

Application of an Improved Ant Colony Algorithm to Optimal Path Planning for Automatic Guided Vehicles

weikang zhu (✉ zwk_miracle@163.com)

Shanghai University

jicheng liu

Shanghai University

Research

Keywords:

Posted Date: August 31st, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-59527/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Application of an Improved Ant Colony Algorithm to Optimal Path Planning for Automatic Guided Vehicles

Weikang Zhu¹ * and Jicheng Liu¹

Abstract

The path planning is the key technology of AGV path finding. This paper uses an improved ant colony algorithm to plan the path of an AGV. For avoiding the defects of traditional ant colony algorithm such as low smoothness of route and local optimal solution, the transition probability and pheromone update method are improved. Various actual turning situations are analyzed in the transition probability, the basis for defining the smoothing factor is provided by the Bezier curve, and a random selection operator is formed for updating local pheromone by extracting characteristic information of iterative process. The simulation results in different environments prove that the smoothing factor plays an important role in optimizing the smoothness of the path and the diversity of the constructed solutions, and the random selection operator is effective in solving the contradiction of the local optimal solution and in finding the optimal solution.

Keywords: Automatic guided vehicle (AGV), Path planning, Ant colony algorithm, Smoothness, Statistical analysis

0 Introduction

Path planning for automatic guided vehicles is a hot topic in the field of logistics research, and it is also a typical vehicle routing problem (VRP). Automatic guided vehicles are required to meet the requirements such as shortest time, shortest path, lowest energy consumption, etc., and search for an optimal or near-optimal safe path in a given working environment. At present, there are relatively traditional algorithms, such as the A* algorithm [1,2], Dijkstra algorithm and artificial potential field algorithm. In recent years, scholars have also proposed to apply bionic algorithm to path planning, such as genetic algorithm [3-6], ant colony algorithm [7-10], particle swarm algorithm [11-14], etc. the path smoothness is an important index for evaluating path planning algorithms. Huang Chen et al. [15] and Song Rui et al. [16] combined the A* algorithm with path smoothing to obtain smooth and high-quality paths. Baoye Song et al. proposed kinematic constraints to make the particle swarm algorithm get a smoother path [17].

The ant colony algorithm is a bionic evolutionary algorithm originated from nature. It was proposed by Marco Dorigo in 1991 [18]. It has the advantages of strong robustness, self-

Weikang Zhu
e-mail: zwk_miracle@163.com

Jicheng Liu
e-mail: liujichenghit@163.com

¹ School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China

organization and easy combination with other optimization algorithms, because the ant colony algorithm uses information positive feedback mechanism to randomly search for optimization, and adopts distributed computation. However, the traditional ant colony algorithm has some defects, such as long computation period, slow convergence speed, local optimal solution, etc. In order to overcome these defects, Marco Dorigo et al. proposed the elite ant system (EAS) [19], whose basic idea is to give extra pheromones to the optimal path of each cycle to enhance the effect of positive feedback, so that the algorithm converges faster and the computation period is shorter. Thomas Stutzel et al. proposed the MAX-MIN ant system [20], whose design idea is to limit the number of pheromones on the path to solve the stagnation problem, so that ants can obtain more search schemes in the initial stage of the algorithm. The rank-based ant system based on sorting proposed by Bullnheimer et al. is similar to the elite ant system, which sorts and volatilizes pheromones in each iteration.

This paper, aims to make the algorithm more accurate in obtaining the optimal solution. We improve the pheromone update method, extract the ant iteration information through statistical analysis and random selection methods, and then construct a random selection operator. In addition, on the basis of pseudo-random transfer probability, a smoothing factor after turning analysis is added to punish the route-finding ants, so as to improve the ants' searching ability and the smoothness of the path. Smoothness plays an important role in the practical application of path planning, because with the increase of the smoothness, the automatic guided vehicle can spend less time at corners and thus reduce energy consumption. This makes the algorithm more in line with the running process of the automatic guided vehicle in practical application.

We will carry out simulation experiments with the improved algorithm and the traditional algorithm in MATLAB, and the performance of path planning and smoothness of the algorithm is compared under different map complexity show that the improved ant colony algorithm has greatly improved the optimization ability of path planning and path smoothness.

1 Environmental model

The grid method is the most commonly used way to correspond grid information with environment. This method is easy to create, simplifies complex problems and reduces the amount of data computation. Therefore, the grid method is used to establish a two-dimensional space environment model of the vehicles. The grid method model is shown in Fig. 1. The standard for dividing the grid area is the unit moving step of the vehicle, the green grid is the starting grid, the grid number increases from left to right and from top to bottom, the red grid is the end grid, the black grid is the set obstacle, and the white grid is the free grid. N is the grid number and M is the grid edge length. The relationship between the grid number and the grid center coordinates is as follows:

$$\begin{cases} x = \text{mod}(N, M) - 0.5, & \text{if } x = -0.5, x = M - 0.5 \\ y = M + 0.5 - \text{ceil}(N/M), \end{cases} \quad (1)$$

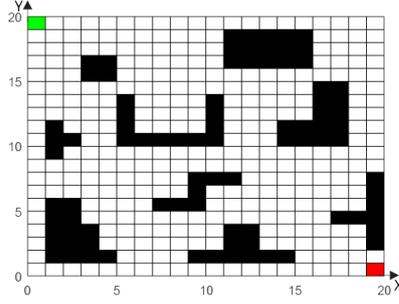


Fig. 1 Grid map

2 Ant Colony Algorithm

2.1 Principle of Ant Colony Algorithm

Ants in nature release a secretion (pheromone), during their foraging movement. As time goes by, the substance will volatilize gradually, so the concentration of pheromone is the strongest at the starting point or around the destination. The probability of ants selecting a path is proportional to the intensity of pheromone on the path. Of course, with the increasing intensity of pheromone, the probability of being selected increases. So repeatedly, finally make the shortest path is retained.

2.2 Model of Ant Colony Algorithm

As a probabilistic algorithm, the ant colony algorithm uses the pheromone quantity of optional transfer nodes to determine the next directions of ants. In the following formula, $P_{ij}^k(t)$ is the calculated value of ant position transfer probability, which indicates the transfer probability of ant k moving from position i to position j at time t .

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t)}{\sum_{S \in allowed_k} \tau_{iS}^\alpha(t) \cdot \eta_{iS}^\beta(t)}, & j \in allowed_k \\ 0, & otherwise \end{cases} \quad (2)$$

where $\tau_{ij}(t)$ represents the pheromone intensity of the current position to the target point at time t , $\eta_{ij}(t)$ is the heuristic information of the ant, that is, the reciprocal of the distance from the current position to the end point, expressed as $\eta_{ij}(t) = 1/d_{ij}$, both have a certain guiding role for the movement of ants. α and β respectively represent the weighted values of pheromone and visibility, and $allowed_k$ indicates the aggregate of transfer nodes that ant k is allowed to select next at node i .

Each ant updates its pheromone after completing an iteration, and each path pheromone is adjusted according to the following formula :

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3)$$

where m is the number of ants, $0 < \rho \leq 1$ is the evaporation rate of the pheromone, so $(1 - \rho)\tau_{ij}(t)$ indicates the residual pheromone after the path pheromone volatilizes, and $\Delta\tau_{ij}^k$ is the pheromone left by the k th ant in the path i to j , as shown in the following formula:

$$\Delta\tau_{ij}^k = \begin{cases} (C_k)^{-1}, & \text{if the } k^{th} \text{ ant traverses}(i,j) \\ 0, & \text{others} \end{cases} \quad (4)$$

where C_k is the total path length obtained by the k th ant after taking the complete path. As described above, the algorithm finally leads to the global optimization path.

2.3 Improved Ant Colony Algorithm

In order to overcome the shortcomings of ant colony algorithm, such as low smoothness and locality of optimal solution, in this paper we propose an improved ant colony algorithm based on the existing algorithm in the following way:

- (1) adding a smoothing factor w to the transition probability;
- (2) constructing random selection operators by statistical analysis to improve pheromone update method.

2.3.1 Improved Transition Probability

Transfer probability is the reference for selecting the next point to move. However, the vehicle will inevitably turn in the running process, and too many invalid turns will cause speed loss and time waste. Therefore, we analyze the effect of turning on the path trajectory in actual situations, and then punish the ants turning by adding a smoothing factor w , so that the ants have a greater probability of choosing a straight line to run in the moving process and reduce unnecessary turns.

Next, we analyze the turning process to make it more in line with the actual situation. Dividing the turning process into two types of discussions, one is the turning without obstacles in the range, the other is the turning with obstacles in the range. The path result obtained by the algorithm is ideal, so we need to analyze the actual turning process in order to define the smoothing factor w . When the turning node is determined, we use the vector curve drawing method, that is. the Bezier curve processing to draw the trajectory. The curve processing method has the properties of endpoint tangent vector, convex hull and slow change of curvature, and is suitable for processing the vehicle motion trajectory. We consider the Bezier curve is the moving path of the vehicle in actual situation. The following figure shows the turning without obstacles. The blue line is the ideal path and the red curve is the processed curve path. P_i , P_{i-1} , P_{i+1} are path nodes. The picture shows that the actual path is shorter than the ideal path.

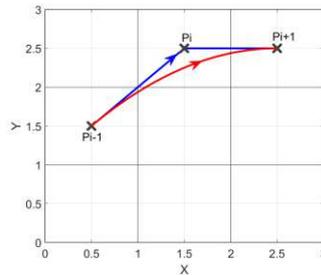


Fig. 2 Barrier-free turn

We analyzed the turns with obstacles in the range. Furthermore, it is found that there are two kinds of ideal situations with large deviation from the actual turning curve. As are shown in the following two figures, the upward figure shows that the angle θ is 135° , and the downward figure shows that the angle θ is 90° .

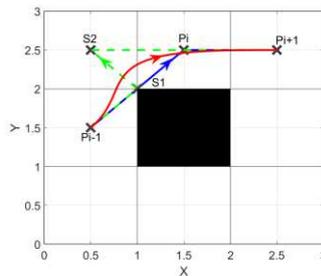


Fig. 3 Barrier turn ($\theta = 135^\circ$)

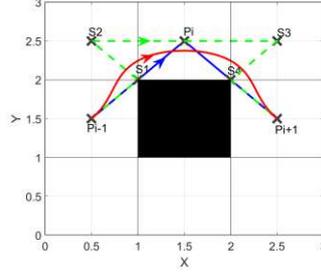


Fig. 4 Barrier turn ($\theta = 90^\circ$)

Black squares in the two figures represent obstacles, the points P_i , P_{i-1} , P_{i+1} respectively represent the previous point, current point and next path point obtained by the algorithm, blue straight line is the ideal path line, and the moving direction is indicated by arrows in the figures. Due to the defects of grid method, we take point S as the control point of the Bezier curve. The processing straight line represented by the green dashed line is constructed, and the red Bezier curve is finally obtained, and then the length of the Bezier curve is approximately computed by the numerical method. The following table 1 shows the length data for different turns.

Table 1. Length data for different turns

Path Route	Barrier-free turn	Barrier turn ($\theta = 135^\circ$)	Barrier turn ($\theta = 90^\circ$)
Bezier curve length	2.1881	2.4676	2.9348
Ideal linear length	2.4142	2.4142	2.8284

From the above table, it is found that the Bezier curve with barrier-free turns is shorter than the ideal path, while the Bezier curve with barrier turns is obviously longer than the ideal path. Therefore, it is concluded that under a certain number of turns, barrier-free turns are more suitable for actual situation than barrier turns.

Therefore, when we set the smoothing factor parameter value of the algorithm, we should conform to the fact that barrier-free turns are better than barrier turns. The definition of the smoothing factor w is divided into two parts as shown in the following formula:

$$w_1 = \begin{cases} 1, & \theta = 0^\circ \\ 0.4, & \theta = 135^\circ \\ 0.3, & \theta = 90^\circ \end{cases} \quad (5)$$

$$w_2 = \begin{cases} 1, & \text{straight line} \\ 0.5, & \text{swerve} \end{cases} \quad (6)$$

w_1 is a smoothing factor in the case of Barrier turn, and w_2 is a smoothing factor in the case of Barrier-free turn. According to the actual turning curve, we set the smoothing factor as the above three levels. The smoothing factor is combined with the pseudo-random scaling rule as shown in the following formula :

$$j = \begin{cases} \arg \max_{j \in allowed_k} \{[\tau(i, j)] * [\eta(i, j)]^\beta \cdot w\}, & \text{if } q \leq q_0 \\ S, & \text{else} \end{cases} \quad (7)$$

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t)}{\sum_{S \in allowed_k} \tau_{iS}^\alpha(t) \cdot \eta_{iS}^\beta(t)}, & j \in allowed_k \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where q_0 is an adjustable parameter, which determines the relative importance between using prior knowledge and exploring new paths; when the random number q is greater than q_0 , the next

node with the largest formula value is directly selected; when q is smaller than q_0 , the selection of the next node is based on the original transition probability. The uncertainty of the pseudo-random mechanism increases the diversity of the ant's solutions in the process of solving, thus increasing the possibility of getting the global optimal solution.

2.3.2 Improvement of Pheromone Update

The ant colony algorithm is a random search method. Like most intelligent optimization algorithms, it seeks the optimal solution from a large number of possible solutions. The construction of the optimal solution is to construct the solution individually by each ant individual, and then select it through information interaction between groups. As an intelligent algorithm, ant colony algorithm is mainly embodied in the update of pheromone and the intelligent relationship between individual ants and groups. However, the existing algorithms do not consider the evolution of ant colony in the iterative process. If the path information of each generation of ant colony is extracted as a feature, the diversity of solutions can be reduced.

According to the concept of statistics on data processing, the relevant characteristic information of a group is extracted. In this paper, the characteristic parameters of each generation of ants are extracted by statistical analysis method, i.e. random value, mean value and maximum difference value of path information of each generation of ants. And it is applied in the updating process of ant colony pheromone. In this paper, on the basis of ACS, EAS and MMAS are used for reference in updating pheromone, and random selection operators are added. The local update rule of pheromone is to reduce path pheromone through negative feedback mechanism, to realize local adjustment of pheromone, and to increase diversity of possible paths explored by ants. The update method of local pheromone is as follows:

$$\tau_{ij} = (1 - \xi) \cdot \tau_{ij} + \xi \cdot \Delta\tau_{ij} \quad (9)$$

where ξ is an adjustable parameter, τ_{ij} is the initial pheromone value, and $\Delta\tau_{ij}$ is a random selection operator. By integrating the random value, mean value and the maximum difference value in the path information of each generation of ants, the following formula is used for calculation:

$$\Delta\tau_{ij} = \begin{cases} \sigma \cdot (L_{rand}^{-1} - L_{wor}^{-1}), & \text{if } \lambda = \lambda_0 \\ \sigma \cdot (L^{-1} - L_{ave}^{-1}), & \text{otherwise} \end{cases} \quad (10)$$

in the formula, λ is the serial number of randomly selected elite ants, σ is the number of elite ants, L_{rand} is the path length corresponding to the selected elite ants in each generation, L_{wor} is the path length corresponding to the worst ants in each generation, L is the path length corresponding to the removal of other selected ants in each generation, and L_{ave} is the average path length of each generation of ants.

For global pheromone update rules we employ traditional pheromone update models. We use pseudo code to represent the entire running process of the improved ant colony algorithm.

algorithm 1 The improve ACO algorithm

Input: K : Set of ants. N : The number of iterations. E : The number of elite ant

Output: $Best$: The length of the shortest path. $BestPath$: The shortest path.

```

1: for  $n = 1 \rightarrow N$  do
2:   for  $k = 1 \rightarrow K$  do
3:     /*a generation of ants began to find their way*/
4:     while The ant does not reach its destination or die do
5:       if  $q \leq q_0$  then
6:         /*a stochastic stage to select the next node*/
7:          $nextnote \leftarrow \arg \max\{\tau_{ij} \cdot \eta_{ij}^\beta \cdot w\}$ 
8:       else
9:          $nextnote \leftarrow p_{ij}$ 
10:      end if
11:       $path \leftarrow nextnote$ 
12:      /*save path node*/
13:    end while
14:  end for
15:  /*update pheromone*/
16:   $\tau_{goal} \leftarrow (\tau_{ij}, \Delta\tau_{ij}^*)$ 
17:  /*Local pheromone updates:combine statistical analysis and random selection*/
18:   $\tau_{ij} \leftarrow \tau_{local} \leftarrow (\tau_{goal}, \Delta\tau_{ij}, E)$ 
19: end for

```

In the above code, we add a smoothing factor w in calculating the transition probability. In the pheromone update process, the global pheromone update is combined with the local pheromone based on statistical analysis.

2.3.3 Evaluation of Algorithm Application

The performance of the algorithm can be evaluated from convergence speed, time complexity or shortest path, etc. As a path planning algorithm, the first consideration is whether the path obtained by the algorithm is the optimal solution and whether the optimal solution is suitable for practical application. Therefore, we set the algorithm path as the first evaluation index.

However, in the process of algorithm implementation, we add the influence factors of actual turning on the path. To evaluate the optimal path of the algorithm, it is necessary to consider the deviation between the ideal path and the turning curve.

3 Experimental Results and Simulation

In order to verify the effectiveness of the improved algorithm in this paper, this section compares the improved algorithm with the algorithms mentioned in this paper, such as EAS, RAS and ACS, which are respectively used in the simulation experiment of vehicle path planning based on grid method. And the improved algorithm and other algorithms are firstly used for path planning in different environments. The simulation experiment is carried out in MATLAB: 2.8GHz processor, 12GB memory and 64-bit system. The global pheromone volatilization factor is 0.1, the local pheromone volatilization factor is 0.08, the heuristic factor is 3, and the pheromone factor is 2.

3.1 Performance Comparison of Algorithms in simple environment

Figures. 5 and 6 are respectively path planning figures and iterative graphs of four algorithms in a U-shaped obstacle environment but with different obstacle positions. Table 2 shows the experimental data in simple environment.

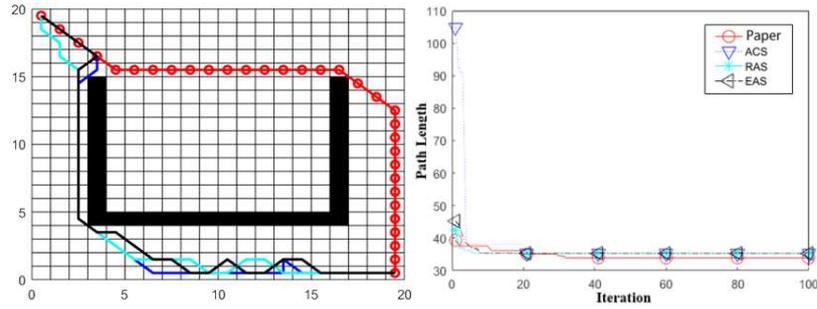


Fig. 5 Path planning and iteration

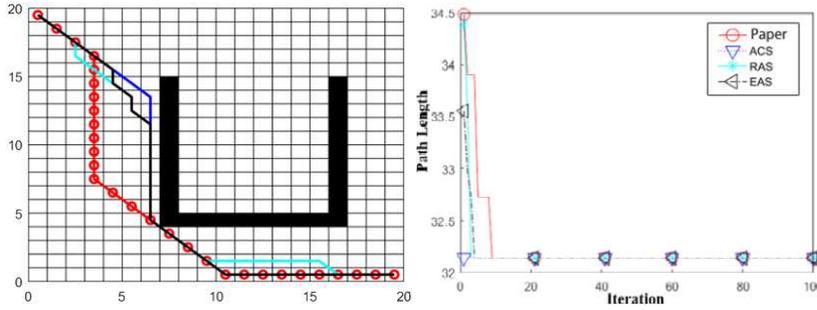


Fig. 6 Path planning and iteration

Table 2. Experimental data in simple environment

Environment Types	Evaluation Parameter	This Paper	ACS	RAS	EAS
Simple_1	Optimal Value	33.90	35.31	34.49	34.49
	Shortest Path Average	33.90	36.17	34.82	34.88
	Standard Deviation of Shortest Path	0	0.97	0.5	0.59
	Average Number of Iterations	25.88	11.02	4.98	6.42
	Average Number of Corners	3.37	7.40	9.67	9.33
	Barrier Corners	0.75	2.34	0.37	0.43
	Simple_2	Optimal Value	32.14	32.14	32.14
Shortest Path Average		32.14	32.14	32.18	32.14
Standard Deviation of Shortest Path		0	0	0.2	0
Average Number of Iterations		4.38	1	3.04	4.32
Average Number of Corners		3.17	3.33	7.93	8.03
Barrier Corners		0	1.04	1.23	1.12

3.1.1 The Result of Performance Comparison in Simple Environment

By combining the path planning chart and data table, it is found that using the improved algorithm can effectively obtain the optimal path. Moreover, it can be clearly found from the table that the improved algorithm makes fewer turns in the path planning, and other algorithms make too many invalid turns on the path that could travel straight, which makes the moving path more complicated. This shows the function of smoothing factor, which is studied in detail in the next section. According to the evaluation reference, the influence caused by the shortest path and obstacle angle of the algorithm is comprehensively evaluated, and the improved algorithm is better than other algorithms in this environment.

3.2 Performance Comparison of Algorithms in Medium Environment

The following two figures are respectively the path planning and iteration of the four algorithms in medium environment. Table 3 gives experimental data in medium environment.

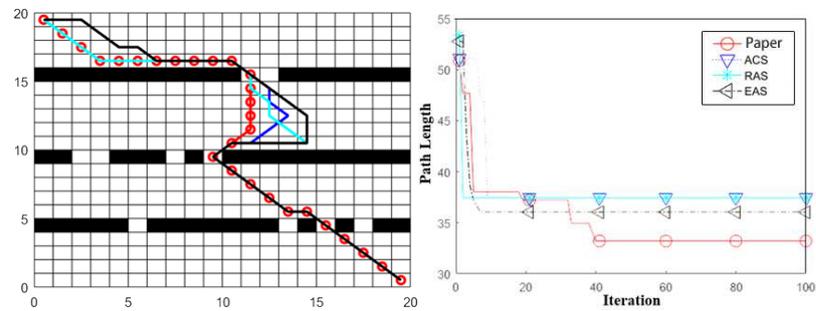


Fig. 7 Path planning and iteration

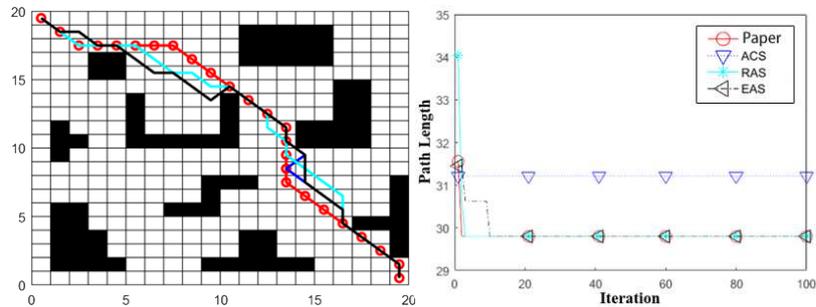


Fig. 8 Path planning and iteration

Table 3. Experimental data in medium environment

Environment Types	Evaluation Parameter	This Paper	ACS	RAS	EAS
Medium_1	Optimal Value	33.21	34.04	33.21	33.21
	Shortest Path Average	33.21	36.29	37.07	36.63
	Standard Deviation of Shortest Path	0	2.09	2.16	1.77
	Average Number of Iterations	36.83	14.73	6.53	7.57
	Average Number	7	10.33	10.90	11.10

		of Corners			
Medium_2	Barrier Corners	3.27	3.54	3.24	3.65
	Optimal Value	29.80	29.80	29.80	29.80
	Shortest Path Average	29.80	31.14	29.96	29.98
	Standard				
	Deviation of Shortest Path	0	0.59	0.34	0.33
	Average Number of Iterations	6.8	11.57	3.53	4.8
	Average Number of Corners	5.03	10.73	9.83	9.43
	Barrier Corners	0.32	3.78	2.54	4.12

3.2.1 The Result of Performance Comparison in Medium Environment

Figures 7 and 8 show the actual effects of the four algorithms in medium-sized environment. We can find from the iteration in Fig. 7 that no optimal solution has been found in other algorithms except the improved algorithm. From the number of corners in the data table, the improved algorithm is obviously better than other algorithms, whether it is the number of corners or the number of corners under obstacles. Moreover, the standard deviation of the shortest path also shows that the improved algorithm is better than other algorithms in the aspect of stability.

3.3 Performance Comparison of Algorithms in Complex Environment

The following figures show the path planning and iteration of the four algorithms in complex environment. Table 4 gives experimental data in complex environment.

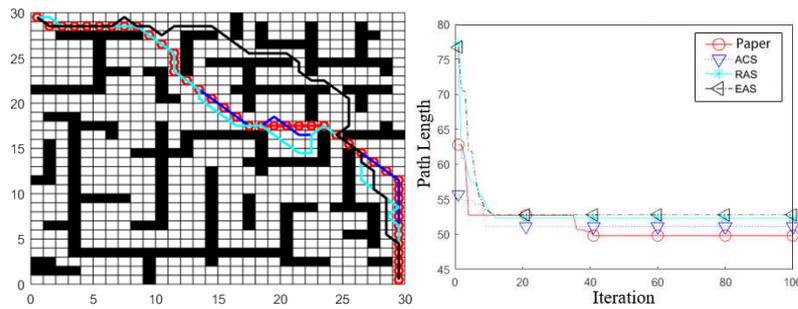


Fig. 9 Path planning and iteration

Table 4. Experimental data in complex environment

Environment Types	Evaluation Parameter	This Paper	ACS	RAS	EAS
Complex	Optimal Value	48.63	49.46	51.46	51.46
	Shortest Path Average	48.63	50.29	53.21	51.46
	Standard				
	Deviation of Shortest Path	0	1.11	0.83	1.3

Average Number of Iterations	34.95	16.8	8.55	10.05
Average Number of Corners	7	11.5	18.1	21.4
Barrier Corners	3	3.56	6.21	6.84

3.3.1 The Result of Performance Comparison in Medium Environment

Figure 9 and Table 4 are comparisons of four algorithms in complex environment. It is obvious from the iteration that the improved algorithm can find the optimal solution. By judging from the average number of corners and the number of corners with obstacles, the number of corners of the improved algorithm is far smaller than that of other algorithms, which shows that the improved algorithm has better smoothness. Although the improved algorithm has some losses in convergence speed, but the decrease in convergence speed is allowed on the premise that the optimal solution can be found.

3.4 The Discussion of Performance Comparison in Different Environment

The performance of the integrated algorithm in different complexity environments shows that the improved algorithm has better performance. From the standard deviation of the shortest path, it is found that the improved algorithm has better stability, accuracy and optimization ability in the process of searching for the path, which effectively improves the unstable results of probabilistic algorithm. In addition, because the smoothing factor is added, the algorithm is obviously better than other algorithms in the ability to avoid turning, which makes the shortest path of the algorithm smoother. In practical application, the improved algorithm has more application potential than other algorithms. The smoothing factor is further discussed in the next section.

3.5 Experimental Verification of Smoothing Factor

The experiments in the previous section show the performance of the improved algorithm in different experimental environments. In this section, the role of smoothing factors in the algorithm is further studied. We apply the improved algorithm with or without smoothing factor to medium and complex environment to judge the effect of smoothing factor. The shortest path, diversity and the number of corners of the algorithm are compared to reach a conclusion, where in diversity is the ability to construct solutions in the iterative process of the algorithm. With the increase of diversity, the probability of finding the optimal solution is enhanced, which can effectively avoid the occurrence of local optimal problems.

3.5.1 Comparison of Algorithms in Medium Environment

The following two figures are the path planning with two algorithms in a medium-sized environment. Figure (a) is an improved algorithm without adding a smoothing factor, and Fig. (b) is an improved algorithm with adding a smoothing factor. Table 5 shows the resulting data.

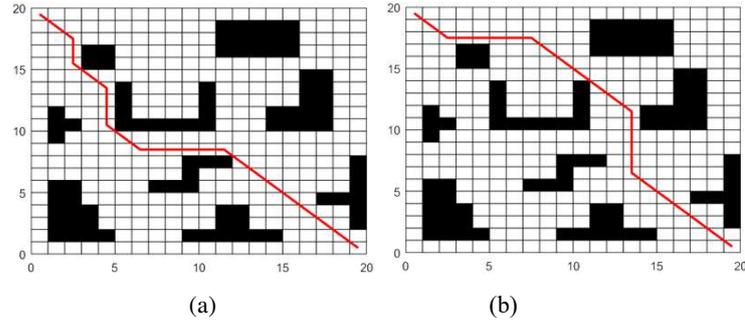


Fig. 10 Path planning

Table 5. Algorithm resulting data

Evaluation Parameter	Improved Algorithm (no smoothing)	Improved Algorithm (smoothing)
Optimal Value	29.799	29.799
Shortest Path Average	29.799	29.799
Average Number of Iterations	23.4	25.4
Algorithm Diversity	3.61134	6.22551
Average Number of Corners	5.56	4.34
Barrier Corners	3.32	0.12

From the above chart, it is found that the smoothing factor has a great influence on the performance of the algorithm. In terms of the average number of corners, both algorithms can effectively find the shortest path, but the average number of corners of the algorithm without w is significantly larger than that of the improved algorithm with w . Through further analysis, we have previously concluded that the number of barrier corners will greatly affect the actual path, so for the smaller number, the algorithm has better ability to obtain the optimal solution under actual constraints.

3.5.2 Comparison of Algorithms in Complex Environment

The following two figures are path planning of two algorithms in a complex environment. Figure (a) is an improved algorithm without adding a smoothing factor, and Fig. (b) is an improved algorithm with adding a smoothing factor. Table 6 shows the resulting data.

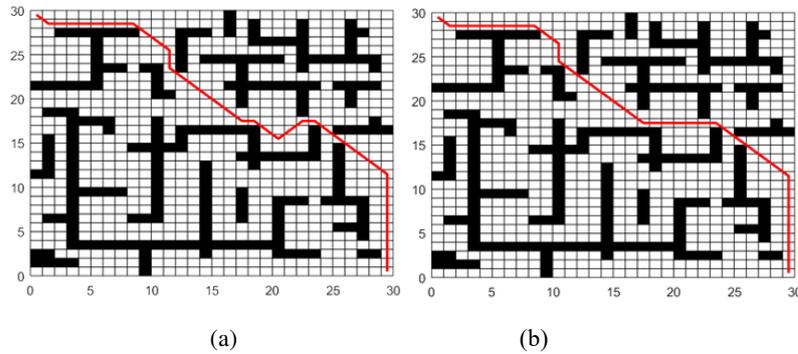


Fig. 11 Path planning

Table 6. Algorithm resulting data

Evaluation Parameter	Improved Algorithm (no smoothing)	Improved Algorithm (smoothing)
----------------------	-----------------------------------	--------------------------------

	smoothing)	
Optimal Value	50.2843	48.6274
Shortest Path Average	50.51862	48.6274
Average Number of Iterations	39.5	40.9
Algorithm Diversity	2.75153	3.22607
Average Number of Corners	10.84	7.53
Barrier Corners	4.65	2.47

From the above chart, it is found that under the same conditions, the improved algorithm with w added can find a shorter path than the algorithm without w , and it is obviously better than the improved algorithm without w in terms of the average number of corners and algorithm diversity.

3.6 Discuss the Influence of Smoothing Factor in the Algorithm

From the experimental comparison of the smoothing factor in the above two environments, we have proved that the smoothing factor have a significant effect on improving the performance of all aspects of the algorithm. Although the algorithm has a slight loss in convergence speed, it is negligible compared with the greater improvement in other performance of the algorithm.

4 Conclusions

In this paper we have analyzed the characteristics of ant colony algorithm in path planning and the positive feedback process of ant information. We have examined the smoothness of the path and discussed the actual influence of obstacles on the turning path. The added smoothing factors have been classified to make the algorithm more suitable for the actual environment. By combining the penalty mechanism with pseudo-random selection rules, the selection probability of straight path have been improved, thus improving the convergence speed, smoothness and diversity of the algorithm. In the process of local pheromone update, we have extracted the worst and average ant information of each generation of ants through statistical analysis and randomly select them. The convergence speed of the coordination algorithm and the ability to get the optimal solution make the algorithm self-adaptive to a certain extent because each generation of ant information has to undergo the same processing. Through the experimental simulation in different complexity environments, it is proved that the improved algorithm can give the optimal solution accurately and stably compared with other algorithms in different environments, and the algorithm has excellent performance in smoothness.

In the convergence aspect of the algorithm in this paper, the convergence speed decreases because of improving the diversity of the algorithm. How to balance the convergence and diversity of the algorithm is the next goal to be solved.

Abbreviations

AGV: Automatic guided vehicle; VRP: vehicle routing problem; EAS: elite ant system; MMAS: the MAX-MIN ant system; ACS: the ant colony system

Authors' contributions

WZ carried out the improved ant colony algorithm studies, simulated experiment and drafted the manuscript. JL participated in the design of the study and helped to draft the manuscript. All authors read and approved the final manuscript.

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests

There are no financial and non-financial competing interests.

Author details

1. School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, 61000, China

5 References

- [1] S. MikaelPersson and I. Sharf, "Sampling-based A* algorithm for robot path-planning," *The International Journal of Robotics Research*, Vol. 33, pp 1683-1708, 2014.
- [2] Y. Xin, H. Liang, M. Du, and T. Mei, "An improved A* algorithm for searching infinite neighborhoods," *Robot*, Vol. 36, pp 627–633, 2014.
- [3] W. Huiying and X. Jianmin, "Genetic algorithm for route optimization of vehicle navigation in logistics distribution," *Journal of South China University of Technology*, Vol. 37, No. 2, pp 82-86, 2009.
- [4] X. Junfeng and Z. Jiabao, "An improved genetic algorithm for path-planning of unmanned surface vehicle" *Sensors*, Vol. 19, No. 11, p 23, 2019.
- [5] H. Lee, H. Shin and J. Chae, "Path planning for mobile agents using a genetic algorithm with a direction guided factor," *Electronics*, Vol. 7, No. 10, p 22, 2018.
- [6] N. Herojit Singh and K. Thongam, "Mobile robot navigation using fuzzy-GA approaches along with three path concept," *Transactions of Electrical Engineering*, Vol. 43, Issue. 2, pp 277-294, 2018.
- [7] M. López-Ibáñez, T. M. Stützle, and M. Dorigo, "Ant colony optimization: A component-wise overview," IRIDIA, Univ. Libre de Bruxelles, Brussels, Belgium, Tech. Rep. TR/IRIDIA/2015-006, 2015.
- [8] H.Min and D.Ping, "An improved ant colony algorithm and its application in vehicle routing problem," *Mathematical Problems in Engineering*, Vol 2013, AID 785383, p.9, 2013.
- [9] Y. Longwang and Chen Hanning, "Unmanned vehicle path planning using a novel ant colony algorithm," *EURASIP Journal on Wireless Communications and Networking*, Vol 2019, Issue. 1, p 136, 2019.
- [10] Q. Xiaohui and Z. Xiaopeng, "Optimal individualized multimedia tourism route planning based on ant colony algorithms and large data hidden mining," *Multimedia Tools and Applications*, Vol 78, Issue 15, pp 22099-22108, 2019.
- [11] X. Junfeng and L. Shixin, "Application of improved particle swarm optimization for navigation of unmanned surface vehicles," *Sensors*, Vol. 19, Issue. 14, p 3096, 2019.
- [12] W. Haiyan and Z. Zhiyu, "A heuristic elastic particle swarm optimization algorithm for robot path planning," *Information*, Vol. 10, Issue. 3, p 99, 2019.
- [13] T. ThoaMac, C. Copot and D. TrungTran, "A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization," *Applied Soft Computing*, Vol. 59, p. 68-76, 2017.
- [14] Ravi Kum and Sukesh Bondada, "An optimized path planning for the mobile robot using potential field method

- and PSO algorithm,” *Soft Computing for Problem Solving*, Vol. 817, pp 139-150, 2018.
- [15] H.Chen and F.Jiyou, “Smooth path planning method based on dynamic feedback A* ant colony algorithm,” *Transactions of the Chinese Society for Agricultural Machinery*, Vol. 48, No. 4, pp. 34-40, 2017.
- [16] S.Rui and L.Yuanchang, “Smoothed A* algorithm for practical unmanned surface vehicle path planning,” *Applied Ocean Research*, Vol. 83, pp.9-20, 2019.
- [17] S. Baoye and W. Zidong, “A new approach to smooth global path planning of mobile robots with kinematic constraints,” *International Journal of Machine Learning and Cybernetics*, Vol. 10, Issue. 1, pp. 107-119, 2019.
- [18] M. Dorigo, V. Maniezzo, and A. Coloni, “Ant system: Optimization by a colony of cooperating agents,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, Vol. 26, No. 1, pp. 29-41, 1996.
- [19] M. Dorigo and L. M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” *IEEE Trans. Evol. Comput.*, Vol. 1, No. 1, pp. 53-66, 1997.
- [20] T. Stützle and H. H. Hoos, “MAX_MIN ant system,” *Future Generat. Comput. Syst.*, Vol. 16, No. 8, pp. 889-914, 2000.

Figures

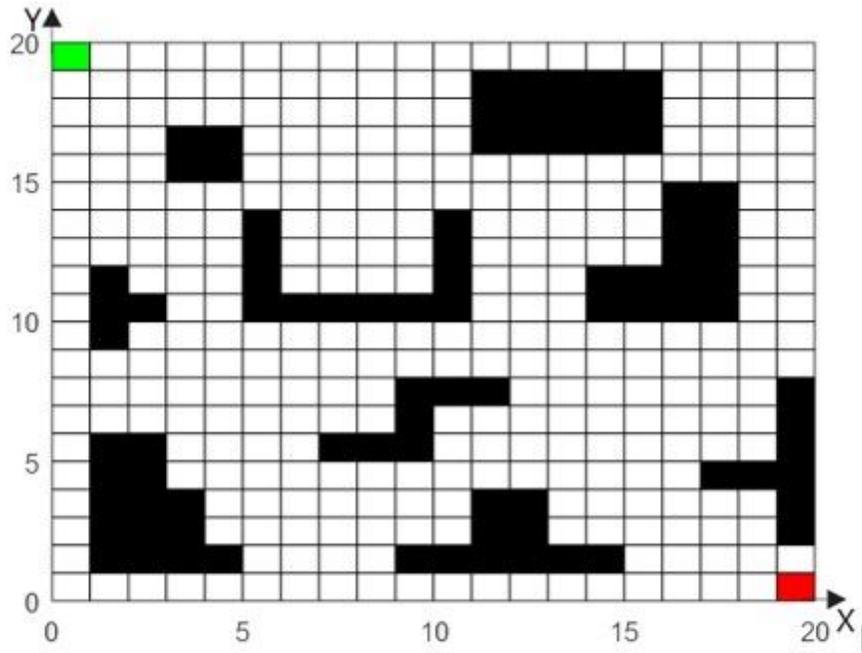


Fig. 1 Grid map

Grid method to establish environment model

Figure 1

Grid method to establish environment model

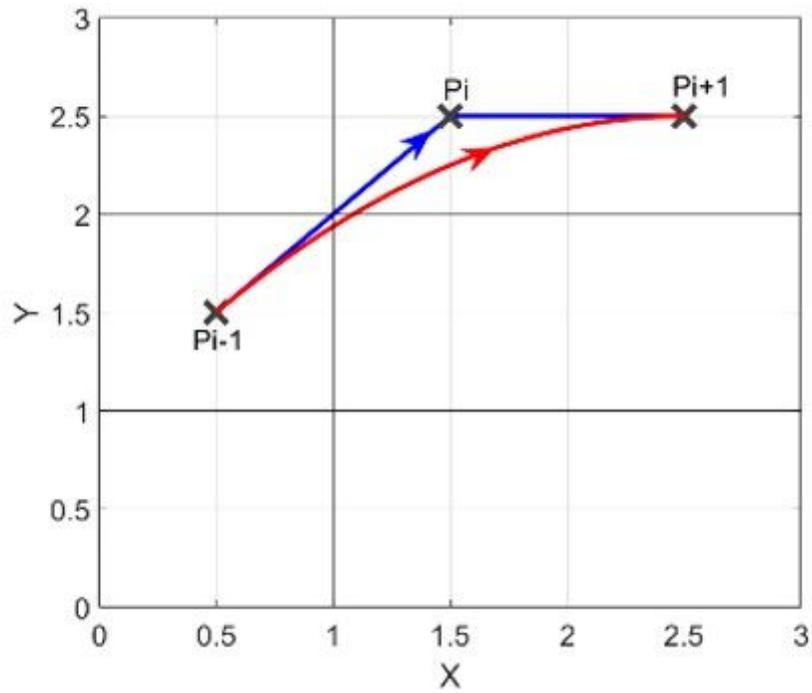


Fig. 2 Barrier-free turn

Figure 2

When there is no obstacle, the turning process of the vehicle. The blue line is the ideal route and the red line is the actual route.

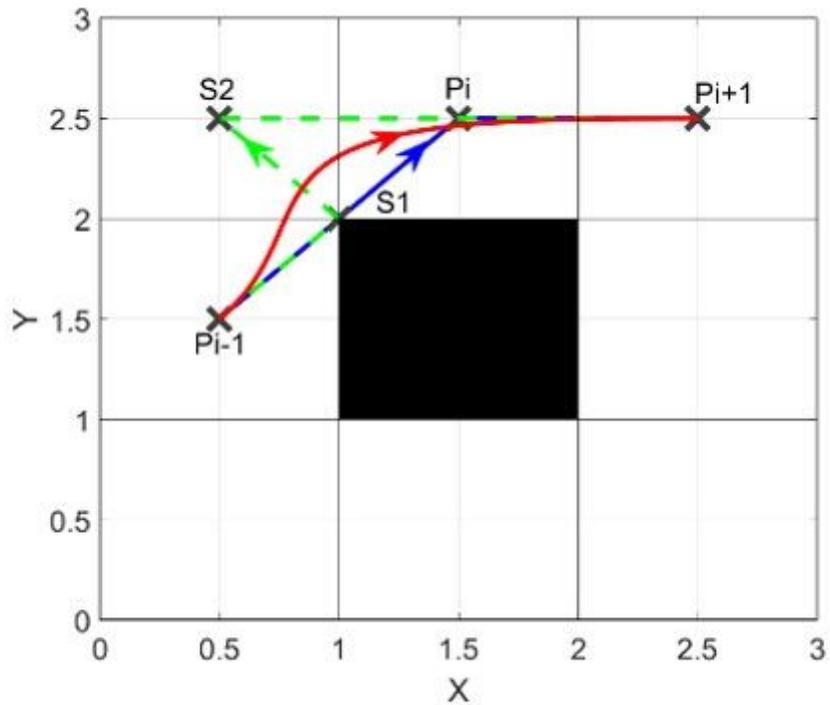


Fig. 3 Barrier turn ($\theta = 135^\circ$)

Figure 3

When there is an obstacle, the vehicle completes the turning process ($\theta=135^\circ$). The blue line is the ideal route and the red line is the actual route.

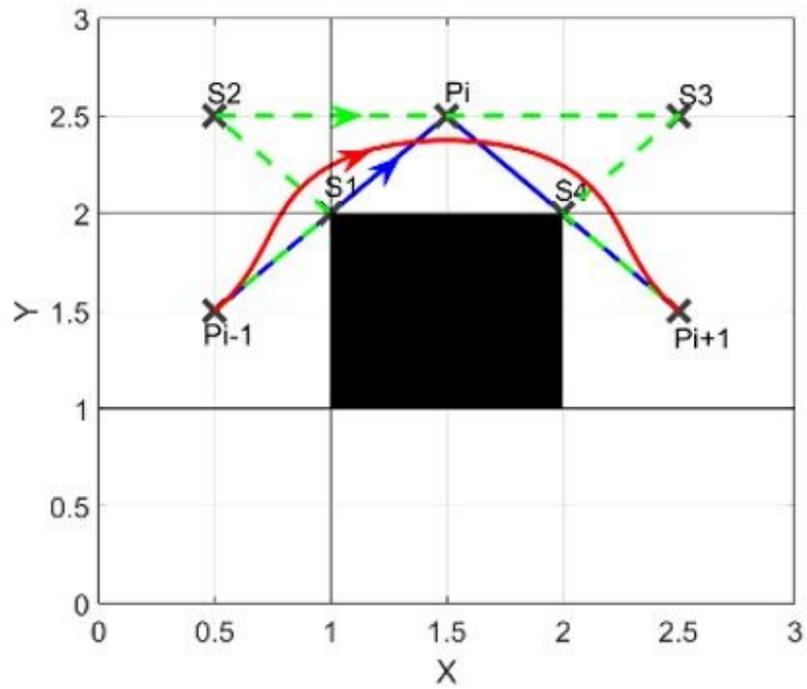


Fig. 4 Barrier turn ($\theta = 90^\circ$)

Figure 4

When there is an obstacle, the vehicle completes the turning process ($\theta=90^\circ$). The blue line is the ideal route and the red line is the actual route.

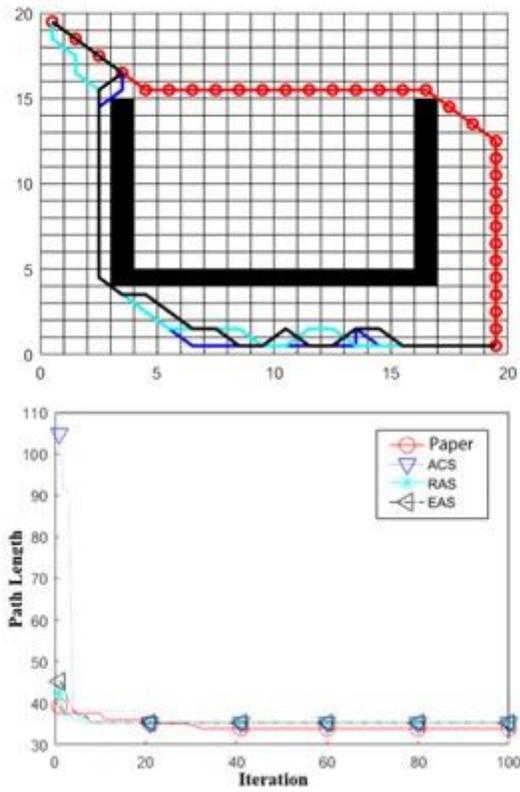


Figure 5

Path planning and iteration Path planning results of four algorithms in simple environment.

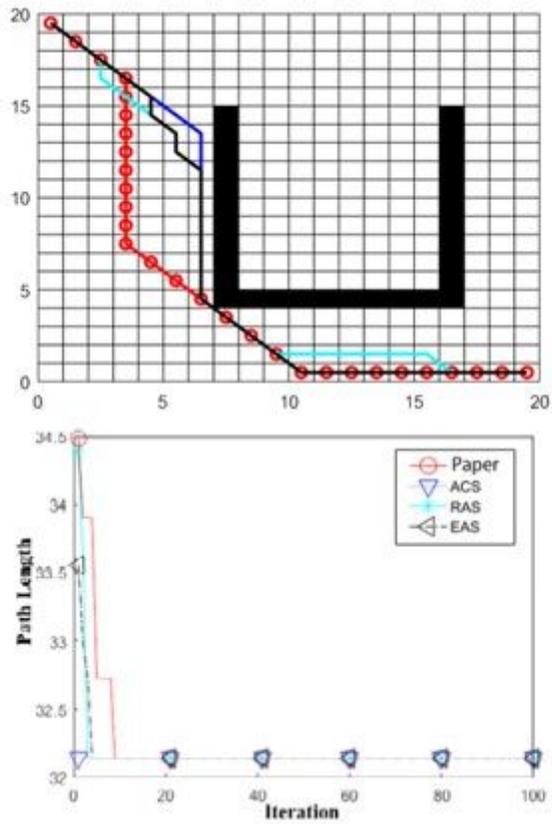


Fig. 6 Path planning and iteration

Figure 6

Path planning results of four algorithms in simple environment

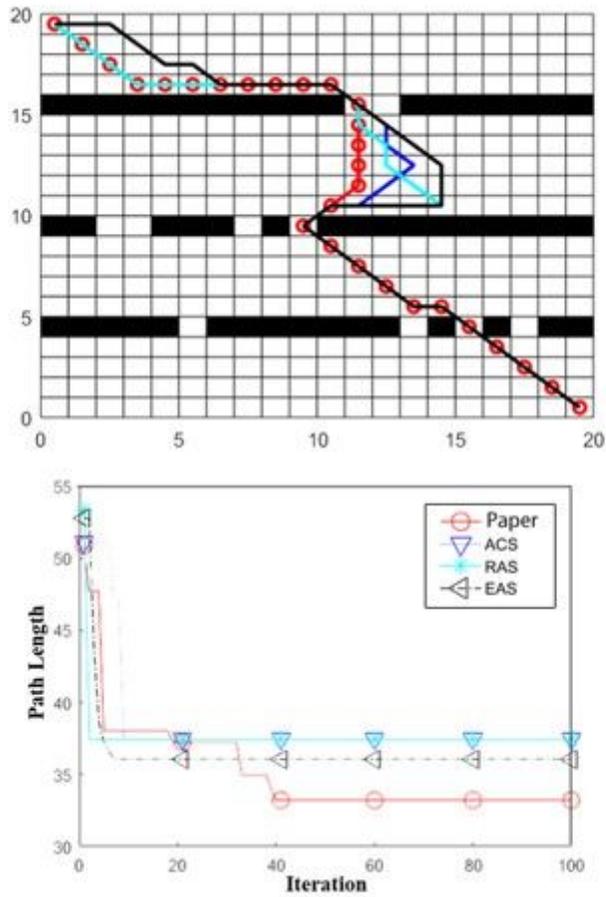


Fig. 7 Path planning and iteration

Figure 7

Path planning results of four algorithms in medium environment

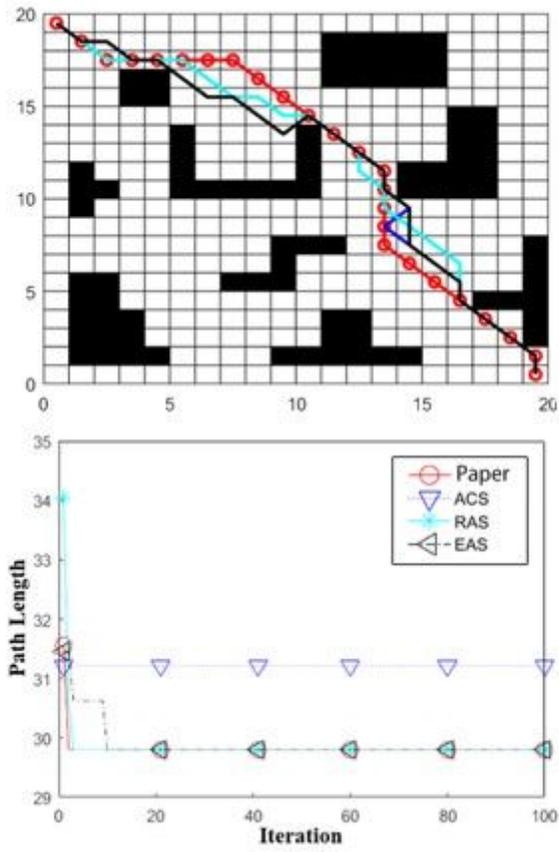


Fig. 8 Path planning and iteration

Figure 8

Path planning results of four algorithms in medium environment

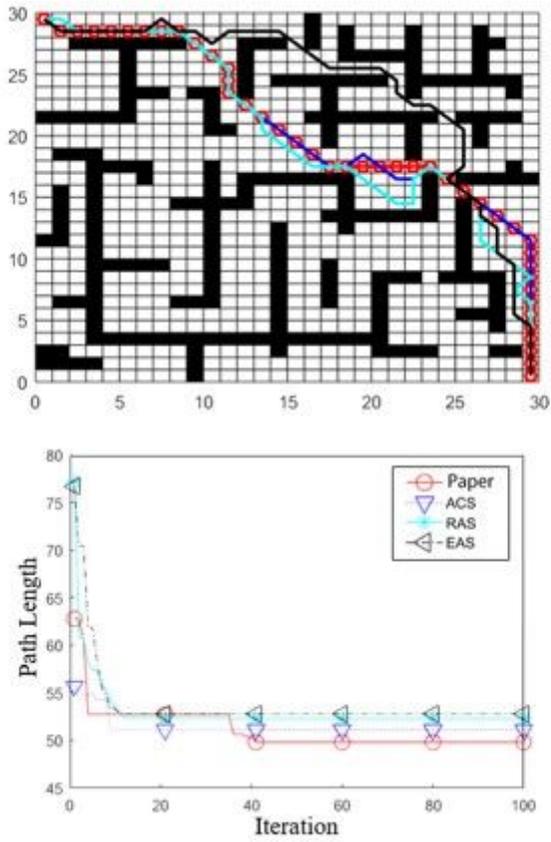
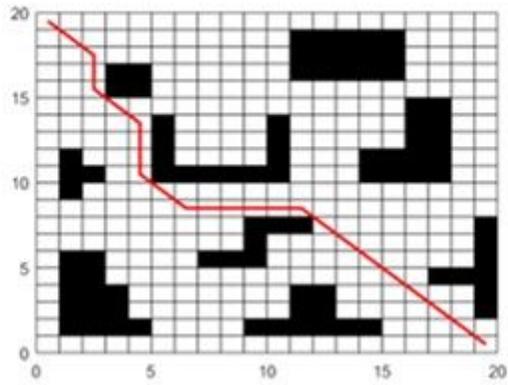


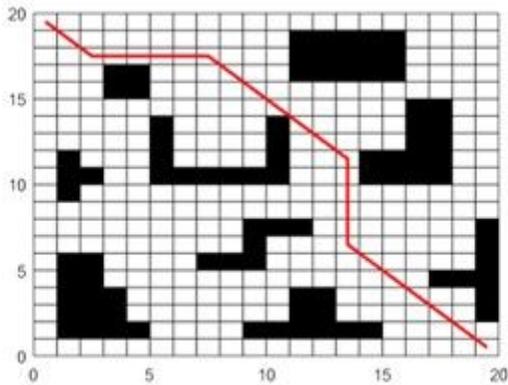
Fig. 9 Path planning and iteration

Figure 9

Path planning results of four algorithms in complex environment



(a)

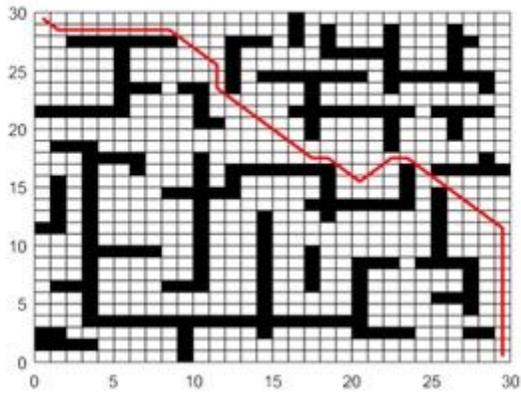


(b)

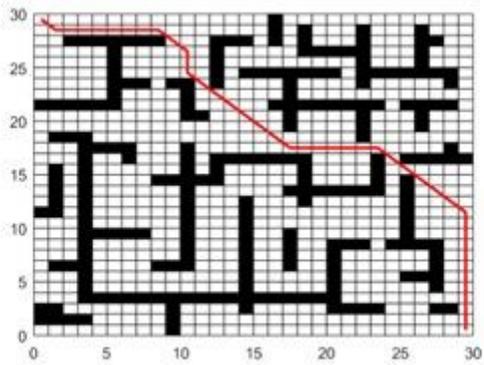
Fig. 10 Path planning

Figure 10

Path planning (a) the result of improved ant colony algorithm without a smoothing factor in medium environment. (b) the result of improved ant colony algorithm with a smoothing factor in medium environment.



(a)



(b)

Fig. 11 Path planning

Figure 11

Path planning (a) the result of improved ant colony algorithm without a smoothing factor in complex environment. (b) The result of improved ant colony algorithm with a smoothing factor in complex environment.