Path Planning for Wheeled Mobile Service Robots based on Improved Genetic Algorithm

Shuang Li
Department of Informatics
University of Hamburg
Vogt-Kölln-Straße 30 22527
Hamburg
sli@informatik.uni-hamburg.de

Hongzhuo Liang
Department of Informatics
University of Hamburg
Vogt-Kölln-Straße 30 22527
Hamburg
liang@informatik.uni-hamburg.de

Jianwei Zhang
Department of Informatics
University of Hamburg
Vogt-Kölln-Straße 30 22527
Hamburg
zhang@informatik.uni-hamburg.de

ABSTRACT

Path planning of wheeled mobile service robots plays an important role in the autonomous navigation system, which uses an optimization algorithm to intelligently search the collision-free path from the starting point to the target point in the known environment. In this paper, we propose a robust improved genetic algorithm (GA) which integrates initial path, prefers elite individuals, improves hybrid crossover operator and optimizes mutation operator. Accurate solutions can be found in the grid map and suboptimal extrema are robustly avoided. Through analyzing the quality and fluctuation of the optimal solution, and its convergence speed, our experimental results shown on MATLAB demonstrate that the presented algorithm performs more robustly and efficiently than traditional elite strategy GA.

CCS Concepts

•Computing methodologies → Motion path planning;

Keywords

path planning; genetic algorithm; grid method

1. INTRODUCTION

Intelligent service robots technology [6] is developing rapidly in recent years. Path planning technology [1] for wheeled mobile service robot, whose most crucial parts are environmental modeling and path planning algorithm, plays a significant role in the intelligent navigation system. Therefore, this paper mainly focuses on the global path planning technology of wheeled service robots to efficiently find the shortest collision-free path from the start point to the target point in two-dimensional static environments by a proper selection of the environmental modeling method and further study on improved path planning algorithm.

HCR 2018 July 14–16, 2018, Shanghai, China

DOI:

The main contribution of this paper is the proposal of the improved GA, which is optimized and improved from these five aspects, *i.e.*, integrating the initial path, efficiently choosing the elite individual, improving hybrid crossover operator and optimizing mutation operator based on the grid method. The feasibility and robustness of this improved GA in planning for the path of mobile robots are verified based on the MATLAB simulation.

2. RELATED WORK

The popular environmental modeling methods in mobile robots path planning are free space method, the grid method and other hybrid representation methods that combine metric and topological features. Considering that the real working environment of service robots has multiple and scattered obstacles, we build the environmental model by the grid method which has a strong capability to express the environment.

Recently many scholars have proposed some path planning methods using convolutional neural network [5] or reinforcement learning [4] that work pretty well in the trained environment but need lots of data. On the other hand, there are various intelligent path planning methods such as A* algorithm [2] and genetic algorithm [7]. Most of these algorithms still have a higher computational complexity and an easier negligence of the applicability of environment modeling technology and the path planning algorithm. Although genetic algorithm with elitist strategy [3] has good global search capability, high robustness, and greater flexibility in terms of path planning, the issues of the local optimal solution, slow convergence, low stability and other shortcomings still remain. To remedy these problems, we definitely improve genetic algorithms with elitist strategy in this paper.

3. ENVIRONMENTAL MODELING

3.1 Environmental Map

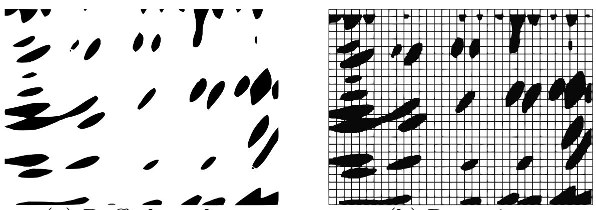
Before fully rasterize the map we need to convert the RGB map to the binary map. Assuming the service robot as a particle, according to the spacing constraints between service robot and obstacles, we puff obstacles appropriately. We choose proper grid size a which is determined together by the area of obstacles and the volume of service robot, then rasterize the image information. Figure 1 shows the puffed map and the rasterized map.

We classify the grid into fully viable grids, completely infeasible grids, and partially viable grids based on the location of obstacles. To guarantee that service robot could safely pass through the feasible region, all incompletely viable grids are marked as infeasible grids. Filling the infeasible rasters with black, the fully rasterized environmental model is shown in Figure 3. All in all, the environmental information represents in the evenly sized, closely adjacent raster arrays.

Use the grid method to design a map, each grid needs to be identified. In accordance with the order from left to right, from bottom to top, we set the bottom left corner as the origin of coordinates and the start of grid node. Every grid can use rectangular coordinates (x, y) and a number n as a unique identification. In order to facilitate the determination of next feasible path node of the service robot, it's necessary to create an adjacency matrix used to store the adjacent information for each grid. Suppose there are N grids on the map M, we establish $N \times N$ two-dimensional adjacency matrix

$$D[m][n] = \begin{cases} 1, & m \in \mathbb{A} \\ 0, & \texttt{else} \end{cases} \tag{1}$$

where $m, n \in [1, N]$, $\mathbb{A}$ denotes the grids that adjacent to grid n, $D[m][n] = 1$ means the service robot can move between grid m and grid n, and vice versa.



(a) Puff obstacles (b) Rasterize

Figure 1: Process of building environmental model based on the grid method.

3.2 Motion Hypotheses

We assume following the hypotheses of mobile robot in grid map: 1) Mobile robot can't cross any obstacle and only move in the 2D static environment; 2) Mobile robot has constant linear and angular speed; 3) Every step of the mobile robot is from the centroid of one grid to the centroid of next neighbor grid; 4) Mobile robot has 8 directions of motion because every grid has 8 neighbor grids except the grids on the boundary.

4. ALGORITHMIC SOLUTION

We propose an improved elite strategy genetic algorithm from four aspects: use the integrated method to optimize initial paths; select elite individuals according to the number of slope changes of the path; propose a hybrid crossover operator depends on the coincidence points of two paths; optimize the mutation operator and add gene examination of the mutated individual.

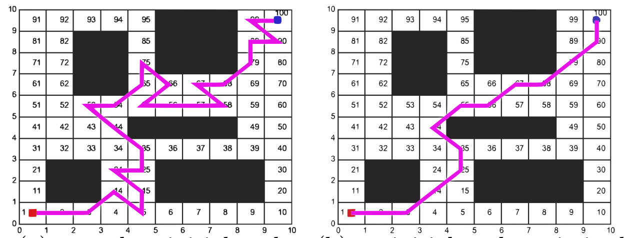
4.1 Genetic Encoding

Let p define the genotype of an individual, so each gene p_i is assigned to a real-valued number that is related to a corresponding grid node of the map, see equation (2). Hence, according to the feasible node information stored in the adjacency matrix of each grid, every genotype implicitly represents a collision-free path from the starting point S to the target point T.

$$p = (p_1|p_2|p_3|...|p_{i-1}|p_i) \tag{2}$$

4.2 Initial Path Optimization

In order to eliminate the redundant bypass path in the initial individuals, an integrated method is proposed to optimize initial paths. Finding out all the feasible nodes after the current node in an individual, we discard the first feasible node and the redundant nodes among the first feasible node and the last feasible node. After simplifying the path, the initial path for genetic operation is mainly located in the diagonal area between the initial point and the target point, effectively improving the convergence speed of the algorithm. For example, in 10×10 grid environment, Figure 2(a) shows an initial individual, and Figure 2(b) shows the integrated path optimized by our method.



(a) a random initial path (b) an initial path optimized by the integrated method

Figure 2: Illustration of integrated method for initial paths.

4.3 Fitness Function

We define $L(p)$ (3) as that the length of the collision-free path p from the starting point S to the target point T. This means if the path length is shorter, the quality of a path is higher.

$$L(p) = \sum_{i=1}^{w-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \tag{3}$$

Where w denotes the number of nodes from S to T, (x_i, y_i) denotes the coordinates of point p_i, (x_{i+1}, y_{i+1}) denotes the coordinates of point p_{i+1}.

The following is an efficient design of the fitness function $F(L(p))$ (4). Measuring the fitness f is typically one of the most crucial challenges in evolutionary optimization. In our case, we simply define the fitness function f as:

$$f = F(L(p)) = \frac{1}{L(p) + c} \tag{4}$$

where c is conservative estimate cost of $L(p)$.

4.4 Elite Selection

The genetic algorithm with elitist strategy determines the elite individual E only with respect to the fitness f of each individual. Obviously, it ignores the two individuals which have the same fitness rather than the same genotype. For example, there are some different routes which have the same length in the grid map, but it's obvious that the path with fewer slope changes is more viable for the actual work of service robots.

Therefore we propose a method of elitist selection which uses the individuals with the highest fitness and the least number of slope changes as elite individuals.

4.5 Parent Selection

The strategy of selecting the parents γ from the mating pool Γ to create new offspring is roulette selection. There is a probability that an individual selected in a population is proportional to its fitness and supports a good balance in search space exploration.

4.6 Order-based Crossover

To guarantee the continuity of recombined paths, the intersection nodes for the crossover operator are randomly selected from the overlapped nodes except S and T in two individuals. We utilize a hybrid crossover method with single-point or two-point crossover with the aim of effectively achieving a dynamic search space exploration while remaining sensitive to exploit local extrema.

However, the issue of discontinuous paths caused by two-point crossover still remains. An improved order-based crossover operator which judges whether the sequence numbers of all the coincident points in the two paths are in ascending order in Algorithm 1. Similarly, the integrated method for initial paths also exploits to optimize the individual path after crossover. The improved hybrid crossover method not only increases the search range of the solution space but also improves the convergence speed of the algorithm.

Algorithm 1 Improved crossover operator

Input: Parents γ, crossover probability w_c
Output: offspring p_{c1}, p_{c2}
1: randomly select two paths p_1, p_2 from γ based on w_c
2: calculate the number N_c of coincident points set C
3: **if** $N_c = 0$ **then**
4: go back to step 1
5: **else if** $N_c = 1$ **then**
6: set point $c \subset C$ as the crossover point
7: perform single-point crossover
8: **else**
9: **for** each item i in N_c **do**
10: **if** $p_1(c_i) > p_1(c_{i+1})$ and $p_2(c_i) > p_2(c_{i+1})$ **then**
11: $N_c = N_c - 1$, delete c_i
12: **end if**
13: **end for**
14: repeat step $2 \sim 7$
15: randomly select two crossover points $c_1, c_2 \subset c$
16: perform two-point crossover
17: **end if**
18: integrate p_{c_1}, p_{c_2};
19: **return** offspring p_{c_1}, p_{c_2};

4.7 Mutation

Single-point mutation operator refers to randomly applying small changes to the genotype where corresponding mutation probability w_m is chosen reasonably small. Concerning about generating continuous paths and maintaining the diversity of the population, it's indispensable to randomly assign the mutation value from the common feasible points of the two points before and after the mutation point. If the number of common feasible points is one, the mutated node will be reselected.

Furthermore, the mutated individuals are similarly integrated to reduce the redundant nodes and are regenerated if its genes are exactly same with the elite individual E.

5. EXPERIMENT

The proposed algorithm and basic GA with elitist strategy have been tested on the MATLAB platform by the following simulated settings: We assume mobile service robot as a particle, then find the shortest path from the starting point S to the target point T in a 26×35 grid map, as shown in Figure 3, where the red square point indicates S, and the blue circle indicates T.

The only parameters required in our algorithm are population size phi, the number of iterations I, the crossover probability w_c and the mutation probability w_m. Note that these parameters greatly influence the required computation time per generation and the challenge is achieving a fast convergence that robustly scores evolutionary progress. We empirically set these parameters as $\{M, I, w_c, w_m\} = \{100, 300, 0.8, 0.05\}$ for the two genetic algorithms. To prevent the premature convergence of our algorithm, when the number of iterations is 50 times larger, the crossover probability w_c and the mutation probability w_m are adjusted to 0.9 and 0.1, respectively.

Firstly, we compare the flexibility and performance of our algorithm and basic GA with elitist strategy. Figure 3 and Figure 4 demonstrate the shortest paths searched by our improved GA and GA with elitist strategy in 100 times experiments. We examine that the length of the shortest path represented by the solid line both are 42.6985, and the average path length of all individuals are 44.4777 and 52.132 respectively in terms of our improved GA and GA with elitist strategy. In addition, comparing the polyline of two shortest paths, it can be observed that the optimal path generated by the improved GA only turns 7 times whereas the one generated by the GA with elitist strategy requires 12 corners to reach T. Essentially, both algorithms reach the convergence in space, but the improved algorithm generates more viable paths and higher quality of the individuals in the evolutional process.

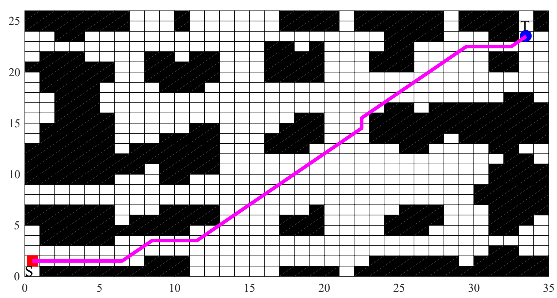


Figure 3: The shortest path searched by improved GA in 100 times experiments

We further investigate the robustness of our algorithm in 100 experiments. We record the maximum length L_{max},

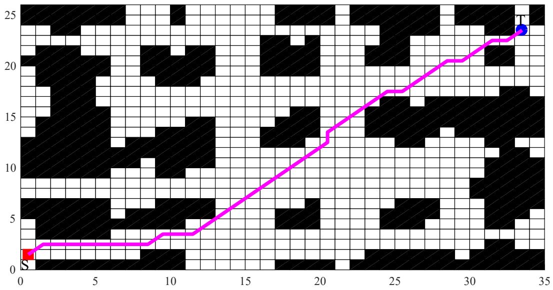


Figure 4: The shortest path searched by GA with elitist strategy in 100 times experiments.

Table 1: Comparision of two algorithms regarding L_{max}, L_{min}, $\overline{L}$, δ in 100 times experiments

GAs	L_{max}	L_{min}	$\overline{L}$	δ
improved GA	43.8701	42.6985	42.8156	0.2498
elite strategy GA	51.9706	42.6985	44.8779	1.5471

the minimum length L_{min}, and the average length $\overline{L}$ of the optimal solution and calculate the sample standard deviation δ. The results are listed in Table 1 and in Figure 5, and demonstrate that the algorithm can robustly evolve the optimal solution with small fluctuations and high concentration.

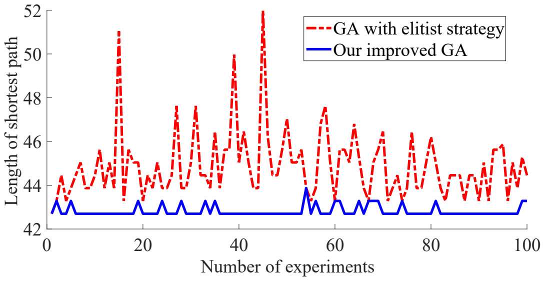


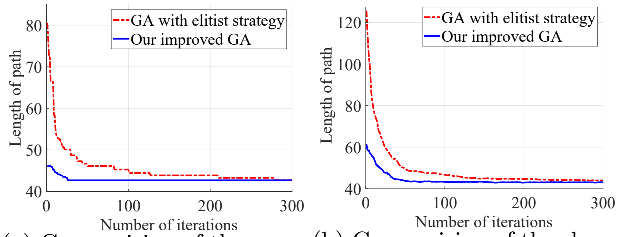
Figure 5: Comparision of two algorithms regarding optimal solution fluctuation in 100 experiments.

Lastly, the algorithm is tested for the efficiency and flexibility of our algorithm. Figure 6 visualizes the comparison between the convergence curves of the shortest path and the average path for two algorithms in one optimal solution. It can be observed that the shortest path and the average path of the improved algorithm quickly tend to converge, and the average path length of the improved algorithm is always less than that of the elite strategy algorithm.

6. CONCLUSION

This work propose a novel evolutional algorithm for solving path planning of wheeled mobile service robots in the grid map. Our algorithm is improved by four aspects: integrating initial path, selecting elite individuals according to the number of slope changes of the path, improving hybrid crossover operator and optimizing mutation operator. By analyzing the optimal solution, convergence rate and fluctuation of optimal solutions, it is verified that the improved GA has better efficiency, flexibility, and stability in the global path planning. In summary, the method can be used for global path planning tasks which require flexible and robust control of the navigation system.

Nevertheless, the current algorithm optimizes solutions in the static 2D environment, ignoring the dynamic obstacles such as human beings. Therefore, adding dynamic avoidance will be considered in future. Finally, an implementation of this algorithm needs to be tested in the actual navigation system.



(a) Comparision of the average path (b) Comparision of the shortest path

Figure 6: Comparision of the convergence cureves between the shortest path and the average path in one optimal solution.

Acknowledgment

This research is funded by the German Research Foundation (DFG) and the National Science Foundation of China (NSFC) in project Crossmodal Learning, TRR-169. And it is partially supported by project STEP2DYNA.

7. REFERENCES

[1] F. Fahimi. *Autonomous robots: modeling, path planning, and control*, volume 107. Springer Science & Business Media, 2008.

[2] A. K. Guruji, H. Agarwal, and D. Parsediya. Time-efficient A* algorithm for robot path planning. *Procedia Technology*, 23:144–149, 2016.

[3] Y. Liang and K.-S. Leung. Genetic algorithm with adaptive elitist-population strategies for multimodal function optimization. *Applied Soft Computing*, 11(2):2017–2034, 2011.

[4] R. Meyes, H. Tercan, S. Roggendorf, T. Thiele, C. Büscher, M. Obdenbusch, C. Brecher, S. Jeschke, and T. Meisen. Motion planning for industrial robots using reinforcement learning. *Procedia CIRP*, 63:107 – 112, 2017.

[5] M. Przybylski and B. Siemiątkowska. A new cnn-based method of path planning in dynamic environment. In *International Conference on Artificial Intelligence and Soft Computing*, pages 484–492. Springer, 2012.

[6] A. Salerno and J. Angeles. A new family of two-wheeled mobile robots: Modeling and controllability. *IEEE Transactions on Robotics*, 23(1):169–173, 2007.

[7] J. Tu and S. X. Yang. Genetic algorithm based path planning for a mobile robot. In *2003 IEEE International Conference on Robotics and Automation*, volume 1, pages 1221–1226. IEEE, 2003.