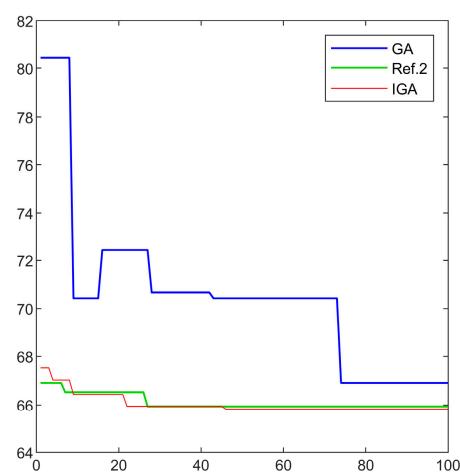


Figure 18. Path length iteration comparison chart under 40×40 map.

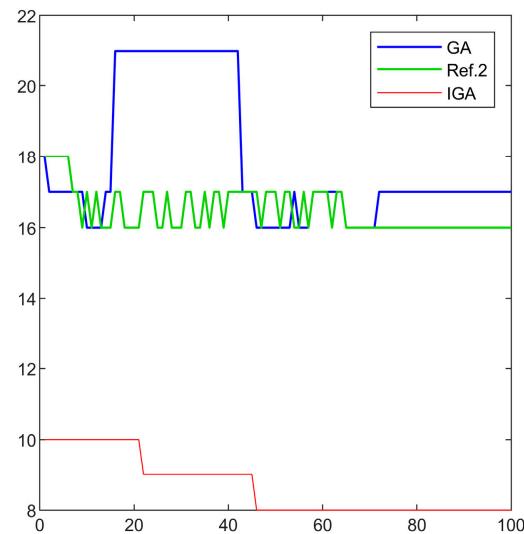


Figure 19. Iteration comparison chart of the number of turns under 40×40 map.

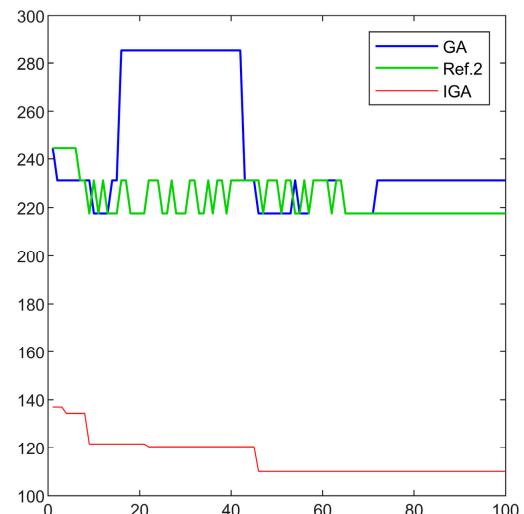


Figure 20. Smoothness iteration comparison chart under 40×40 map.

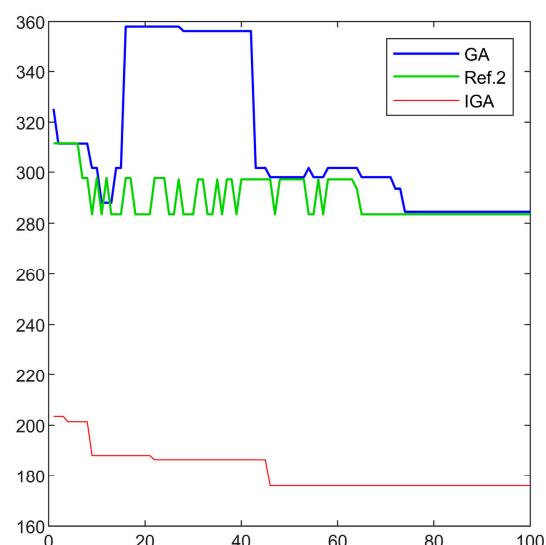


Figure 21. Fitness iteration comparison chart under 40×40 map.

As shown in Figures 8–10 and Figures 15–17, the quality of the paths generated by the three algorithms improves sequentially under different scaled maps. However, the traditional genetic algorithm generates a large number of redundant nodes, unnecessary paths and too many corners after the same iteration due to its inherent limitations. The improved algorithm in reference [2] reduces the path length, but still contains excessive corners and unnecessary paths due to the fact that it does not consider the smoothness of the paths in the fitness function. In this paper, an improved algorithm is proposed to minimize the path corners, length and smoothness by increasing the initial population size, improving the adaptive function and other operations. After the same number of iterations, the IGA has the highest quality, the shortest path and the least number of corners among the three algorithms.

Comparing Figures 11 and 18, it can be observed that the IGA exhibits the fastest convergence rate during the path iterations, despite the presence of more obstacles, which results in fewer feasible paths. This is due to the new initialization population operation as well as the improved crossover and mutation operations proposed in this paper. In addition, due to the use of more advanced crossover and mutation operations, IGA shows the best search capability and produces the shortest paths compared to the three different algorithms.

The three algorithms compared in Figures 12, 13, 19 and 20 can be hit first, IGA has the best search capability and the shortest path. Therefore, IGA has the best quality. This is due to the introduction of the concept of path smoothing in this paper, which greatly reduces the number of turns at the beginning of the iteration.

Figures 14 and 21 show that among the three algorithms, IGA has the best path fitness value due to the improvements made at different stages. IGA also has advantages in terms of number of iterations, path length and convergence speed compared to the algorithm in reference [2].

Here are the results of comparing the path planning algorithms on a 40×40 grid map: Table 2 below shows the simulated data for different map sizes.

Table 2. Statistics of simulation results in maps of different scales.

Map Size	The Type of Algorithm	Path Length/m		Path Angle/°		Number of Turns	
		Average Value	Optimum	Average Value	Optimum	Average Value	Optimum
10 × 10	GA	15.2853	14.4852	392	360	8.2	8
	[2]	14.9237	14.4852	381	360	8.2	8
	IGA	14.6345	14.2624	213	127	4.2	3
20 × 20	GA	30.1064	29.7990	629	585	13.4	13
	[2]	29.4463	28.8995	594	495	12.8	11
	IGA	28.0954	27.8147	316	207	10	8
30 × 30	GA	46.1955	45.1127	1047	945	22.4	21
	[2]	44.5269	44.5269	802	765	17	17
	IGA	44.6581	44.4910	289	84	7.2	3
40 × 40	GA	68.1886	66.9117	783	720	16.9	16
	[2]	65.9460	65.9411	728	720	16.19	16
	IGA	65.8953	65.8278	469	349	9.7	8

The simulation results and data comparison show that the traditional genetic algorithm produces poor-quality initial population results. This leads to uneven paths and too many redundant nodes, which do not effectively reduce the path length. Although the improved algorithm in reference [2] reduces the path length, it still has unnecessary turns and

redundant paths. This paper proposes the Improved Genetic Algorithm (IGA), which enhances path smoothness and path length, prevents premature convergence, improves population initialization quality, and increases convergence speed.

4.2. ROS Platform Experiment

To test the improved algorithm's practical feasibility, we conducted experiments on an omnidirectional AGV platform with Mecanum wheel drives, as shown in Figure 21.

The platform's main components are as follows:

The industrial computer has the ROS (Robot Operating System) system installed.

The motion control uses microcontrollers, specifically the STM32F103ZET6 model with three USART interfaces, a CAN bus interface, and a 485 bus interface. Additionally, it integrates the JY901B IMU sensor.

The motors used are MOTEC's DSEM-G series DC servo geared motors rated at 24 V with a 30.25:1 reduction ratio and integrated high-resolution encoders. These motors are used with MOTEC-CONRA series DC servo drives and support RS232 and CAN communication modes.

The power supply comes from a 100 Ah Li-FePO4 battery with an output voltage of 48 V and an operating voltage range of 40 V to 58 V. The IGAs are designed for practical evaluation purposes.

The experimental platform provides a comprehensive verification environment for evaluating the practical applicability of the IGA.

Figure 22 shows the experimental scenario, including the cost maps displayed by RVIZ. Black areas represent obstacles, while gray areas represent inflated areas. The navigation task configuration and results are presented below.



Figure 22. Omnidirectional AGV experimental platform.

Figure 23 illustrates the experimental scenario. The AGV is passing through an obstacle in the figure.



Figure 23. Experimental scene.

Figures 24–27 show that in a real-world working environment, the GA algorithm contains a large number of redundant nodes and right-angle turns, which increases the probability of collisions. In contrast, the IGA is more efficient than the previous algorithm because it has a shorter planning path and avoids redundant and nonessential turns. Table 3 indicates that the improved algorithm reduces the path length by approximately 28%, turns by about 76%, and improves the convergence speed by about 59%. Table 4 shows that the improved algorithm reduces the path length by approximately 20%, turns by approximately 84%, and improves convergence speed by approximately 58%. After conducting several experiments, the combined data indicate that the improved algorithm reduces the path length by approximately 27%, turns by approximately 63%, and improves convergence speed by approximately 58%. These results demonstrate the feasibility and stability of the improved algorithm proposed in this study.

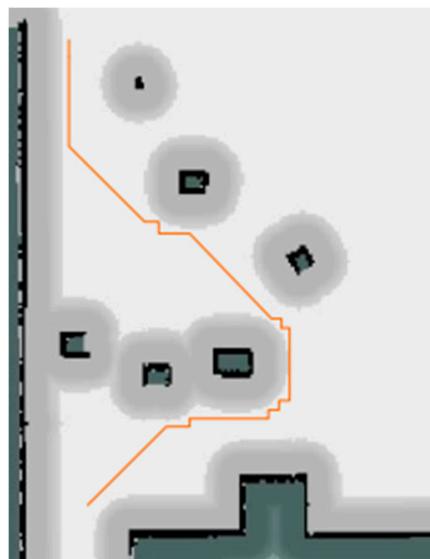


Figure 24. GA algorithm planning path1.

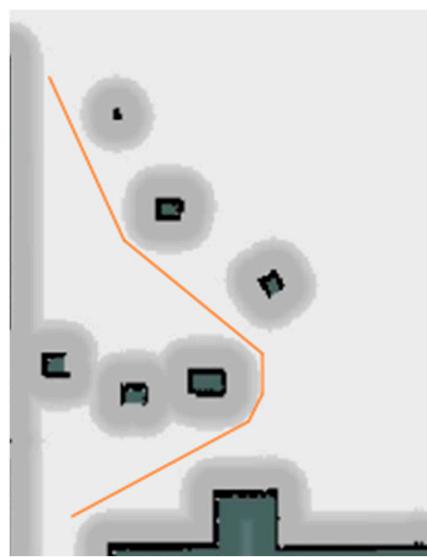


Figure 25. IGA algorithm planning path2.

The following is a comparison of the planned paths by the two algorithms after resetting the start and goal points following adjustments to the obstacle positions in the experimental scene:

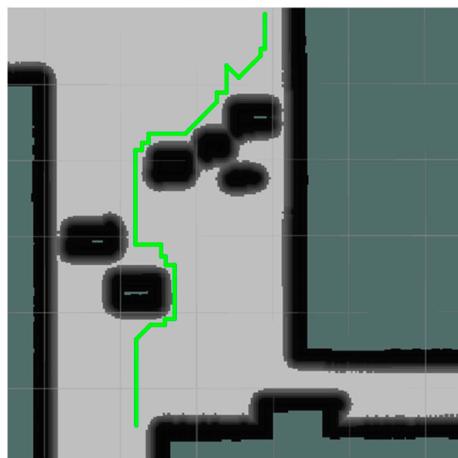


Figure 26. GA algorithm planning path3.



Figure 27. IGA algorithm planning path4.

Table 3. Comparison of Path 1 and Path 2 results.

Algorithm Type	Path Length/m	Turn Count	Convergence Count
GA	9.746254	17	41
IGA	6.985546	4	17

Table 4. Comparison of Path 3 and Path 4 results.

Algorithm Type	Path Length/m	Turn Count	Convergence Count
GA	12.44134	25	38
IGA	9.99693	4	16

5. Conclusions

This paper proposes an improvement to the traditional genetic algorithm to address issues of premature convergence, excessive redundant nodes in generated paths, and slow convergence speed in path planning. A new initialization method, which combines the improved ant colony algorithm, is proposed to enhance the quality of the initial population during the population initialization process. During the selection stage, the algorithm's searching ability is enhanced and premature convergence is prevented by fusing the roulette wheel selection method and tournament selection method. In the crossover phase, a two-layer crossover operation based on edit distance is designed to prevent ineffective crossover and improve the iteration speed of the algorithm. During the mutation stage, a new three-stage mutation operation inspired by the simulated annealing idea is proposed. This enables the algorithm to effectively escape local optima. Additionally, dynamic mutation and crossover strategies enhance the algorithm's searching ability in the early stage and improve the determination of optimal individuals in the later stage. The software simulation results indicate that the Improved Genetic Algorithm (IGA) presented in this paper outperforms both the GA and the methods described in the literature [2] in terms of path length, path smoothness, turn freedom, and algorithm convergence speed. Practical experiments on the ROS experimental platform have shown that the path generated by the IGA algorithm can effectively control the algorithm to avoid falling into a local optimum and improve the convergence speed compared to the traditional GA algorithm while ensuring the shortest path.

Author Contributions: Conceptualization, Q.N. and Y.F.; investigation, Y.F.; methodology, Q.N. and Y.F.; project administration, Y.F.; resources, Y.F.; software, Y.F. and X.D.; supervision, Q.N.; validation, Q.N.; writing—original draft, Y.F and X.D.; writing—review and editing, Y.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Chen, Y.; Lin, Q.; Deng, Z.; Chen, G. Research Progress on AGV Path Planning Algorithms. *J. Mech. Electr. Technol.* **2022**, *2022*, 39–43. [[CrossRef](#)]
- Wang, L.; Li, M. Application of Improved Adaptive Genetic Algorithm in Mobile Robot Path Planning. *J. Nanjing Univ. Sci. Technol.* **2017**, *41*, 627–633. [[CrossRef](#)]
- Zhu, D.; Yan, M. Overview of Mobile Robot Path Planning Technology. *Control Decis.* **2010**, *25*, 961–967. [[CrossRef](#)]
- Tian, X.; Liu, G.; Zhou, W.; Guo, K. Research on Robot Path Planning Based on Improved Adaptive Genetic Algorithm. *Mach. Tool Hydraul.* **2016**, *44*, 24–28+62.
- Li, Y.; Wan, Y.; Zhang, Y.; Kuang, H. Warehouse Robot Path Planning Based on Artificial Bee Colony—Adaptive Genetic Algorithm. *J. Instrum. Meas.* **2022**, *43*, 282–290. [[CrossRef](#)]

6. Qiao, S.; Wu, Y.; Zhang, J.; Shi, G. Path Planning Based on Genetic Algorithm and Artificial Potential Field. *Mod. Electron. Tech.* **2012**, *35*, 75–78. [[CrossRef](#)]
7. Liu, E.; Yao, X.; Lan, H.; Jin, H. Dynamic Path Planning and Implementation of Automatic Guided Vehicle Based on Improved Genetic Algorithm. *Comput. Integr. Manuf. Syst.* **2018**, *24*, 1455–1467. [[CrossRef](#)]
8. Xu, M.; Wang, J.; Liu, B.; Ma, L.; Chai, L.; Xiang, L.; Zhou, J. Robot Path Planning Based on Improved Genetic Algorithm. *J. Shihezi Univ. (Nat. Sci.)* **2021**, *39*, 391–396. [[CrossRef](#)]
9. Akopov, A.S.; Beklaryan, L.A.; Beklaryan, A.L. Simulation-Based Optimisation for Autonomous Transportation Systems Using a Parallel Real-Coded Genetic Algorithm with Scalable Nonuniform Mutation. *Cybern. Inf. Technol.* **2021**, *21*, 127–144. [[CrossRef](#)]
10. Zhou, R.; Liu, Y.; Zhang, K.; Yang, O. Genetic Algorithm-Based Challenging Scenarios Generation for Autonomous Vehicle Testing. *IEEE J. Radio Freq. Identif.* **2022**, *6*, 928–933. [[CrossRef](#)]
11. Chaymaa, L.; Said, B.; Ali, E. Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning. *Procedia Comput. Sci.* **2019**, *127*, 180–189.
12. Chen, X.; Li, Y.; Xiao, S. Optimization of Fresh Agricultural Products Logistics Distribution Path Based on Improved Genetic Algorithm. *J. Dalian Jiaotong Univ.* **2022**, *43*, 97–102+111. [[CrossRef](#)]
13. Li, Y.; Dong, D.; Guo, X. Mobile robot path planning based on improved genetic algorithm with A-star heuristic method. In Proceedings of the 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 11–13 December 2020; Volume 9.
14. Zhao, J. Research on Ant Colony Optimization Algorithm for Mobile Robot Path Planning. Doctoral Dissertation, Northeastern University, Boston, MA, USA, 2012.
15. Wang, J.; Wang, X.; Tian, Q.; Sun, A.; Zhang, X.; Yuan, L. Robot Path Planning Based on Improved Fuzzy Adaptive Genetic Algorithm. *Mach. Tool Hydraul.* **2021**, *49*, 18–23.
16. Niu, Q.; Li, B. Path Planning of Omnidirectional AGV Based on Simulated Annealing Genetic Algorithm. *Comput. Integr. Manuf. Syst.* **2023**, 1–21. Available online: <http://kns.cnki.net/kcms/detail/11.5946.TP.20220517.1105.014.html> (accessed on 25 May 2023).
17. Niu, Q.; Dong, X.; Fu, Y. Improved AGV Path Planning Based on Ant Colony Algorithm Heuristic Strategy. *J. Comput. Integr. Manuf. Syst.* **2024**, 1–19. [[CrossRef](#)]
18. Levenshtein, V. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.
19. Cao, J.; Zhu, R. A Genetic Algorithm Based on Adaptive Operators and Migration Strategy. *Mech. Sci. Technol.* **2003**, *2003*, 64–66.
20. Xu, X.; Yu, X.; Zhao, Y.; Liu, C.; Wu, X. Global Path Planning of Mobile Robot Based on Improved Genetic Algorithm. *Comput. Integr. Manuf. Syst.* **2022**, *28*, 1659–1672. [[CrossRef](#)]
21. Chang, J.; Ren, Y. Robot Path Planning Based on Improved Genetic Algorithm. *Mod. Mach. Autom. Technol.* **2023**, *2023*, 23–27. [[CrossRef](#)]
22. Zhang, Z.; Ke, Z.; Zhou, J.; Qian, Q.; Hu, X. Robot Path Planning Based on Improved Multi-objective Adaptive Genetic Algorithm. *J. Xi'an Univ. Technol.* **2023**, 69–78. Available online: <http://kns.cnki.net/kcms/detail/61.1294.N.20221012.1019.002.html> (accessed on 18 August 2023).
23. Xu, L.; Liu, Y.; Wang, Q. Application of Adaptive Genetic Algorithm in Robot Path Planning. *Comput. Eng. Appl.* **2020**, *56*, 36–41.
24. Hu, C.; Feng, W. Optimization of Mobile Robot Path Based on Improved Adaptive Genetic Algorithm. *J. Lanzhou Univ. Technol.* **2011**, *37*, 41–45.
25. Zhang, X. Mobile Robot Path Planning Based on Genetic Algorithm. Master's Thesis, Xi'an University of Architecture and Technology, Xi'an, China, 2014.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.