

Research Article

Disordered and Multiple Destinations Path Planning Methods for Mobile Robot in Dynamic Environment

Yong-feng Dong,¹ Hong-mei Xia,² and Yan-cong Zhou³

¹*School of Computer Science and Engineering, Big Data Computing Key Laboratory of Hebei Province, Hebei University of Technology, No. 5340 Xiping Road, Shuangkou, Beichen District, Tianjin 300401, China*

²*School of Computer Science and Engineering, Hebei University of Technology, No. 5340 Xiping Road, Shuangkou, Beichen District, Tianjin 300401, China*

³*School of Information Engineering, Tianjin University of Commerce, Tianjin, China*

Correspondence should be addressed to Hong-mei Xia; 727001435@qq.com

Received 30 November 2015; Accepted 10 May 2016

Academic Editor: Sook Yoon

Copyright © 2016 Yong-feng Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the smart home environment, aiming at the disordered and multiple destinations path planning, the sequencing rule is proposed to determine the order of destinations. Within each branching process, the initial feasible path set is generated according to the law of attractive destination. A sinusoidal adaptive genetic algorithm is adopted. It can calculate the crossover probability and mutation probability adaptively changing with environment at any time. According to the cultural-genetic algorithm, it introduces the concept of reducing turns by parallelogram and reducing length by triangle in the belief space, which can improve the quality of population. And the fallback strategy can help to jump out of the “U” trap effectively. The algorithm analyses the virtual collision in dynamic environment with obstacles. According to the different collision types, different strategies are executed to avoid obstacles. The experimental results show that cultural-genetic algorithm can overcome the problems of premature and convergence of original algorithm effectively. It can avoid getting into the local optimum. And it is more effective for mobile robot path planning. Even in complex environment with static and dynamic obstacles, it can avoid collision safely and plan an optimal path rapidly at the same time.

1. Introduction

The development of science and technology makes smart home concept realizable, where mobile robots play a very important role. There is a heavy need in daily life for a robot to work continuously in an environment which contains disordered and multiple destinations. Path planning for a mobile robot working in such an environment is a challenging task and has attracted more and more attention of scholars worldwide. This paper presents disordered and multiple destinations path planning methods for mobile robot in an environment with dynamic obstacles. Path planning problem can be understood as an optimization problem with constraints. It refers to how a robot searches one optimal or approximates optimal route with specific performance (such as the shortest distance, less time, or the minimum

energy consumption) from starting point to target point in the environment with obstacles.

There are many studies in the literature about path planning. Depending on the methods and principles used in the path planning, the robot path planning methods are divided into two types: traditional path planning methods and intelligent path planning methods. Traditional path planning methods include visibility graph method, grid decoupling method, graph searching method, and artificial potential method [1]. Intelligent path planning methods include fuzzy logic [2], ant colony algorithm [3], neural networks [4], and genetic algorithm [5]. In order to solve the shortcomings of these algorithms, scholars have conducted a lot of researches. In some particular work environments, the ants will fall into traps. Yu and Shiyong [6] proposed the ant rollback strategy to solve this problem, which could reduce

the operating efficiency of the algorithm. By analysing the shortcomings of the artificial potential field method, Tang et al. [7] proposed an obstacle avoidance method based on gravity chain. For solving the “dead cycle” problem in U-shaped obstacles, a systemic neural-fuzzy control algorithm was proposed by Bao et al. [8]. An Obstacle Avoidance Algorithm (OAA) and Distinguish Algorithm (DA) were proposed to generate the initial population by Yun et al. [9] in order to improve the efficiency by selecting only the feasible paths during the evolution of genetic algorithm. But it inhibits the diversity of the population. To overcome the premature convergence in the traditional genetic algorithm, Hu and Feng [10] proposed a new way to generate the genetic operator, by means of modifying insert a space before (SAGA). In order to avoid infeasible path and improve the convergence speed, Tuncer and Yildirim [11] proposed a new mutation operation, which selected the mutated node with the best fitness value after being replaced with the original node (BMFGA). In dynamic uncertain environments, Wei et al. [12] proposed the incremental replanning which reused the information of previous. At the same time, it could also reduce the possibility of finding the optimal path. Zhu [13] used the virtual ants to predict the potential collision with the moving obstacles. The local motions planning for avoiding collisions were scheduled under the ACO, which can plan optimal path rapidly in cluttered environment. These algorithms solve only the problem with certain aspects, but they are not very comprehensive consideration. Some algorithms do not consider the operating efficiency, or the probability of finding the optimal path is too low, or the ability of real-time processing is poor. This paper presents a simple and effective dynamic obstacle avoidance strategy.

2. The Mobile Robot Path Planning Based on the Evolutionary Genetic Algorithm

2.1. Encoding Scheme. In this paper, we set the sequence number for each grid. The basic sequence is from left to right, top to bottom. The upper border is x -axis, and the left border is y -axis. We use grid method to model the working environment as the paper [14]. Minkowski and the principle of expanding obstacles [15] are adopted to divide the work space into a number of grids. In the grid domain, a black grid denotes an obstacle and a white grid denotes free space. Robot path is indicated by the index numbers, which can save memory and express more succinctly compared with the method of coordinate. What is more important is that the genetic operations (such as the selection operator, crossover operator, and mutation operator) are simpler.

2.2. Sequencing. There are disordered destinations in the task of robot in practical work. For example, the robot goes to the kitchen to get a cup and pour water, goes to the living room to take drug, and finally goes to the bedroom. In the workflow, the sequence of robot taking drug and taking the cup is arbitrary. But the sequence of taking cup and pouring is determined. Therefore, the robot must determine the order of each destination according to the work environment firstly.

We propose the sequencing rule to determine the order of each destination. The sequencing rule can be denoted by

$$C_i = O_{\text{num}} * w_{\text{num}} + L_{\text{pre}} * w_{\text{pre}} + L_{\text{next}} * w_{\text{next}}. \quad (1)$$

C_i is the fitness value of the disordered destination D_i . O_{num} is the number of barrier grids in the rectangle which diagonal is the line segment from the former destination (or starting point) to the destination D_i . L_{pre} is the length from the destination D_i to the former destination (or starting point). L_{next} is the length from the destination D_i to the next destination (or goal point). w_{num} , w_{pre} and w_{next} are the weight coefficients, and $w_{\text{num}} + w_{\text{pre}} + w_{\text{next}} = 1$. In considering the population diversity and the efficiency of finding the optimal path, through repeated testing, the best combination of weight coefficients is obtained ($w_{\text{num}} = 0.6$, $w_{\text{pre}} = 0.2$, and $w_{\text{next}} = 0.2$). If the value of C_i is smaller, D_i has the higher priority. Robot determines the destination sequence based on this standard. Within each branching process, execute genetic operation, find the optimal path, and then get the optimal path in the static environment.

2.3. Initialization of Population. A good initial population in the feasible region is very effective to enhance the speed of evolution. We present the law of attractive destination. According to the relative position of the starting point and destination in each branching process, we adopt different strategy to generate the set of feasible paths. We categorize the relative positions of the starting point and its associated destination into four types as shown in Figure 1. Figure 1(a) represents that the destination is in the first quadrant of the starting point. Figure 1(b) represents that the destination is in the second quadrant of the starting point. Figure 1(c) represents that the destination is in the third quadrant of the starting point. Figure 1(d) represents that the destination is in the fourth quadrant of the starting point. R represents the robot and D represents the direction of destination. When the robot chooses the next grid to move, we give priority to the grid marked with “1,” which is closer to the destination in the region whose diagonal is a line segment from the starting point to the destination. If the grids marked with “1” are not in the feasible region of robot, we consider the grids marked with “2.” If the grids marked with “2” are also not in the feasible region of robot, we consider the grids marked with “3.” At the same time, the fallback phenomenon appears. We will set the current grid as the barrier grid to enhance the speed of population initialization and avoid the circuitous path effectively. When the starting point and destination are in the same horizontal or vertical line, we use the method in Figure 1(a). Looking for the next grid, the same horizontal (or vertical) coordinate is expanded 3 units from the left and right, respectively, and forms the search rectangle.

If the robot selects the next grid all in accordance with the law of attractive destination in the process of initial population, it is easy to fall into the local optimal. Therefore, when the number of walking steps is an integer multiple of 3, the robot selects the next walking grid randomly in the feasible region, which can increase the diversity of population.

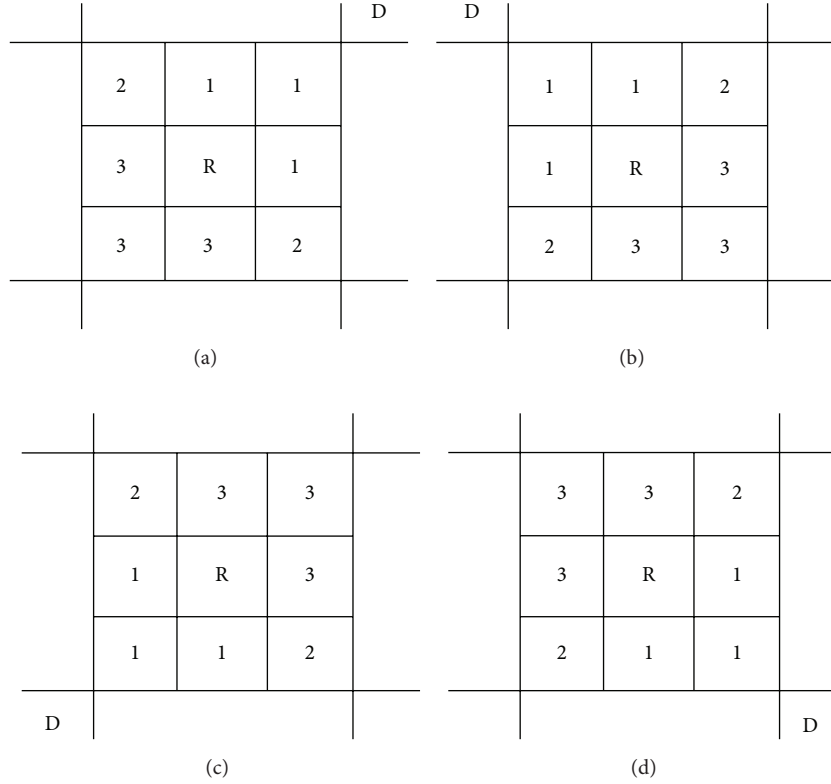


FIGURE 1: Schematic diagram of direction selection priority.

After the initialization, we simplify the individuals. We will delete the path between the repeated grids, which can improve the quality of initial population. For example, the original individual is (0, 1, 11, 13, 22, 32, 34, 23, 24, 34, 45, 56, 66, 77, 87, 88, 99); the individual is (0, 1, 11, 13, 22, 32, 34, 45, 56, 66, 77, 87, 88, 99) after being simplified.

2.4. Fitness Function. In the process of walking, the path length and energy consumption are the two important considered factors. The energy consumption is mainly correlated with the number of turns [16]. According to the above analysis, we adopt the fitness function as follows:

$$F = \frac{100}{(1 + 1/\sqrt{1+n}) \times D \times w_d + \text{curve} \times w_c}, \quad (2)$$

where n is the number of all grids in the path, $1/\sqrt{1+n}$ is a correction item, D is the length of path, and curve is the total number of turns. w_d and w_c are the weights, and $w_d + w_c = 1$. According to the requirement about these three standards, the weight coefficient is customized. This paper sets $w_d = 0.8$, $w_c = 0.2$. We define the path length by Euclidean distance, so the length of an edge with the points $p_i(x_i, y_i)$ and $p_{i+1}(x_{i+1}, y_{i+1})$ is given as follows: $d = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$. So the total length is $D = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$.

2.5. Design of Genetic Operator

2.5.1. Selection Method. The main principle of the selection method is that the best genes on the chromosome should be survived and transferred to next generation. So we use the roulette wheel method to choose the next generation population, which is good at protecting the better individuals. The better path has the higher probability to be selected as the next generation.

2.5.2. Crossover Operator and Mutation Operator. In this paper, we adopt a new improved adaptive genetic algorithm, which changes the crossover probability and mutation probability adaptively, according to the population and the distribution of fitness values. Change curves of the crossover probability and mutation probability are from the oscillation to stability gradually. In the early evolution, it can generate the larger cross probability and mutation probability, which can enhance the search ability. In the later evolution, we take a relatively low crossover probability and mutation probability to make the best individual preserved, which can weaken the bad influences caused by the too larger crossover probability and mutation probability that are generated because of the fitness close to the average fitness values or maximum fitness values. At the same time it also overcomes the evolutionary

stagnate and the problem of the local optimum. SAGA is expressed as follows:

$$\begin{aligned}
 & p_c \\
 & = \begin{cases} \frac{p_{c1} + p_{c2}}{2} + \frac{p_{c1} - p_{c2}}{2} * \sin\left(\frac{f' - f_{avg}}{f_{max} - f_{avg}} * \frac{\pi}{2}\right) & f' \geq f_{max} \\ p_{c1} & f' < f_{avg} \end{cases} \\
 & p_m \\
 & = \begin{cases} \frac{p_{m1} + p_{m2}}{2} + \frac{p_{m1} - p_{m2}}{2} * \sin\left(\frac{f' - f_{avg}}{f_{max} - f_{avg}} * \frac{\pi}{2}\right) & f' \geq f_{max} \\ p_{m1} & f' > f_{avg} \end{cases}
 \end{aligned} \tag{3}$$

where p_c and p_m represent the mutation probability value and mutation probability value, respectively, f_{max} represents the maximum fitness value in the population, f_{avg} represents the average fitness value in the population, f' represents the larger values between the two crossed individuals, p_{c1} and p_{c2} are the maximum and minimum crossover probability values, respectively, and p_{m1} and p_{m2} are the maximum and minimum mutation probability values, respectively. The single-point crossover is adopted in the adjacent chromosomes. The point which can make the path owing the maximum fitness value is the mutation point in the mutation points set.

2.6. The Cultural Algorithm. The main idea of the cultural algorithm is that individuals in the population space have individual experience during the evolutionary process, and the individual experience will influence the belief space through $Accept()$ function. In belief space, after the formation of the group experience, the behavior of individuals in the population space will be modified by $Influence()$ function, in order to enable individuals to achieve higher evolutionary efficiency [17]. The basic framework of cultural algorithm is as shown in Figure 2.

In the population space, the algorithm mainly completes the selection, crossover, and mutation operations. The concepts of reducing turns by parallelogram and reducing length by triangle are adopted by the belief space.

Through reducing turns by parallelogram [18], we can reduce energy consumption in the robot walking and enhance the path fitness value, as shown in Figures 3 and 4. We select three consecutive break points. If line 1 from the first break point to the node which is in front of the first break point is parallel to line 2 from the second break point to the third break point, we make the three break points as the three points of a parallelogram. Then we add an extra edge to draw a complete parallelogram based on the three points. If the extra edge does not contain any barrier grid, we remove the original path of the parallelogram, join the extra edge, and form a new path. That will reduce the three break points to two or one.

As we all know, the hypotenuse is the shortest in the right angled triangle. We propose a right angled triangle optimization strategy using this theorem. The length of the path after optimization is shorter than the original. The central idea of the strategy is to find a right angle and two angled sides. Intuitively, this is to draw the hypotenuse as long as possible while it is feasible. The effect drawings are

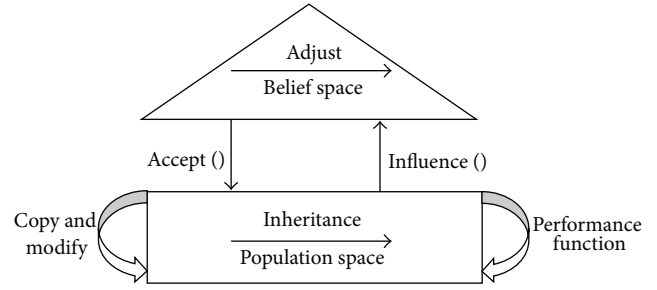


FIGURE 2: Basic framework of cultural algorithm.

| | | | | | | | | | | |
|----|----------------|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | S ₀ | | | | | | | | | |
| 1 | | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 2 | 10 | | | | | | | | | |
| 3 | | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 4 | 20 | | | | | | | | | |
| 5 | | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 6 | 30 | | | | | | | | | |
| 7 | | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
| 8 | 40 | | | | | | | | | |
| 9 | | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 10 | 50 | | | | | | | | | |
| | | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 |
| | 60 | | | | | | | | | |
| | | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| | 70 | | | | | | | | | |
| | | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| | 80 | | | | | | | | | |
| | | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |
| | 90 | | | | | | | | | G |

FIGURE 3: Original path diagram.

as shown in Figures 5 and 6. The whole flow chart is as shown in Figure 7.

In order to improve the convergence speed, we adopt the population replacement strategy from literature [17]. Replacement rate is p ($0 < p < 1$). The offspring may destroy the excellent parent individuals, after the selection, crossover, mutation operation, and optimization in the belief space. Therefore, keeping the excellent parent individuals can accelerate the convergence speed of the algorithm. Offspring individuals with poorer fitness are replaced by the parent individuals with better fitness. In order to improve convergence speed and the global searching capability of the algorithm, we set p from 0.3 to 0.6, as in the literature.

3. Routing Optimization Based on the Evolutionary Genetic Algorithm in the Static Environment

To deal with the multiple destinations after the orders being determined, we take the following activities within each branching process. (1) Generate the high quality initial population, in the feasible region. (2) Put forward the appropriate

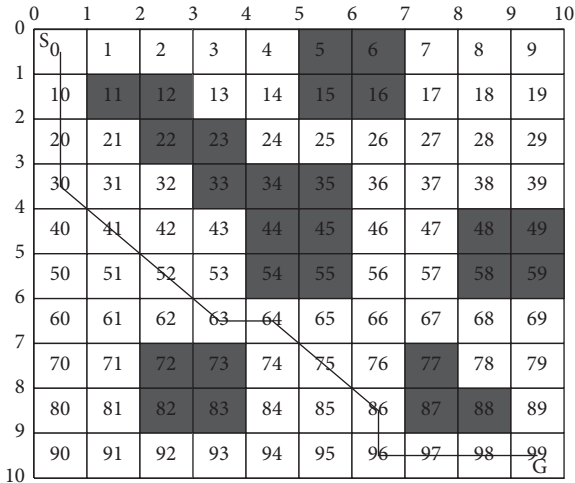


FIGURE 4: Path diagram after reducing the break points.

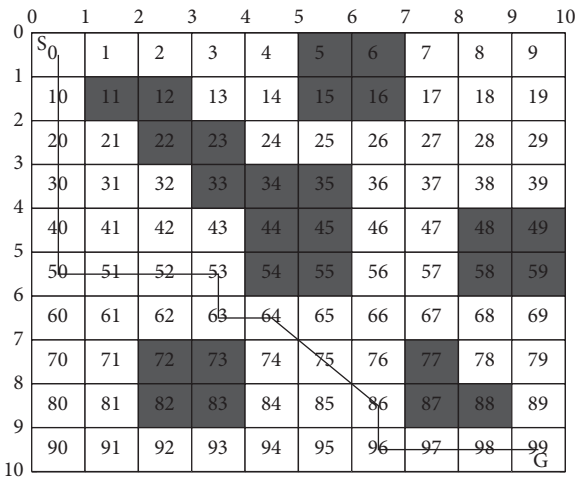


FIGURE 5: Original path diagram.

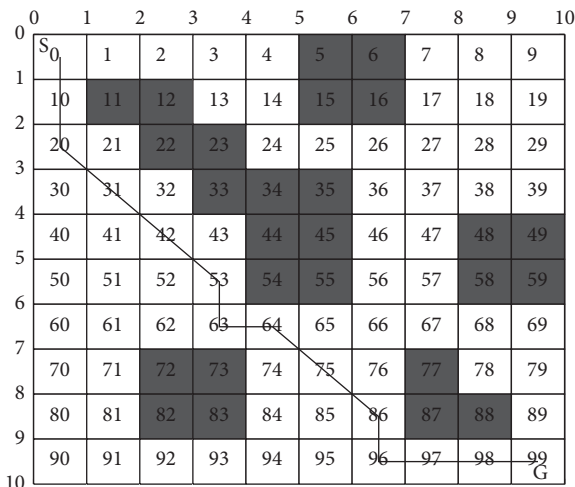


FIGURE 6: Path diagram after optimization.

fitness function. (3) Adopt adaptive genetic algorithm and get different crossover probability and mutation probability for different population. (4) Enhance the quality of path by reducing turns by parallelogram and reducing length by triangle in the belief space. The whole algorithm flow chart is as shown in Figure 8.

The general steps for solving the routing optimization problem by cultural-genetic algorithm are as follows.

The steps in the population space are as follows.

- (1) Grid method is used to model the working environment. The sequencing rule is used to determine the order of each destination. Set the number of destinations goal_num = 1.
- (2) Initialize the algorithm. Generate the feasible path of robot path which is the initial population in shunt process by the law of attractive destination. Set the current experiment iterations $t = 1$.
- (3) According to the fitness function, calculate fitness of each individual in the population; then the roulette wheel selection is used to produce the new population.
- (4) To get new chromosomes, we use the single-point crossover in the adjacent chromosomes.
- (5) To produces new individuals, mutation operation is performed on the part of the individuals.
- (6) According to the Accept() function, the new population is passed to the belief space.
- (7) According to the function of Influence() in the belief space, we get the parent population for new generation.
- (8) If the termination condition is satisfied, output the best individual of shunt process and go to step (10). Otherwise, continue to optimize the operation.
- (9) Set iteration $t = t + 1$ and go to step (3).
- (10) If it arrives at the final destination, the robot's final optimal individual is spliced by the best individual of shunt process. Otherwise, set iteration goal_num = goal_num + 1 and go to step (2).

The steps in the belief space are as follows.

In the belief space, we use the concept of reducing turns by parallelogram and reducing length by triangle to improve the quality of population. The strategy of population replacement is used to get the new parent population. Then, we pass the new parent populations to the population space.

4. Dynamic Path Planning

In a dynamic environment, the position of dynamic obstacle changes with time. In order to simplify complexity, we make the following assumptions.

- (1) Simulate the size of a dynamic obstacle as a grid. Dynamic obstacle walks integral multiple of grids in a unit time.

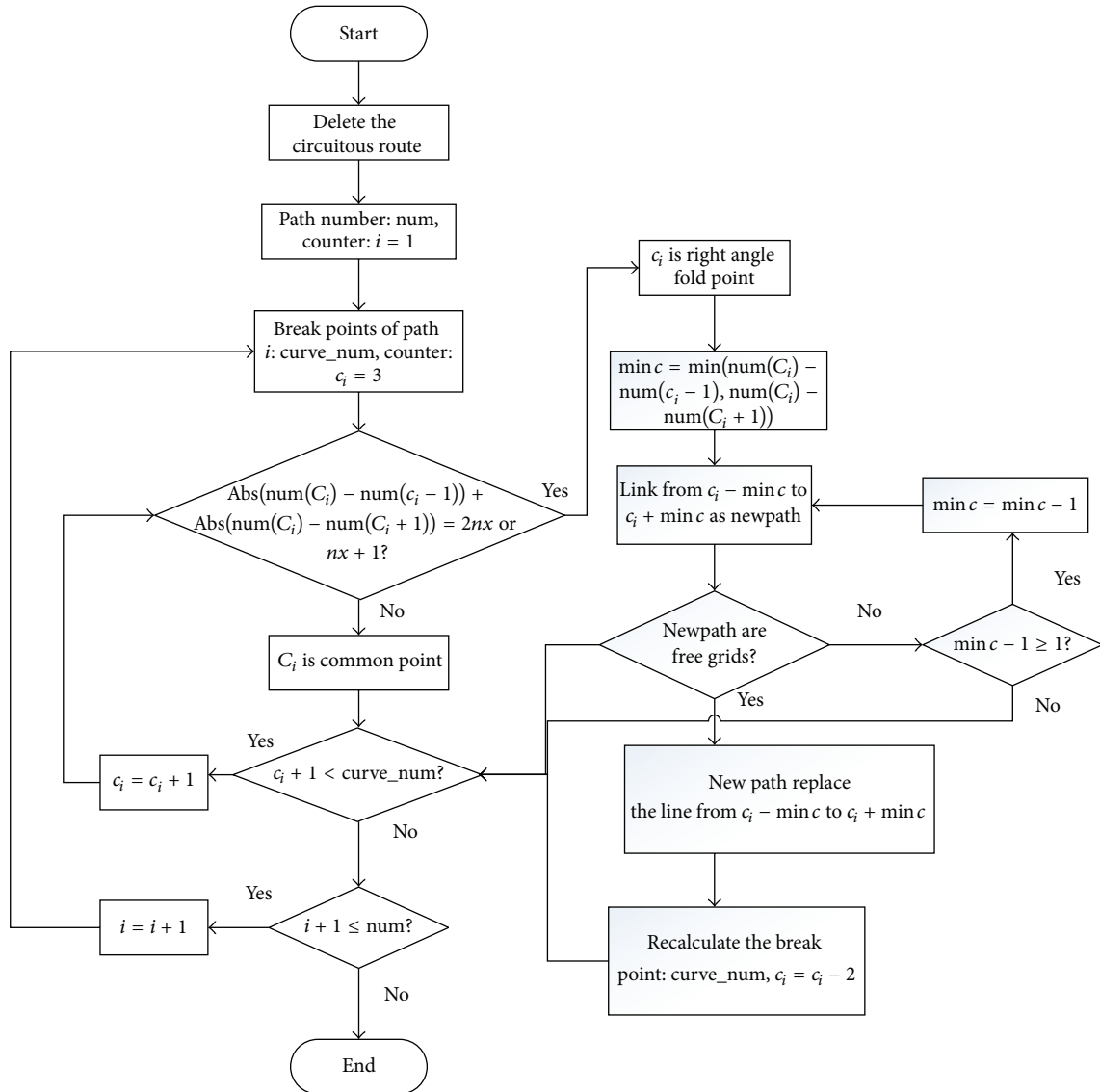


FIGURE 7: Flow chart of right angled triangle optimization strategy.

- (2) The movement of the dynamic obstacles is in uniform linear motion and speed is not known.
- (3) The robot can detect the movements of obstacles in the domain while walking.

The robot walks along the optimal path of static environment, which has been discussed in last section. The speed and the direction of the robot are R_{dir} and R_v . Visual domain radius is r unit grid. Through the simulation of virtual collision, we can predict the collision point and the speed of dynamic obstacle (D_v) and the direction of dynamic obstacle (D_{dir}). Therefore, we can choose different obstacle avoidance rules for obstacle avoidance.

We divide the collision into three types: (1) side collision: when the robot and dynamic obstacle have collision, their direction of motion is not in a straight line, (2) collision

after turning: When the robot and dynamic obstacle have collision, their direction of motion is in a straight line, but the moving direction of robot last step and the moving direction of dynamic obstacle are not in a straight line, and (3) other collisions.

Different strategies have been applied for different collision types under specific conditions as shown in Table 1.

5. Simulation

5.1. *Simulation Environment and Parameter Settings.* All parameters are shown in Table 2.

5.2. *Simulation Results in Static Environment.* In order to study the efficiency of the algorithm, we set one destination in the work environment firstly, as shown in Figure 3.

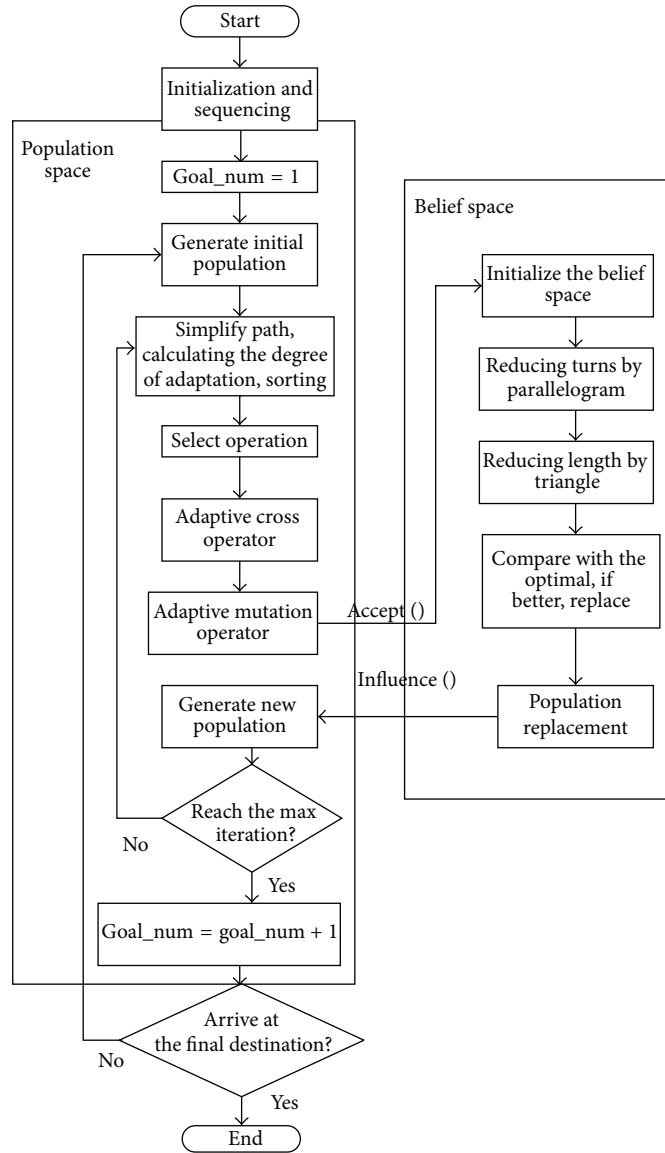


FIGURE 8: Flow chart of improved genetic algorithm.

The 0 and 99 are the starting point and the destination point, respectively.

We make the comparison with three algorithms: the evolutionary genetic algorithm (EGA), the BMFGA, and the SAGA. The three algorithms are all run 50 times, where the success probability and average fitness of the three algorithms are compared. The results are shown in Figure 9 and Table 3.

Randomly select one result from the 50 running times; the optimal fitness convergence results are shown in Figure 10. From the figures, we can conclude that the EGA has better convergence than others.

What can we see from the experimental results is that EGA has the biggest probability of searching the optimal path, with fastest convergence speed. From the above results, EGA

has higher success rate, and the convergence speed is also fast in the static environment, which has only one destination.

The working environment with disordered and multiple destinations is shown Figure 11, where S and G are the starting point and final destination, respectively, D1 and D2 are disordered destinations, and D3, D4, and D5 are destinations of determined sequence. According to (1), we can determine the running order, which is (S, D1, D2, D3, D4, D5, and G). By using this algorithm, we get the optimal path as shown in Figure 11.

5.3. Simulation Results in Dynamic Environment. Robot avoids dynamic obstacles in the dynamic environment as shown in Figure 12. The robot walks along the optimal path,

TABLE 1: Collision avoidance strategy.

| Type of collision | Collision avoidance strategy |
|-------------------------|--|
| Side collision | Waiting strategy: the robot waits for unit time in the current position, which can avoid the collision with dynamic obstacles. |
| Collision after turning | Rollback strategy: because the motion direction of the robot on the last step is not on the same line with the dynamic obstacle, the robot just needs back up one step, which can avoid collision. |
| Other collisions | Hiding in the nearby: the robot finds a point as the collision avoidance point in the vicinity of the current location. After avoiding dynamic obstacles successfully, it goes back to the current position. |

TABLE 2: Parameters.

| Population size | Number of iterations | P_{c1} | P_{c2} | P_{m1} | P_{m2} | p |
|-----------------|----------------------|----------|----------|----------|----------|-----|
| 150 | 150 | 0.75 | 0.25 | 0.2 | 0.01 | 0.3 |

which is shown in Figure 11. When the robot walks to the grid 60, it detects a dynamic obstacle in the grid 60. Through virtual collision, we get that the collision point is grid 81, and the type is side collision. According to Table 1, we choose the waiting strategy. The robot avoids collision successfully for the first time. The robot continues moving along the original path. When the robot walks to the grid 207, it detects a dynamic in the grid 201. Through virtual collision, we get that the collision point is grid 81, and the type is collision after turning. According to Table 1, we choose the rollback strategy first and then choose the waiting strategy. The robot avoids collision successfully for the second time. When moving to the grid 257, the robot detects a dynamic obstacle in the grid 297. Through virtual collision, we get that the collision point is grid 277, and the type is other collisions. According to Table 1, we choose the strategy of hiding in the nearby. The robot avoids collision successfully for the third time. The robot arrives at the final destination, as shown in Figure 12. We can see from the experimental results that the collision avoidance strategy is simple and effective.

6. Conclusion

Aiming at the working environment with disordered and multiple destinations in the smart home, we determine the order of destinations by the sequencing rule. Then the initial feasible path set is generated according to the law of attractive destination. We use adaptive genetic algorithm to optimize the population. The work above focuses on exploring the population space of cultural-genetic algorithm. We introduce the concept of reducing turns by parallelogram and reducing length by triangle in the belief space, which can improve the quality of population and accelerate the convergence speed. Static simulation results show that the EGA can search the optimal path quickly and accurately; it can solve the problems of local optimal and early maturing. Obstacle avoidance method in this paper is simple, efficient, and safe.

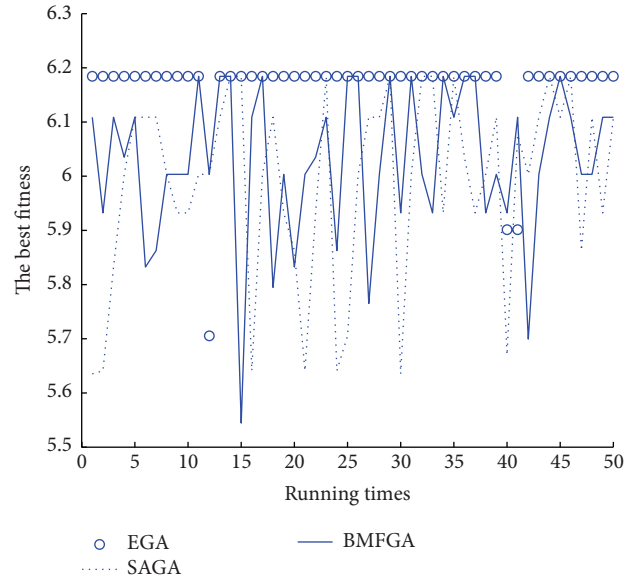


FIGURE 9: Convergence curves about the optimal fitness running 50 times.

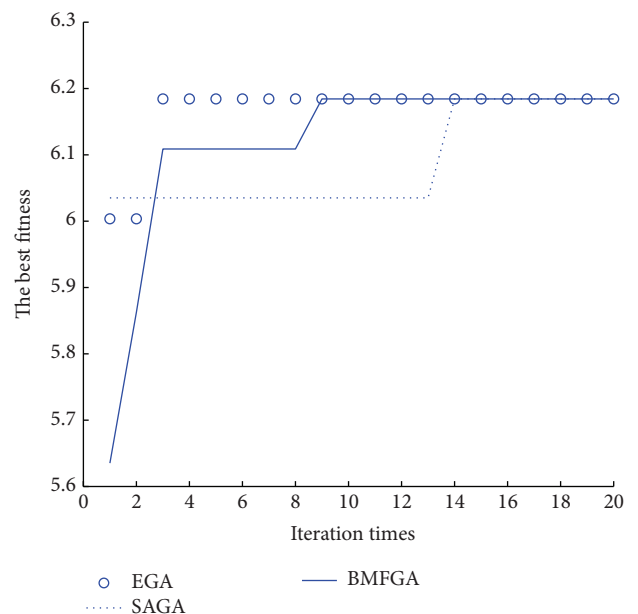


FIGURE 10: Convergence curves about the optimal fitness.

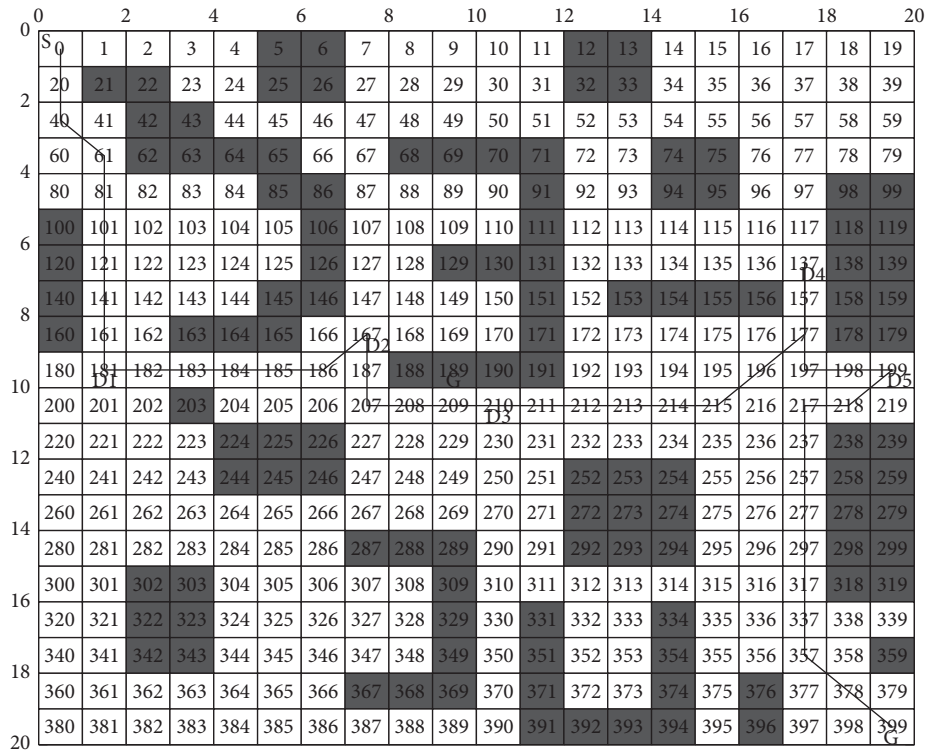


FIGURE 11: Ptimal path in the disordered and multiple destinations.

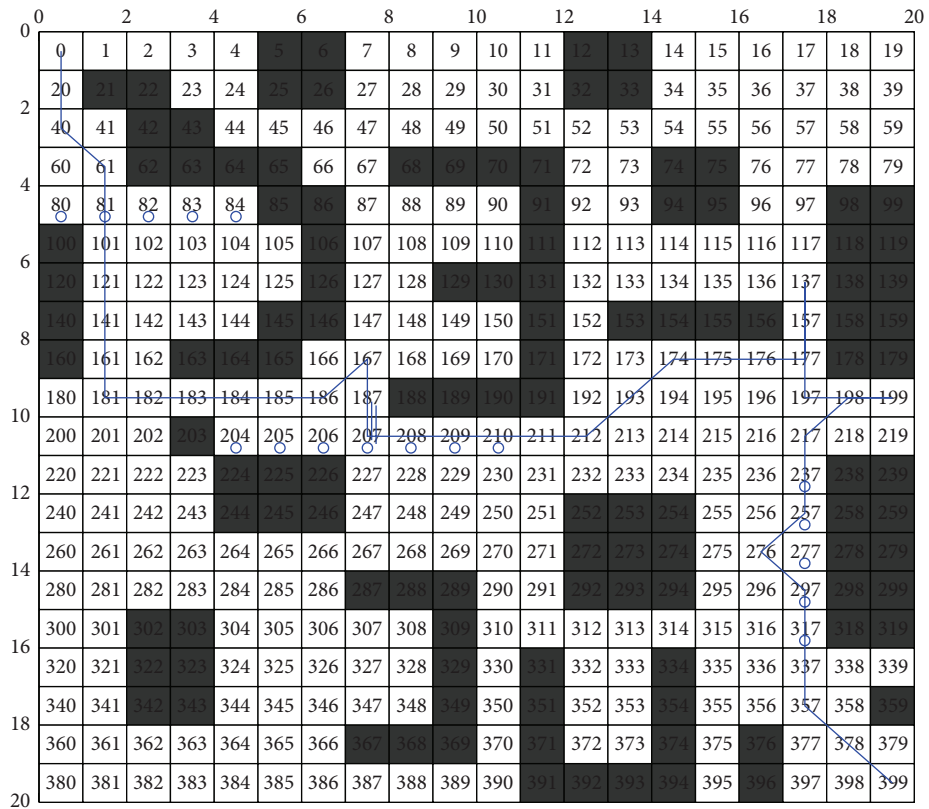


FIGURE 12: Arrival at the final destination.

TABLE 3: Performance comparison.

| Planning algorithm | Average fitness | Optimal fitness | The percentage of the optimal path |
|--------------------|-----------------|-----------------|------------------------------------|
| EGA | 6.1634 | 6.1843 | 95% |
| BMFGA | 5.9974 | 6.1843 | 12% |
| SAGA | 5.9880 | 6.1843 | 18% |

Competing Interests

The authors declare that they have no competing interests.

References

- [1] D.-F. Zhang and F. Liu, "Research and development trend of path planning based on artificial potential field method," *Computer Engineering & Science*, vol. 35, no. 6, pp. 88–95, 2013.
- [2] V. Ganapathy, C. Y. Soh, and J. Ng, "Fuzzy and neural controllers for acute obstacle avoidance in mobile robot navigation," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '09)*, pp. 1236–1241, July 2009.
- [3] M. Brand, M. Masuda, N. Wehner, and X.-H. Yu, "Ant colony optimization algorithm for robot path planning," in *Proceedings of the International Conference on Computer Design and Applications (ICCD '10)*, pp. V3-436–V3-440, IEEE, Qinhuangdao, China, June 2010.
- [4] W. Wang, S. M. Wei, Y. Q. Yang, Y. F. Jiang, and L. I. DuanLing, "Path planning for a mobile robot using neural networks," *Journal of Beijing University of Technology*, vol. 45, no. 10, pp. 221–225, 2009.
- [5] I. Al-Taharwa, A. Sheta, and M. Al-Weshah, "A mobile robot path planning using genetic algorithm in static environment," *Journal of Computer Science*, vol. 4, no. 4, pp. 341–344, 2008.
- [6] G. Yu and L. Shiyong, "Path planning for robot based on improved ant colony algorithm," *Computer Measurement & Control*, vol. 17, no. 1, pp. 187–189, 2009.
- [7] L. Tang, S. Dian, G. Gu, K. Zhou, S. Wang, and X. Feng, "A novel potential field method for obstacle avoidance and path planning of mobile robot," in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT '10)*, vol. 9, pp. 633–637, July 2010.
- [8] F. Bao, Y.-H. Pan, and W.-B. Xu, "Dynamic path plan of mobile robot based on neural-fuzzy control system," *Computer Engineering and Applications*, vol. 45, no. 10, pp. 221–225, 2009.
- [9] S. C. Yun, V. Ganapathy, and L. O. Chong, "Improved genetic algorithms based optimum path planning for mobile robot," in *Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision (ICARCV '10)*, pp. 1565–1570, Singapore, December 2010.
- [10] C.-B. Hu and W.-Y. Feng, "Optimization problem of mobile robot route with improved adaptive genetic algorithm," *Journal of Lanzhou University of Technology*, vol. 37, no. 5, pp. 11–15, 2011.
- [11] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Computers and Electrical Engineering*, vol. 38, no. 6, pp. 1564–1572, 2012.
- [12] W. Wei, D.-T. Ouyang, S. Lu, and Y.-X. Feng, "Multi objective path planning under dynamic uncertain environment," *Chinese Journal of Computers*, vol. 34, no. 5, pp. 836–838, 2011.
- [13] Q.-B. Zhu, "Ants predictive algorithm for path planning of robot in a complex dynamic environment," *Chinese Journal of Computers*, vol. 28, no. 11, pp. 1898–1892, 2005.
- [14] J. Ma, Z.-P. Fan, and L.-H. Huang, "A subjective and objective integrated approach to determine attribute weights," *European Journal of Operational Research*, vol. 112, no. 2, pp. 397–404, 1999.
- [15] S. Sun and Y. Qu, "Robot motion planning using genetic algorithm," *Journal of Northwestern Polytechnical University*, vol. 16, no. 1, pp. 79–83, 1998.
- [16] Z.-Y. Deng and C.-K. Chen, "Mobile robot path planning based on improved genetic algorithm," *Machinery Design & Manufacture*, vol. 7, pp. 147–149, 2010.
- [17] J. Cao and C. Sun, "Path planning based on multi-objective genetic algorithm for AUV on VCF electronic chart," *Geomatics and Information Science of Wuhan University*, vol. 35, no. 4, pp. 441–445, 2010.
- [18] Y.-C. Zhou, Y.-F. Dong, H.-M. Xia, and J.-H. Gu, "Routing optimization of intelligent vehicle in automated warehouse," *Discrete Dynamics in Nature and Society*, vol. 2014, Article ID 789754, 14 pages, 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

