

RESEARCH

Open Access

Unmanned vehicle path planning using a novel ant colony algorithm



Longwang Yue¹ and Hanning Chen^{2*} 

Abstract

The ant colony optimization algorithm is an effective way to solve the problem of unmanned vehicle path planning. First, establish the environment model of the unmanned vehicle path planning, process and describe the environmental information, and finally realize the division of the problem space. Next, the biomimetic behavior of the ant colony algorithm is described. The ant colony algorithm has been improved by adding a penalty strategy. This penalty strategy can enhance the utilization of resources and guide the ants to explore other unknown areas by using the worse value in the search history to enhance the volatility of the pheromone.

Keywords: Path planning, Ant colony algorithm, Grid method, Penalty strategy

1 Introduction

Unmanned vehicle path planning explores a feasible path in the known or unknown environment by sensing the surrounding environment. Path planning problem not only simply expresses the search of a route from the start point to the end point but also presents an optimal path among all reachable paths [1]. When generating the best path, there are several related issues to consider, such as security, obstacles, and computation time. Due to its importance, many researchers have conducted a large path planning algorithms. In the literature [2], the paper proposes a drone route planning based on particle swarm optimization algorithm. The corresponding mutation and fine adjustment of the inactive particles are carried out to ensure the particle group has strong vitality in the evolution process. In the literature [3], the author obtains the best path for each UAV in parallel by genetic algorithm. According to the dimensions of path planning, genetic coding, etc., new genetic operators are introduced to select appropriate chromosome pairs for crossover operations. In the literature [4], Fan et al. proposed a kind of manual market method of infeasible path correction strategy for the differential evolution algorithm mutation factor to improve the effectiveness of the algorithm to find the shortest path.

The ant colony optimization algorithm, which is derived from the study of ant group behavior, simulates a bionic intelligent optimization algorithm based on the cooperation between ant colonies. When ants are foraging, they will leave exogenous hormone, and others can recognize the intensity of pheromone. And ants tend to move toward higher pheromone concentrations. That can be said as a kind of positive feedback phenomenon of the ant group during the foraging process [5]. It is because of this positive feedback mechanism that the ant colony can search for food more quickly. This algorithm has strong global search ability, can perform parallel and distributed computing, and has fast convergence speed and strong adaptability [6], so it has been widely used in solving path planning problems. In the literature [7], the paper proposes an improved ant colony algorithm. The article mainly improves the positional distribution of the initial population and increases the adaptive evaporation factor and simulated annealing. It is found through experiments that the algorithm can effectively reduce the problem of search time. In the literature [8], the author can avoid the blindness of initial planning by adjusting the transition probability based on the classical ant colony algorithm and introducing relevant strategies to solve the deadlock problem. The simulation experiment proves that the algorithm is superior to the classical ant colony algorithm, which can effectively guide the mobile robot to avoid dynamic obstacles in the environment, obtain the optimal or suboptimal path without collision,

* Correspondence: perfect_chn@hotmail.com

²School of Computer Science and Technology, Tianjin Polytechnic University, Tianjin, China

Full list of author information is available at the end of the article

and better adapt to the changes of the environment. In the literature [9], based on the basic ant colony algorithm, the author improves the heuristic information and pheromone update strategy by introducing the ant colony regression strategy to improve the adaptability, convergence speed and optimization ability of the algorithm.

2 Environmental model

The currently constructed environment model is generally simplified to a two-dimensional map. The path planning environment model of the unmanned vehicle of this paper will be constructed by the grid method proposed by W.E. Howden. According to the grid method, first divide the simulation environment into several identical grids according to the scale and segmentation requirements, then set the environmental parameters for each grid, next set the conditions of each grid obstacle according to the actual or hypothetical environment. The data structure obtained by the grid method is actually a two-dimensional table, so the map is relatively easy to create and maintain in the computer. The map is easy to transfer into the coordinate system, so that the position and feasible area of the obstacle can be displayed more intuitively, and the position information is markedly marked by the row and column [10].

The grid method decomposes the unmanned work space into a series of binary information grid cells. In a static environment, it is assumed that the size and position of the obstacle are known, and the two-dimensional workspace model of the unmanned vehicle is D , which is evenly divided using the grid. Since the actual shape of the obstacle is irregular, when it is less than one grid, it is added as a grid. Considering that the two-dimensional map can be of any shape, the entire graph is set to a rectangle or a square, complemented by a barrier grid at the boundary of the original graph. According to the meshed area and the area where the obstacle is located, as shown in Fig. 1, it can be divided into a feasible grid (white grid) and an infeasible grid (black grid), where green is the starting point and red is the ending point.

Using the numbering method, the divided grids are numbered in order from left to right and top to bottom. Each grid has its corresponding serial number and coordinates, that is, the i -th row, the grid of the j -th column is denoted as $D(i, j)$, and the corresponding serial number is k . The relationship between the grid number k and the coordinates (x_i, y_j) is as shown in Eqs. (1) and (2).

$$x_i = [(k-1) \bmod N_h] + 1 \tag{1}$$

$$y_j = \text{int}[(k-1)/N_h] + 1 \tag{2}$$

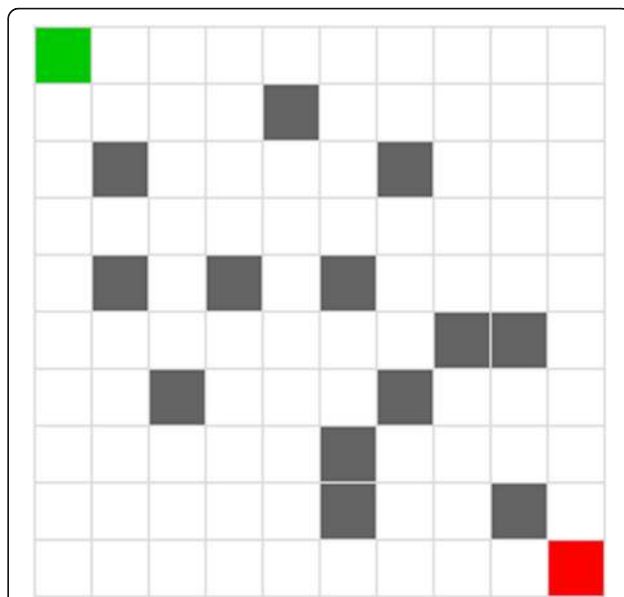


Fig. 1 Grid method to establish environment model. Figure illustrates that the unmanned work space is decomposed into a series of binary information grid cells

where N_h is the number of rasters per row, int is the remainder of the remainder, and mod is the remainder calculation.

During the movement of the robot, the default movement direction of the robot is set to eight, as shown in Fig. 2 below. Since each step of the robot is from the center of one grid to the center of another adjacent grid, the motion step of the robot is R or $\sqrt{2}R$.

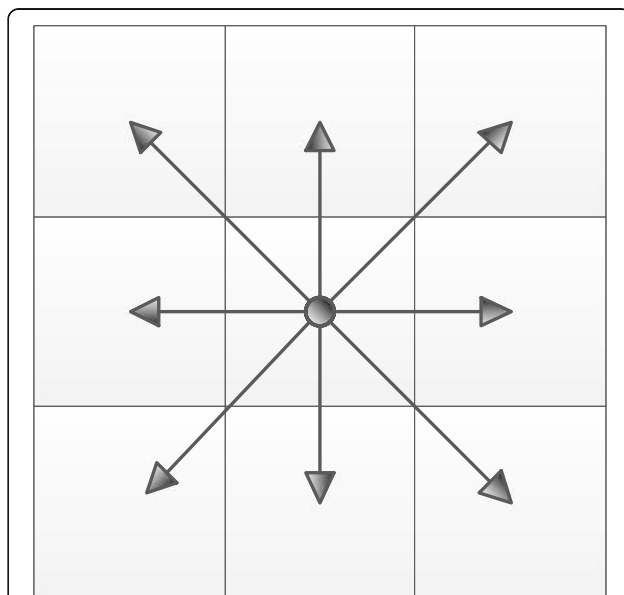


Fig. 2 Unmanned vehicle movement direction. Figure shows the default movement direction of the vehicle

3 Ant colony algorithm

3.1 Ant colony optimization algorithm

The ant colony algorithm (ACO) is used to solve the path planning problem, which can be divided into two parts: path construction and pheromone update [11]. The artificial ant colony algorithm and the real ant foraging process are moving from one position to the next, and the position change is realized by the probability selection strategy. A pheromone release and volatilization mechanism is also set in the process of artificial ant movement, but the artificial ant colony algorithm adds some characteristics that are different from the real ant colony foraging behavior:

- (1) In the algorithm, the artificial ant colony is placed in a discrete space and does not have any association with time. The next move position is determined by the selection strategy. The update of the pheromone is independent of time and is only related to the path and update method.
- (2) The artificial ant colony has a path memory function, which records the walking route before the ant [12].
- (3) The artificial ant colony is not completely blind. It has heuristic information about the problem space.
- (4) In order to improve the search efficiency of the algorithm. Artificial ant colonies have added new features such as local optimization, prior knowledge, predictive future and tabu search.

In the following formula (3), $P_{ij}^k(t)$ is the calculation method of ant position transfer, which indicates the transition probability of ant k moving from position i to position j at time t :

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}(t)]^\beta}{\sum_{k \in \text{allowed}_k} [\tau_{ij}(t)]^\alpha \times [\eta_{ij}(t)]^\beta} & \text{if } j \in \text{allowed}_k \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $\tau_{ij}(t)$ represents the pheromone intensity of the current position to the target point at time t , and $\eta_{ij}(t)$ is the visibility 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 represent the weight of the pheromone intensity and the ant visibility, respectively. $\text{Allowed}_k = \{N - \text{tabu}_k\}$ indicates the city collection that ant k can currently select; tabu_k is the taboo list of ant k , indicating the city collection where ants are not selectable and N is the total number of cities from

the current location to the next location. And tabu_k is constantly changing as the position of the ant changes.

In order to simulate the change of pheromone in the process of ant foraging, the update of artificial ant colony pheromone also considers the two processes of volatilization and release of pheromone [13]. In the real world, as time goes by, the pheromone on the path will gradually evaporate. And its volatilization is helpful for ants to explore other areas to find better paths, so it does not converge too quickly to a local optimal solution; in the path of ants to explore food, the corresponding pheromones are also released, so that the ants can communicate with each other and have a certain guiding effect on other ants who are looking for food. The initial setting of the pheromone is neither too large nor too small; too large will make its guiding effect lower, while too small will make the ant group converge too quickly to a local optimal path [14]. Thus, the pheromone update formula (4) is expressed as follows:

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (4)$$

where m is the number of ants, $0 < \rho < 1$ is the evaporation rate of the pheromone and is usually set to 0.5 in the ACO. Therefore, $(1 - \rho)\tau_{ij}(t)$ represents the residual amount of a path pheromone found by the ant after volatilization, and $\Delta\tau_{ij}^k$ is the pheromone left by the k th ant in the path i to j . As shown in Eq. (5):

$$\Delta\tau_{ij}^k = \begin{cases} (C_k)^{-1} & \text{shortest path} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where C_k is the total path length obtained after the k th ant walks the complete path. Here, as described above, the optimal path distance is obtained.

Take the TSP problem as an example. The process of algorithm design is as follows:

- Step 1: Initialize the relevant parameters, including ant colony size, pheromone factor, heuristic function factor, pheromone volatilization factor, pheromone constant and maximum number of iterations
- Step 2: And reading the data into the program and pre-processing: for example, converting the city's coordinate information into a distance matrix between cities
- Step 3: Randomly place the ants at different starting points and calculate the next visiting city for each ant until there are ants accessing all the cities
- Step 4: Calculate the path length of each ant, record the optimal solution of the current iteration number, and update the pheromone concentration on the path

Step 5: Determine whether the maximum number of iterations is reached. If not, return to step 2 or end the program.

Step 6: Output the result, and output relevant indicators in the optimization process, such as running time, convergence iteration number, etc., as needed.

3.2 Ant colony algorithm with punitive measures

In the research of ant colony algorithm, it is found that there must be an optimal solution and the worst solution for the whole ant colony search results. Previous artificial ant colony algorithm only based on the optimal solution of ant colony search behavior, to generate positive feedback behavior of the whole group. However, the total amount of resources invested in the path exploration process is constant. Therefore, this paper hopes to use the search results of the poor path to enhance the volatilization degree of the pheromone on the poor path and reduce the number of traversal times. At this time, the concentration of the pheromone in the unexplored area will be significantly larger than the value confirmed as the worse path, which will give the ant a greater chance to explore the unknown area, because there may be a better solution in the unknown field. First, all the paths found by the ants are sorted, and then, the path is sorted to the last ω ants to punish the pheromone volatilization. The ω ants are sorted according to the length of the path, and the penalty condition is weighted according to different sorting levels, that is, the higher the path sorting, the higher the level, and the greater the penalty for the ant search path. Based on this idea, this paper attempts to design the pheromone penalty model of the ant colony algorithm, as shown in formula (8):

$$C = \rho + \frac{\lambda \sum_{\omega=0}^{k-\omega} (k-\omega)}{D} \tag{6}$$

where ρ represents the pheromone volatilization rate and D is the distance from the point to the target point. λ represents 1/10 of the distance D to ensure that the entire value is not too large. ω is the order of the poor path of the ant search, and k is a fixed value equal to ω . In a specific application, if the optimal path and the worst path pass through this point, no penalty is imposed.

In fact, this has an effect on the value of the ant's probability selection formula. That is, the value of the pheromone on the poor path is reduced, and the probability of being traversed again is lower. At this time, there is a greater probability of exploring the unknown domain. In the aspect of pheromone update, the global pheromone update method is adopted, because the local update method is more likely to cause the ant to fall into

Table 1 Partial issues in the TSPLIB data set

Question	Description	Optimal solution
ulyeese	Odyssey of Ulysses(Groetschel/Padberg)	74
Att48	48 capitals of the US (Padberg/Rinaldi)	33,522
Eil76	76-city problem (Christofides/Eilon)	538

the local optimal solution [15]. That is, after all the ants of each generation have completed the search, the pheromone is updated according to the result of the optimization and the pheromone update model described above. In the search mode of the AS algorithm, the premature convergence mechanism is avoided, so that the algorithm obtains better performance. The total amount of pheromone on the path is also controlled within a certain range ([MIN, MAX]), which can well avoid the ant to converge too quickly to the local optimal path, so that ants can search for a wider range of unknown areas. In the initial search phase of the ant, in order to improve the algorithm search ability, the initial value of the pheromone is set to the upper limit. Its pheromone is updated according to the following rules:

$$\tau_{ij}(t) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best} \tag{7}$$

where $\Delta\tau_{ij}^{best}$ is the update of the optimal path pheromone, and C_k is the optimal path distance, as follows:

$$\Delta\tau_{ij}^{best} = \begin{cases} (C_k)^{-1} & \text{shortest path} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Although the historical optimal solution is retained after the ant colony search ends, when the path pheromone is updated, the pheromone matrix is updated on the optimal path of the current generation [16]. It will make better use of the positive feedback mechanism generated by the optimal path and give more opportunities to explore the unknown. As a result, the path exploration of the whole algorithm becomes relatively more diverse, avoiding premature convergence to the local optimal solution and reducing the exploration of the path that has been confirmed as a worse path, further improving the performance of the ant colony search algorithm.

Table 2 Algorithm parameter settings

Algorithm	Algorithm parameter settings
PSO	The problem size is n , the number of ants is $m = n$, and other variables are set according to the classical algorithm guidance.
GA	
ASrank	
MMAS	
AS-N	The problem size is n , the number of ants $M = N$ $a = 2, \beta = 8, \rho = 0.02, W = 5$

Table 3 Test results

		ASrank	GA	PSO	MMAS	AS-N
ulyees	Average value	76.6355	76.5267	76.5355	76.1533	75.8746
	Maximum	78.4533	77.2298	77.4509	77.4575	76.0355
	Minimum	75.3625	75.9035	75.9843	75.3672	75.1283
Att48	Average value	33,864.65	33,821.96	32,268.86	33,751.98	33,728.76
	Maximum	34,139	34,684	34,218	34,375	34,092
	Minimum	33083	33,187	33,201	33,548	33,529
Eil76	Average value	572.66	574.63	570.98	567.9876	560.87
	Maximum	589.43	590.86	588.42	584.87	573.54
	Minimum	564.87	564.87	562.64	549.76	548.82

4 Performance test

4.1 Data set

This chapter mainly tests the performance of the algorithm through the TSPLIB data set. The improvement effect of AS-N algorithm is expounded by comparing AS-N algorithm with other classical algorithms on the data set. Table 1 shows some of the issues in the TSPLIB data set.

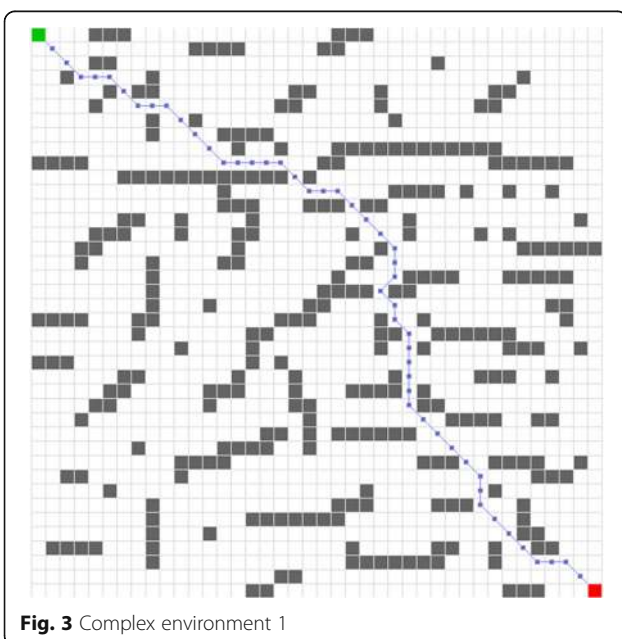
4.2 Parameter settings

Table 2 lists the algorithm’s internal initial value parameter settings. The size of the population is adjusted according to the size of the TSP problem space. For example, eil76 is the coordinate point of 76 cities in Christofides, so the population size is 76. The number of iterations per test is set to 200 generations, and the number of independent runs of each algorithm is 30.

4.3 Analysis of results

This section uses data analysis methods on different test questions to describe the algorithm and describe the algorithm’s search capabilities. In the test, this article used three questions in the TSPLIB test data set, and tested 30 times for each question. Both the algorithm test work and the result data analysis were performed using the MATLAB 2014 software tool. The experiment recorded 30 sets of test results for each question; took the average, maximum, and minimum values of the data for data analysis; and evaluated the performance of the algorithm. Table 3 gives the test results for each algorithm.

From the perspective of data analysis, the improved algorithm AS-N has achieved better results, and it is better to search for the optimal solution stably and effectively. Among them, the MMAS algorithm performs second, which is obviously better than the AS, GA, PSO and MMAS algorithms.



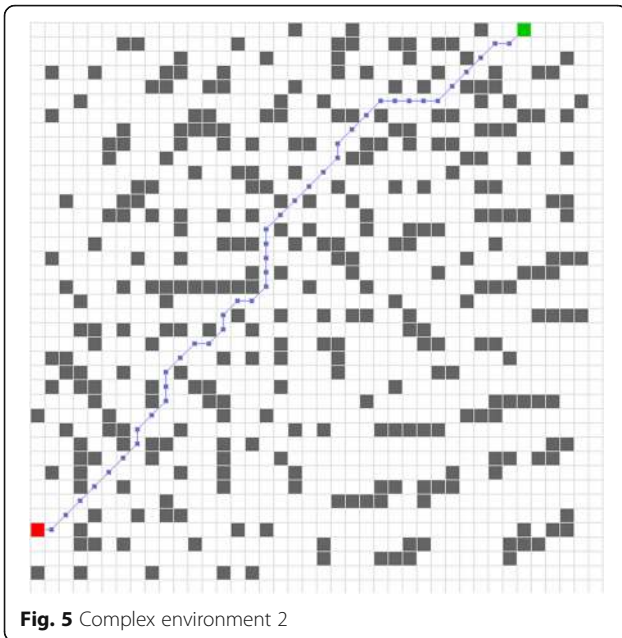


Fig. 5 Complex environment 2

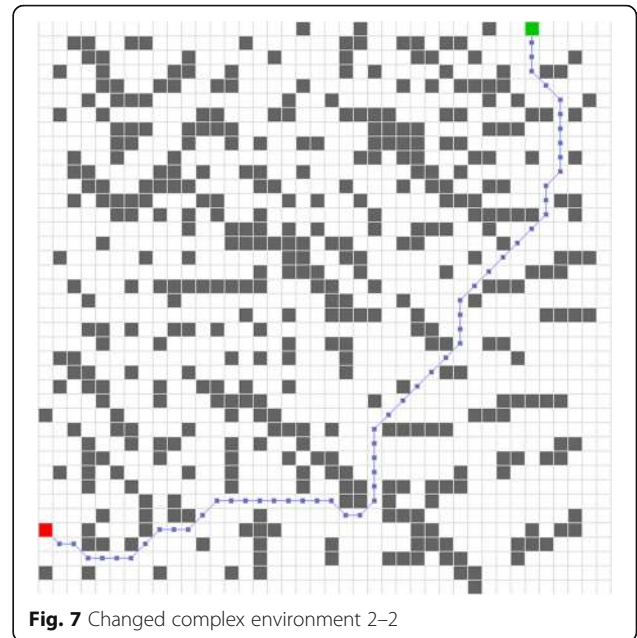


Fig. 7 Changed complex environment 2-2

5 Application experiment

In this section, MAX-MIN Ant System (MMAS) and the ant colony algorithm with punitive measures (AS-N) are compared experimentally. The MMAS algorithm is the most typical representative of the direct improvement of the basic ant colony algorithm. The experiment was completed in vs2013, and the computer was configured as 4GB RAM, 2.50-GHz processor.

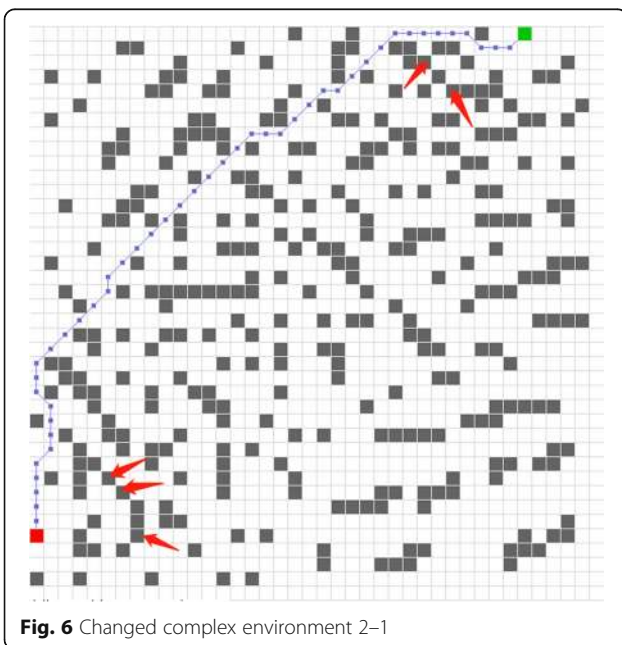


Fig. 6 Changed complex environment 2-1

5.1 Parameter setting and operation result

The ant colony algorithm includes parameters such as α , β , ρ , γ , δ , and number of ants: The optimal parameter settings are derived with reference to the classical ant colony algorithm and a large number of references. The pheromone intensity α is limited to [1, 5] and is set to $\alpha = 2$ during the experiment; the ant visibility β is limited to [5, 12], and the experimental data is set to $\beta = 8$. The evaporation rate ρ is limited to [0.01, 0.05], and $\rho = 0.02$ during the experiment. The modeling environment is: design a 40×40 grid work area with different complexity. The starting point is at (1, 1), the target point is (40, 40), and each small grid is 1 cm long. In a dynamic environment, testing is performed using two different complex environments 1, 2. For the complex environment 1 (Fig. 3), the population size is set to 50. When the 25th generation is run, the existing environment is transformed, the feasible path is intercepted, and the obstacles are placed to search for the new feasible path again. As shown in Figs. 3 and 4 below:

For the complex environment 2 (Fig. 5), the population size is set to 50, and the existing environment is transformed when running 25 or 50 generations, respectively, and other parameters are not changed. Intercept the feasible path twice and set the obstacle to

Table 4 Complex environment 1

Algorithm	MMAS	AS-N
Figure 3 average distance	63.5451	63.4512
Figure 4 average distance	64.7329	64.6763
Average running time	1.1963	1.2305

Table 5 Complex environment 2

Algorithm	MMAS	AS-N
Figure 5 average distance	61.3387	61.0082
Figure 6 average distance	63.8864	63.0581
Figure 7 average distance	64.9977	64.2471
Average running time	3.3951	3.1737

search for the new feasible path again. As shown in Figs. 5, 6, and 7 below:

The data analysis will be carried out separately for the test results of the above simulation environment, the convergence diagram is compared, and the box diagram explains the superiority of the AS-N algorithm.

5.2 Data analysis

For complex problems 1, 2, as shown in Figs. 3 and 5, 30 sets of data are used to compare the superiority of the algorithm. Tables 4 and 5 give the average distance and average running time of the optimization results of the two algorithms AS-N and MMAS. In the dynamic environment, changing the search environment and improving the superiority of the algorithm AS-N can clearly see that the improved algorithm AS-N has achieved ideal results regardless of running time or optimal distance.

5.3 Box chart comparison

It can be concluded from the box graph data that the composition of the data is the optimal solution obtained for each set of tests. In a dynamic environment, this section shows the search results before and after environmental changes, as shown in Fig. 8. The plus sign in the figure is the abnormal point, the red line represents the median value, the blue line above the red line is the maximum value, and the blue line below the red line is

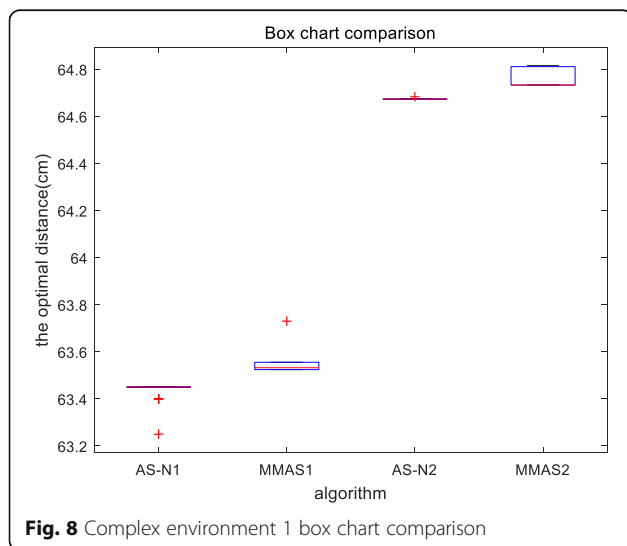


Fig. 8 Complex environment 1 box chart comparison

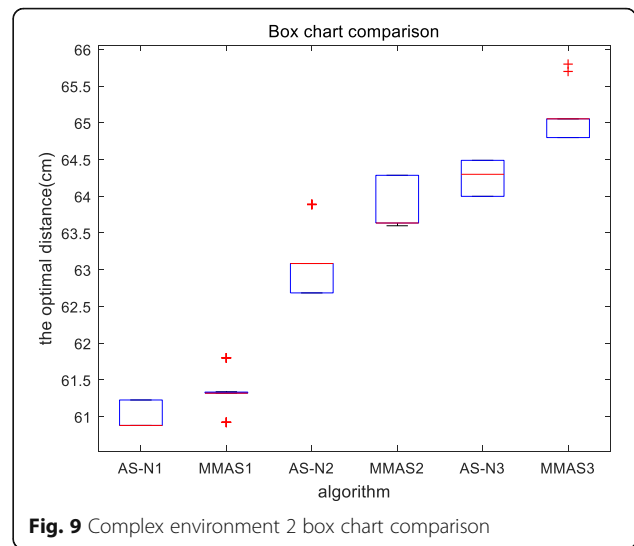


Fig. 9 Complex environment 2 box chart comparison

the minimum value. The figure shows the average of 30 sets of data. AS-N1 and MMAS1 represent the data distribution in the initial environment, that is, the environment in Fig. 3, and AS-N2 and MMAS2 represent the data distribution in the environment of Fig. 4, that is, the data distribution after the obstacle is added on the basis of Fig. 3. It can be seen from Fig. 8 that AS-N is more concentrated than MMAS algorithm, and the results obtained by multiple tests are more stable.

Figure 9 shows the data distribution of the complex environment 2. Similarly, the three sets of arrays are the results of running multiple times in Figs. 5, 6, and 7. In the figure, the AS-N and MMAS algorithms, the median and minimum values of AS-N, are smaller than the MMAS.

In summary, the distribution of data on the box graph can more clearly see the superiority of the AS-N

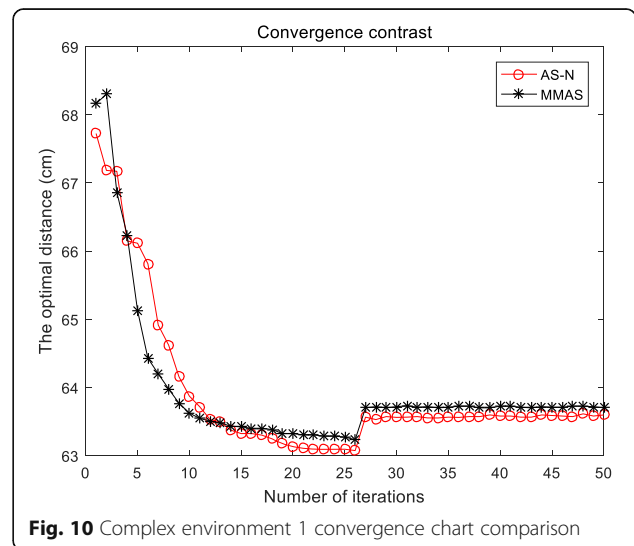
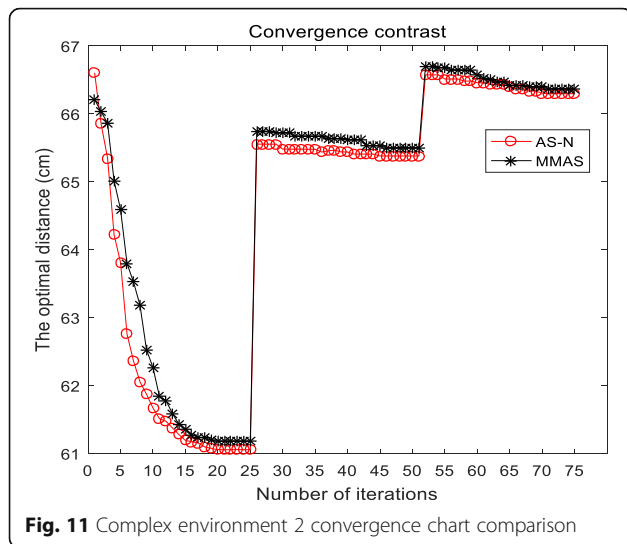


Fig. 10 Complex environment 1 convergence chart comparison



algorithm, and its data distribution is relatively stable, which leads to a stronger ability to find and optimize.

5.4 Convergence contrast

It can be seen from the illustration that in the dynamic environment, the active occlusion of the ant feasible path in the experiment, but the ant can quickly find the feasible path and reach the target point. The convergence diagram in different environments is shown in Figs. 10 and 11. It can be seen that the AS-N algorithm not only can search for the optimal path in the static environment, but also maintains the superiority of the algorithm in the dynamic environment, and its adaptability to the environment changes is also very high.

6 Conclusion

Ant colony algorithms have been widely used to solve various optimization problems in different fields, especially in the field of engineering design. Ant colony algorithm is an intelligent algorithm with positive feedback mechanism. The main content includes the construction of path and the update of pheromone. After an in-depth study of the ant colony algorithm, an ant colony algorithm with punitive measures is proposed. The salient aspect of this punitive measure is that at the end of each generation, the ant finds a poorer path pheromone volatilization rate, thereby reducing the re-exploration of this path and increasing the opportunity to explore the unknown. In this paper, the MMAS algorithm and the AS-N algorithm are used to simulate the unmanned vehicle path planning problem in the dynamic environment. Finally, the simulation results and their comparisons are given. The AS-N algorithm performs better in dealing with unmanned vehicle path planning.

Abbreviation

ACO: Ant colony optimization algorithm; AS-N: Ant colony algorithm with punitive measures; ASrank: Ant system-based rank; MMAS: MAX-MIN Ant System

Acknowledgements

The research presented in this paper was supported by Ministry of Science and Technology of the People's Republic of China and National Natural Science Foundation of China.

Authors' contributions

LY is the main writer of this paper. He proposed the main idea and applied the MMAS algorithm and AS-N algorithm to simulate the unmanned vehicle path planning problem in dynamic environment. HC proposed MMAS algorithm and AS-N algorithm and gave some important suggestions for the simulation. All authors read and approved the final manuscript.

Funding

This work is supported by National key Research and Development Program of China under grant No. (2017YFB1103603, 2017YFB1103003), National Natural Science Foundation of China under grant No. (61602343, 51607122, 61772365, 41772123, 61802280, 61806143, 61502318), Tianjin Province Science and Technology Projects under grant No. (17JCYBJC15100, 17JCNJC04500) and Basic Scientific Research Business Funded Projects of Tianjin (2017KJ093, 2017KJ094), and Natural Science Foundation of Henan province under grant No. (182300410286).

Availability of data and materials

The simulation code can be obtained by contacting the e-mail of corresponding authors.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹School of Mechanical and Electrical Engineering, Henan University of Technology, Zhengzhou, China. ²School of Computer Science and Technology, Tianjin Polytechnic University, Tianjin, China.

Received: 20 February 2019 Accepted: 15 May 2019

Published online: 28 May 2019

References

- Hsu C C, Hou R Y, Wang W Y. Path Planning for Mobile Robots Based on Improved Ant Colony Optimization[C]. Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, 2013. A
- H. Shorakaei, M. Vahdani, B. Imani, et al., Optimal cooperative path planning of unmanned aerial vehicles by a parallel genetic algorithm[J]. *Robotica* **34**(04), 823–836 (2016)
- L. Wang, R.H. Shi, Application of particle swarm optimization algorithm to path planning of unmanned aerial vehicle[J]. *Comput. Simul.* **28**(4), 74–77 (2011).
- Y.W. Fan, C.M. Zhang, Hybrid differential evolution algorithm for path planning of mobile robots. *J Taiyuan Univer Sci Technol* **40**(01), 6–12 (2019)
- M. Dorigo, T. Stützle, in *Handbook of Metaheuristics*. Ant colony optimization: overview and recent advances (Springer, Cham, 2019), pp. 311–351
- F. Zheng, A. Zecchin, J. Newman, et al, An adaptive convergence-trajectory controlled ant colony optimization algorithm with application to water distribution system design problems[J]. *IEEE Trans. Evol. Comput.* **21**(5), 773–791 (2017)
- Cheng J , Miao Z , Li B , et al. An improved ACO algorithm for mobile robot path planning. *IEEE International Conference on Information & Automation*. IEEE, 2017.A
- H. Qu, L.W. Huang, X. Ke, Research of improved ant colony based robot path planning under dynamic environment[J]. *Dianzi Keji Daxue Xuebao/J. Univ. Electron. Sci. Technol. China*, **44**(2), 260–265 (2015).

9. G. Baoqing, H. Shuyun, Z. Liqiang, et al., Multi-AGV parking path planning based on improved ant colony algorithm. *J. Transp. Syst. Eng. Eng.* **18**(6), 55–62, 80 (2018)
10. J. Cheng, Z. Miao, B. Li, et al., *An improved ACO algorithm for mobile robot path planning*, Information and Automation (ICIA), 2016 IEEE International Conference on. IEEE (2016), pp. 963–968
11. S.P. Carabaza, E. Besada-Portas, J.A. Lopez-Orozco, et al, Ant colony optimization for Multi-UAV minimum time search in uncertain domains[J]. *Appl. Soft Comput.* **62**, 789-806 (2017).
12. J. Ning, Q. Zhang, C. Zhang, et al., A best-path-updating information-guided ant colony optimization algorithm. *Inf. Sci.* **s 433–434**, 142–162 (2018)
13. S. Bououden, M. Chadli, H.R. Karimi, An ant colony optimization-based fuzzy predictive control approach for nonlinear processes. *Inf. Sci.* **299**(C), 143–158 (2015)
14. W. Wu, Y. Tian, T. Jin, A label based ant colony algorithm for heterogeneous vehicle routing with mixed backhaul. *Appl. Soft Comput.* **47**, 224–234 (2016)
15. Q. Zhang, S. Xiong, Routing optimization of emergency grain distribution vehicles using the immune ant colony optimization algorithm. *Appl. Soft Comput.* **71**, 917–925 (2018)
16. O. Castillo, H. Neyoy, J. Soria, et al., A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot. *Appl. Soft Comput.* **28**(C), 150–159 (2015)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
