ITC 3/49

Information Technology and Control
Vol. 49 / No. 3 / 2020
pp. 438-447
DOI 10.5755/j01.itc.49.3.25295

Optimal Search and Rescue Route Design Using an Improved Ant Colony Optimization

Received 2020/02/14 | Accepted after revision 2020/06/27

crossref **http://dx.doi.org/10.5755/j01.itc.49.3.25295**

HOW TO CITE: Zhang, H., Sun, J., Yang, B., Shi, Y., Li, Z. (2020). Optimal Search and Rescue Route Design Using an Improved Ant Colony Optimization. *Information Technology and Control*, 49(3), 438-447. https://doi.org/10.5755/j01.itc.49.3.25295

Optimal Search and Rescue Route Design Using an Improved Ant Colony Optimization

Haichuan Zhang*, Jingwen Sun, Baolong Yang, Yinghu Shi, Zhanying Li
School of Information Science and Engineering, Dalian Polytechnic University, Dalian, China, 116034;
e-mail: zhanghc@dlpu.edu.cn

***Corresponding author:** zhanghc@dlpu.edu.cn

This paper presents a search and rescue route design algorithm to improve the efficiency of maritime search and rescue. This algorithm is based on the basic Ant Colony Algorithm. To solve the problem that the Ant Colony Algorithm is easy to fall into local optimal solutions in the process of searching, the pheromone concentration updating strategy of the original Ant Colony Algorithm is improved. Compared with the original algorithm, the probability of finding the optimal solution in this paper can be improved by about 30% at most. The path weight based on the time of falling into the water is introduced to make the algorithm more realistic. The simulation results show that the improved algorithm can improve the search efficiency and speed, and can combine with the actual situation to get a better route due to the introduction of weight in the algorithm.

KEYWORDS: Ant Colony Optimization, Path planning, Maritime search and rescue.

1. Introduction

In recent years, the number and scale of marine activities are increasing, so is the probability of marine accidents. Maritime search and rescue as the security of each personnel, has an important position. The route design of vessels is an essential activity in the process of search and rescue [2]. Designing a suitable route can shorten the search and rescue time effectively and improve the search and rescue efficiency. In addition, due to the complex maritime environment, the success of the search and rescue depends on many factors. For instance, the position of each search and rescue target will be moved due to the influence of wind and waves, which brings great disadvantage. In addition, since the survival rate is greatly improved if the drowning person is rescued within 24 hours of falling into the water, it is necessary to try to ensure that the rescue time for each person is not too long. Most route planning algorithms are usually based on ideal conditions. Therefore, a more suitable algorithm is needed for the design of search and rescue routes. At present, some research on route design has been put forward. For instance, a new multi-criteria ACO-based algorithm was proposed to solve a path planning problem for ships in the environment with static and dynamic obstacles [9]. Meanwhile, to save the searching time, a dynamic optimizing ship routing algorithm was applied to the dynamic ship routing in complex navigation conditions successfully [6]. Moreover, a path-planning algorithm, according to evolutionary algorithm, was come up in navigation situation [17, 18]. Furthermore, an automatic method is developed to improve its searching efficiency using simulated annealing algorithm [8].

There are many algorithms, as Dijkstra, A*and so on, to calculate the optimal path. Due to the complexity of problem, many excellent algorithms, such as Ant Colony Optimization (ACO), Dragonfly Algorithm and Polar Bear Algorithm have appeared in recent years [11, 14, 16]. Among them, Ant Colony Algorithm has been widely applied in various fields of life due to its positive feedback, strong robustness and easy integration with other algorithms [3, 4]. Inspired by the behavior of ants in finding food, Italian scholars M. Dorigo, V. Maniezzo and A. Colorni introduced Ant Colony Optimization in the early 1990s. Compared with other heuristic algorithms, the inspiration of this algorithm is unique. The original version of the algorithm is suitable for finding the optimal solution problems. However, with a lot of modification in this algorithms these days that make it capable of solving a wide range of problems [15]. The algorithm has been applied in the field of analyze 2D input images [12], path planning for mobile robot [1, 10, 23], plan the 3D measuring path for coordinate measuring machines [5] and multi-path routing in LEO satellite networks [7].

The increasing demand of calculation methods for solving optimization problems causes that parallelization and modification of the existing algorithms are necessary. Because the Ant Colony Algorithm is universal and easy to be integrated with other calculation methods, there are several ways to optimize it. The optimization of the algorithm mainly includes time minimization, search efficiency improvement and parameter improvement [13]. This paper uses an improved ACO to design search and rescue routes. By improving the strategy of pheromone concentration updating in original ACO, the algorithm is more likely to found the optimal solution. This alleviates the defect that the Ant Colony Algorithm is easy to fall into the local optimum effectively. According to the actual situation of maritime search and rescue, the path weight based on the time of falling into the water is introduced into the algorithm. This makes the algorithm take the falling time of each target into consideration in the calculation process, so as to make the calculation of the optimal route more reasonable. The algorithm can timely adjust the search and rescue route with the increase of the falling time, this can ensure the timeliness of the rescue. Experimental results show that the improved algorithm can choose a path more effectively. Combining with the actual situation, the algorithm is introduced with the relevant weight to make the algorithm more meaningful in practical application rescue ships well.

2. Ant Colony Optimization

Ant Colony Optimization is a bionic optimization search algorithm, and has obvious advantages to solve some complex problems. It develops based on the behavior of ants in finding their short path to the food. During the process of searching food, each ant releases an amount of pheromone on the path where it passed. Meanwhile the ants always tend to travel towards the trail where the pheromone concentration is high. Therefore, the algorithm shows positive feedback. The higher the pheromone concentration on the path, the more ants will chose it. Eventually, the ants find the shortest path to the food in this way [20, 22, 24].

In the original Ant Colony Algorithm, the ant k will choose its next grid according to

$$P_{ij}^{k}(t)=\begin{cases}\frac{[\tau_{ij}(t)]^{\alpha}\times[\eta_{ij}(t)]^{\beta}}{\sum_{k\in \mathrm{allowed}_k}[\tau_{ik}(t)]^{\alpha}\times[\eta_{ik}(t)]^{\beta}}, & j\in \mathrm{allowed}_k, \\ 0 & ,\ \mathrm{other}\end{cases} \tag{1}$$

where $P_{ij}^{k}(t)$ is the probability of ant k moving from grid node i to node j at time t; $\tau_{ij}(t)$ is the intensity of the pheromone between grid cubes node i and j; $\eta_{ij}(t)$ represents the heuristic function between grid cubes i and j (In this paper, $\eta_{ij}(t)=1/d_{ij}$, d_{ij} is the distance between satellite i and satellite j); α and β are weighting parameters that show the relative influence of the pheromone and distance.

When all the ants complete the travelling, the pheromone levels on each arc will be updated by volatilizing the old pheromone and adding the pheromones deposited by each ant. The pheromone updating formula is as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{2}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t), \tag{3}$$

where ρ is pheromone evaporation rate and (1-ρ) represents the pheromone residual rate, 0<ρ<1; $\Delta\tau_{ij}^{k}(t)$ represents the amount of pheromone left by the ant k at the current iteration. There are three different algorithmic models for $\Delta\tau_{ij}^{k}(t)$: Ant-Quantity, Ant-Cycle, Ant-Density. They were all proposed by M. Dorigo. Ant-Cycle is often used because of its good performance which can be expressed as follows

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L_k}, if\ ant\ k\ travels\ on\ edge(i;\ j) \\ 0\ , \quad\ otherwise \end{cases}, \tag{4}$$

where Q is a constant and L_k represents the length of the path covered by the ant k.

3.Improved Ant Colony Optimization

Ant Colony Algorithm shows good performance in solving shortest path, but it also has some shortcomings. In this paper, the Ant Colony Algorithm is optimized by improving the pheromone concentration update strategy, and path weights are introduced to calculate a reasonable route for search and rescue vessels.

3.1. Improvement of Pheromone Update Rules

The primary Ant Colony Algorithm still has some disadvantages during searching, such as the searching time is excessively long and the probabilistic selection may fall into local optimum. The improved Ant Colony Optimization can improve the local pheromone update rule, which makes the result more likely to achieve an optimal solution.

The improved Ant Colony Optimization has introduced adaptive dynamic factors σ into pheromone update strategy, which makes the pheromone concentration reflect the path information better. The adaptive dynamic factor σ can make the pheromone concentration get a larger addition in better path by control the updating proportion adaptively of the optimal pheromone concentration in an iteration.

The improved pheromone updating formula can be shown as:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \mu\sigma\Delta\tau_{ij}(t) \tag{5}$$

$$\mu = \sqrt{\frac{L_{min}}{L_K}} \tag{6}$$

$$\sigma = \frac{1}{\pi} arctan\left(\lambda \frac{\bar{L}-L_K}{\bar{L}-L_{min}}\right) + \frac{1}{2}, \tag{7}$$

where σ represents the adaptive dynamic factors and b represents the coefficient of the d adaptive dynamic factors; L_{min} is the shortest length at the current iteration; L is the average of length for all ants at the current iteration, μ is the coefficient of σ. The value of μ depends on L_K. The closer the value of L_K is to the value of L_{min}, the larger the value of μ.

The adaptive dynamic factor σ above is the inverse tangent function

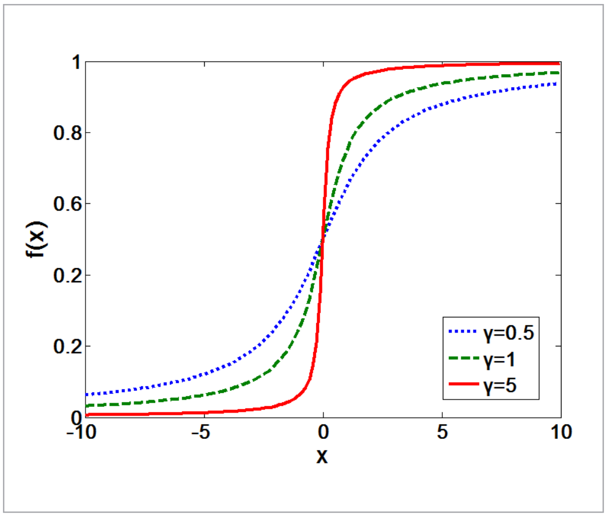
$$f(x) = \frac{1}{\pi} arctan(\varphi x) + \frac{1}{2}, \tag{8}$$

where $x = \frac{\bar{L}-L_K}{\bar{L}-L_{min}}$. The output curve of the function is smooth, and the value of the function is in the range (0,1). Based on the Equation (8), a conclusion can be drawn that when is larger and the search path is longer, the dynamic factor is closer to 0. On the contrary, when is smaller and the search path is shorter, the dynamic factor is closer to 1.

The increment of pheromone in each cycle, in the basic Ant Colony Algorithm, just depends on the total distance travelled by each ant. However, the adaptive dynamic factors selected in this paper can adjust the update of pheromone adaptively. With help of that, pheromones can also obtain the increment according to the optimal solution and average length of the current iteration. The hyperbolic tangent function with different values of γ are shown in Figure 1 below. It is obvious that the function looks different when the parameter γ is varied. The higher the value of γ is, the higher the sensitivity of the function is.

Figure 1

Graph of dynamic factor function



3.2. Ant Colony Algorithm Based on Path Weight

The mathematical model of Ant Colony Algorithm is an ideal assumption. However, there are many external factors in the actual maritime search and rescue operation. In order to increase the practicability of Ant Colony Algorithm in maritime search and rescue route selection, the path weight matrix is introduced into the algorithm.

When choosing search and rescue routes, optimal path is not necessarily the shortest path, time of falling into the water of each search and rescue target should also be taken into account. Search and rescue vessels should have priority to reach the target with a longer time to fall into the water if the distance between the two roads is not much different. In this paper, the weight of each path is valued depends on how long the next target point is in the water, such as T1, T2, T3, T4. According to the duration of each target point, the weight value of all routes are calculated as follows:

$$\varphi_{ij} = (1-\varepsilon) + \varepsilon e^{T_j - 20}, \tag{9}$$

where, T_j is the falling time of target j; ε is a parameter that controls the degree of impact of the fall time, 0<ε<1. The value of weight will change with the increase of time. Since the exponential function is included in the formula, the longer the falling time of the target point is, the faster the weight value will increase. That is, when the falling time exceeds 20 hours, the weight value will increase sharply..

According to t1he weight value of each path, the corresponding state transition probability can be obtained as:

$$P_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha} \times [\eta_{ij}(t)]^{\beta} \times [\varphi_{ij}(t)]^{\gamma}}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^{\alpha} \times [\eta_{ik}(t)]^{\beta} \times [\varphi_{ik}(t)]^{\gamma}}, j \in allowed_k, \\ 0 \qquad , other \end{cases} \tag{10}$$

where φ_{ij} is the weight of path (i,j) at time t; γ is the factor of weight.

3.3. Overall Procedure of Improved Ant Colony Optimization

According to the description of above improvement method, the procedures of the design of search and rescue routes are described as follows:

Step1. Enter the coordinates of each target point received. Set the parameters according to the actual situation

Step2. Initialize algorithm.

Step3. Each ant starts at the starting point, the initial position of the search vessel, constructs its route according to Equation (10). Update the value of every ant's tabu table and L_K.

Step4. When all ants reach the end, record the value of L_K and $\bar{L}$ in this iteration. The updating rule in Equations (5)-(7) is applied to change pheromone level.

Step5. Update the global optimal solution. If the current shortest path length is shorter than the global shortest path length, the global solution is replaced by the current solution.

Step6. If the largest iteration number reaches, the calculation stops. Otherwise, go to Step3.

The pseudocode of the improved algorithm is as follows:

```
procedure AS
    for each edge
        set initial pheromone $t_{0.}$
    end for
while not stop
    for esch ant $k$
        set an initial city.
        for $i$=1 to $n$
        choose next city j with the probability
        given by Equation (10).
        end for
    end for
    compute the length ck of the tour
    constructed by the $k$th ant.
    for each edge
    update the pheromone value by Equation (5).
    end for
end while
print result.
end procedure
```

4. Algorithm Simulation

This article uses MATLAB simulation for both the original and improved algorithms in different grid maps to verify the effectiveness and reliability of the improved ACO.

The simulation experiment set a point as the starting point and have other 10 points. It aims to find the shortest path by going through 10 points from the starting point. Experimental method is used to determine the optimal combination of each parameter in the algorithm, and the values of parameters are shown in Table 1, where α, β and γ represent the weight factors of pheromone, distance and falling time, respectively. The different collocation of these parameters may affect the calculation, and the values of these parameters are generally between [0.7-1]. Pheromone recurrence factor ρ determines the residual pheromone after each round of search. The reasonable value of this parameter can prevent the pheromone concentration from increasing rapidly and make the algorithm fall into local optimum. λ determines the function image of the adaptive dynamic factor, and the value in this paper is between [1-2]. Moreover, the number of ants(k) is twice the number of target points.

Table 1

The values of parameters

parameter	value
Pheromone factor (α)	0.7
Heuristic factor (β)	0.9
Time factor (γ)	1
Volatility coefficient (ρ)	0.3
The parameter of function (λ)	2
Pheromone amount(Q)	100
Maximum iterations(m)	200

When the weight of path is not introduced (φ_{ij} = 1), the shortest one is the optimal path. Table 2 shows a comparison between the results of the basic ACO, the improved ACO and another modified Ant Colony mentioned in literature [22, 24]. The experiment simulates 10, 15 and 20 target points, respectively, and was performed 1000 times each.

Table 2

(a) Statistical results of the probability of finding the optimal solution/%

The number of target points	10	15	20
original ACO	67.5	51.7	37.8
algorithm in [21]	89.7	74.5	49.0
algorithm in [19]	90.3	77.2	54.7
improved ACO	97.9	80	68.3

(b) Normalize the length of the average path

The number of target points	10	15	20
original ACO	1.0020	1.0076	1.0023
algorithm in [21]	1.0006	1.0050	1.0007
algorithm in [19]	1.0005	1.0064	1.0005
improved ACO	1.0001	1.0043	1.0003

(c) Statistical results of average iteration times

The number of target points	10	15	20
original ACO	50.812	80.122	61.926
algorithm in [21]	42.480	76.476	67.044
algorithm in [19]	33.808	55.008	41.403
improved ACO	33.642	62.793	38.378

Table 2(a) shows the probability of finding the optimal solution under different conditions, that's the probability of finding the optimal solution in 1000 trials. It is observed that the improved algorithm is more likely to find the optimal solution by the algorithm in this paper has improved the pheromone updating strategy. Due to the introduction of adaptive dynamic factor in pheromone updating strategy, the improved algorithm makes pheromone updating process more reasonable. In this way, the problem of the algorithm easily falling into the local optimal solution are alleviated, and the probability of finding the optimal solution is improved.

Calculate the average length of all paths calculated by each group and normalizing the length of the average path (the length of the average path / the length of the shortest path). The results are shown in Table 2(b). It can be concluded that the results of normalizing the length of the average path in the improved algorithm is smaller. That is, the search results of the algorithm mainly focus on the shortest path and some suboptimal path

Count the average of iterations in each experiment shows in Table 2(c). It can be seen that the algorithm has good convergence and can find the optimal solution quickly.

In order to prove the improvement effect of the algorithm, preform the non-parametric ranking based Friedman test in Table 3. The ranking is on account of the calculation results of each algorithm under three indexes. Based on this, the comparison of each algorithm are analyzed.

The test statistics are calculated as follows:

$$X^2 = \frac{12}{3 \times 4 \times 5}(12^2 + 8^2 + 6^2 + 4^2) - 3 \times 3 \times 5 = 7 \quad (11)$$

Since X^2 is greater than $X^2_{0.05[2]}$, it can be concluded that the calculation results of the four algorithms are significantly different. That is, the improved algorithm is significantly improved compared with other algorithms.

Figure 2 shows the convergence of different algorithms. It can be seen more intuitively that the original ACO may stop searching in the early stages of optimization and fall into local optimum. However, the addition of the adaptive dynamic factors , the improved algorithm will keep searching to find the optimal solution.

Figure 2

Contrast diagram of convergence

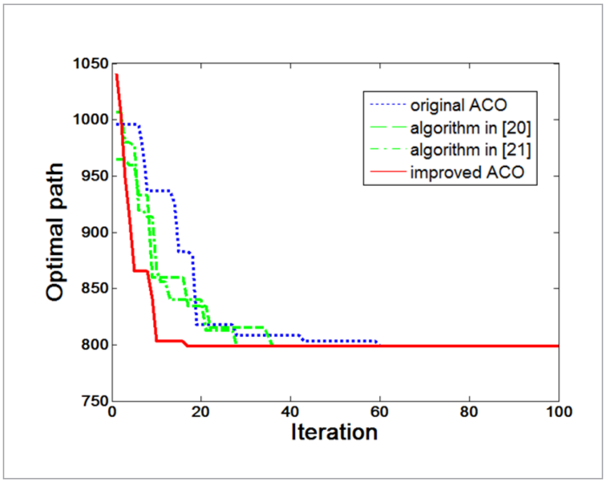


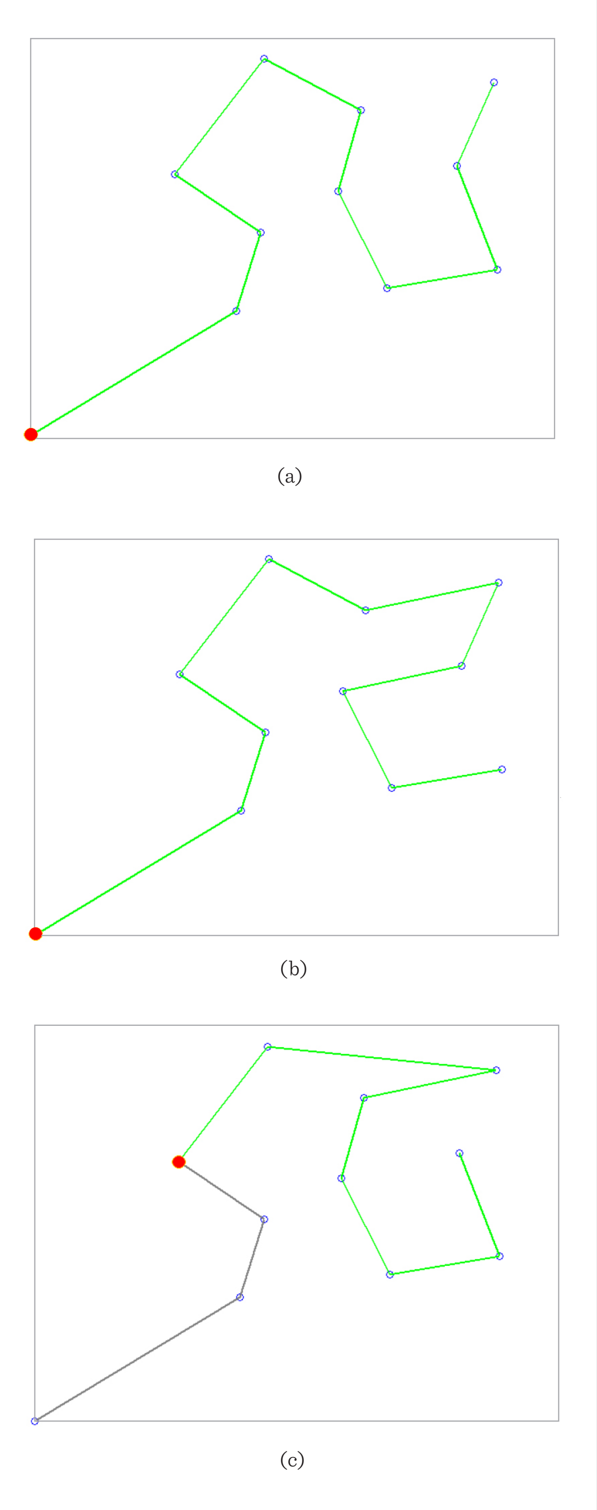
Table 3

Ranking the indicators

	original ACO	algorithm in [21]	algorithm in [19]	improved ACO
Probability of search	4	3	2	1
average path	4	2	3	1
average iteration times	4	3	1	2

Figure 3

Optimal path planning


(a)

(b)

(c)

When the path weight is not introduced, the algorithm is required to find the shortest route as many other ACOs. In the experiment No. 1 in Table 2, the optimal path optimization result is shown in Figure 3(a). The total distance of the path is 798.4493. When the path weight is introduced as need, the weight value of each path is related to the corresponding falling time. The optimal path optimization result is shown in Figure 3(b). The total distance of the path is 817.5914.

The comparison test results show that when the time factor is introduced as the path weight, as the Ant Colony Algorithm is used to design the search and rescue route, the optimal route may become longer, but it will go through the target point with a longer time of falling into the water first. In addition, as the time of drowning increases, the route will be optimized in real time, as shown in Figure 3(c). If a person falls into the water for too long in the process of search and rescue, the route will be changed in real time to ensure the survival of the personnel. In this way, everyone's rescue time can be guaranteed to ensure the survival rate. This has important practical significance in search and rescue operations.

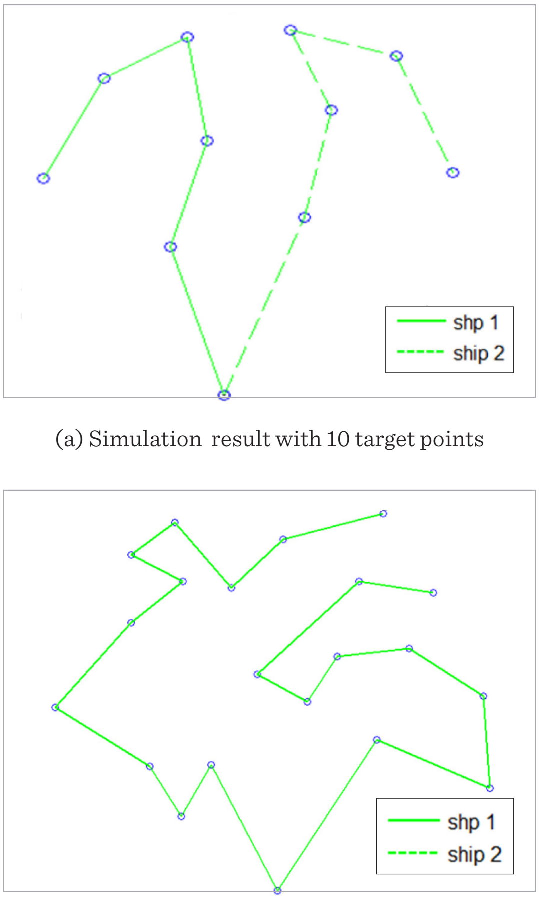
5. The Application of Improved ACO on Search and Rescue Based on Multi-Objective

Maritime search and rescue system is mainly divided into two parts, monitoring center and user terminal. Here is the working principle of the system. When a person falls into water, the user terminal will send positioning information to the monitoring center which will design a reasonable search and rescue route afterwards. Then, the ship will search and rescue according to that.

In the process of maritime search and rescue, there are usually many targets. The targets may drift due to the action of water flow and wind, and their positions will change. Meanwhile, the number of search and rescue vessels is often more than one. Therefore, this algorithm can also be used to design reasonable search and rescue routes for multiple vessels, in order to achieve higher efficiency in search and rescue.

MATLAB is used to simulate the search and rescue route design, the experiment assumed that there

Figure 4
Simulation result



(a) Simulation result with 10 target points

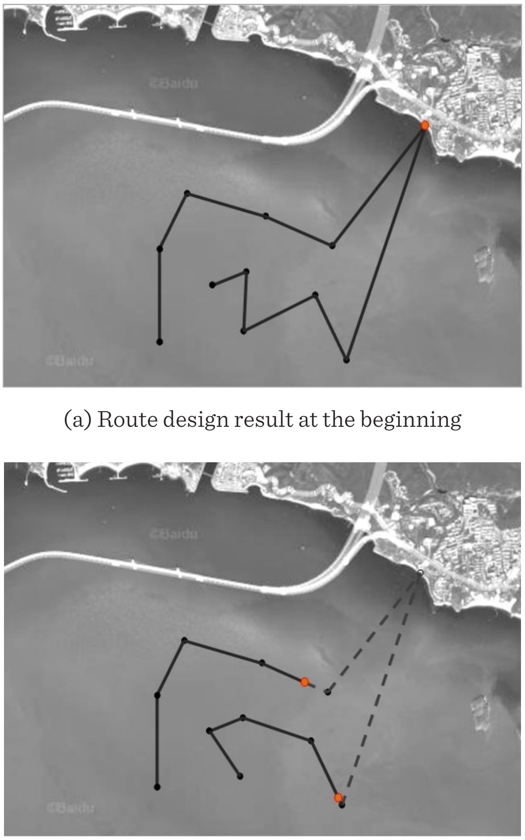
(b) Simulation result with 20 target points

are two vessels. The result of route design is shown in Figure 4 and in Figure 4(a) is the route designed with 10 target points, while there are 20 target points, the route is shown in Figure 4(b). It can be proved that this algorithm can design reasonable route for multi-target search and rescue. In the course of route design, the falling time factor is taken into account, and the route can be optimized in real time during search and rescue.

The effect of the algorithm in the actual application is shown in Figures 5-6. There are 10 search and rescue points distributed in one sea area, and two search and rescue vessels on the shore. Each search and rescue point sends out the help signal at different times. Based on the location and time information of each point received, the monitoring center designs a reasonable search and rescue route for the two vessels to achieve better results. The route design results of the two ships are shown in Figure 5(a). In addition, as the search time increases, the route can be adjusted in real time in Figure 5(b). This can avoid the danger of some targets being too long due to falling water, so as to ensure the survival rate of personnel.

The experiment proves that the improved ACO can design routes for multiple targets effectively, and can be applied to multiple search and rescue vessels. Since the time of falling into the water is taken into account in the algorithm, it is more in line with actual

Figure 5
Route design results



(a) Route design result at the beginning

(b) Route design result during driving

needs. The use of this algorithm to design search and rescue routes can improve the timeliness of search and rescue effectively, and has great significance for maritime search and rescue operations.

6. Conclusion

This article proposes a solution to the problem that the original ACO is easy to fall into local optimum during searching. By adding inverse tangent function as dynamic factor in updating pheromone, the improved Ant Colony Optimization can adaptively adjust the pheromone update strategy of the optimal solution in each iteration. It can effectively alleviate the problem that the Ant Colony Algorithm is prone to fall into the local optimal solution, and the probability of getting an optimal solution increase. At the same time, the path weight is introduced into the algorithm, so that the obtained optimal path is more in line with the requirements in practical applications.

In addition, the improved Ant Colony Optimization can be applied to maritime search and rescue activities. Simulation results show that the algorithm can effectively design a reasonable search and rescue routes for multiple vessels. This is of great significance in maritime search and rescue.

References

1. Akka, K., Khaber, F. Mobile Robot Path Planning Using an Improved Ant Colony Optimization. International Journal of Advanced Robotic Systems, 2018, 1-7. https://doi.org/10.1177/1729881418774673
2. Breivik, Ø., Allen, A. A., Maisondieu, C., Olagnon, M. Advances in Search and Rescue at Sea. Ocean Dynamics, 2013, 63, 83-88. https://doi.org/10.1007/s10236-012-0581-1
3. Brociek, R., Słota, D. Application and Comparison of the Intelligent Algorithms to Solve the Fractional Heat Conduction Inverse Problem. Information Technology and Control, 2016, 45(2), 184-194. https://doi.org/10.5755/j01.itc.45.2.13716
4. Brociek, R., Słota, D. Application of Real Ant Colony Optimization Algorithm to Solve Space and Time Fractional Heat Conduction Inverse Problem. Information Technology and Control, 2017, 46(2), 171-182. https://doi.org/10.5755/j01.itc.46.2.17298
5. Han, Z., Liu, S., Yu, F., Zhang, X., Zhang, G. A 3D Measuring Path Planning Strategy for Intelligent CMMs Based on an Improved Ant Colony Algorithm. International Journal of Advanced Manufactoring Technology, 2017, 93, 1487-1497. https://doi.org/10.1007/s00170-017-0503-y
6. Haobing, N., Shengzheng, W., Zhiwu, H., Chaojian, S. Dynamic Optimizing Ship Routeing Algorithm for Applications in Maritime Search and Rescue. Journal of Shanghai Maritime University, 2011, 32(4), 1-6, 37.
7. Houtian, W., Qi, Z., Xiangjun, X., Ying, T., Dong, C., Naijin, L. A Multi-Path Routing Algorithm of LEO Satellite Networks Based on an Improved Ant Colony System. High Technology Letters, 2014, 20(3), 253-260.
8. Hui, Y., Zhi. W. Design of Optimum Route Based On Ant Colony Simulated Annealing Algorithm. Modern Electronics Technique, 2009, 11, 188-190.
9. Lazarowska, A. Multi-Criteria ACO-Based Algorithm for Ship's Trajectory Planning. The International Journal on Marine Navigation and Safety of Sea Transportation, 2017, 11(1), 31-36. https://doi.org/10.12716/1001.11.01.02
10. Liu, J., Yang, J., Liu, H., Tian, X., Gao, M. An Improved Ant Colony Algorithm for Robot Path Planning. Soft Computing, 2017, 21, 5829-5839. https://doi.org/10.1007/s00500-016-2161-7
11. Liu, X., Fu, M. Cuckoo Search Algorithm Based on Frog Leaping Local Search and Chaos Theory. Applied Mathematics and Computation, 2015, 266, 1083-1092. https://doi.org/10.1016/j.amc.2015.06.041
12. Mirjalili, S., Dong, J. S., Lewis, A. Ant Colony Optimizer: Theory, Literature Review, and Application in AUV Path Planning. Nature-Inspired Optimizers, 2019, 7-21. https://doi.org/10.1007/978-3-030-12127-3_2
13. Połap, D., Kęsik, K., Woźniak, M., Damaševičius, R. Parallel Technique for the Metaheuristic Algorithms Using Devoted Local Search and Manipulating the Solutions Space. Applied Sciences-Basel, 2018, 8(2). https://doi.org/10.3390/app8020293
14. Połap, D., Wozniak, M. Polar Bear Optimization Algorithm: Meta-Heuristic with Fast Population Movement

and Dynamic Birth and Death Mechanism. Symmetry, 2017, 9, 203, 1-20. https://doi.org/10.3390/sym9100203

15. Połap, D., Woźniak, M., Napoli, C., Tramontana, E., Damaševičius, R. Is the Colony of Ants Able to Recognize Graphic Objects? Information and Software Technologies, 2015, 538, 376-387. https://doi.org/10.1007/978-3-319-24770-0_33
16. Rahman, C. M., Rashid, T. A. Dragonfly Algorithm and Its Applications in Applied Science-Survey. Computational Intelligence and Neuroscience, 2019, 23, 1-24. https://doi.org/10.1155/2019/9293617
17. Śmierzchalski, R., Kuczkowski, Ł., Kolendo, P., Jaworski, B. Distributed Evolutionary Algorithm for Path Planning in Navigation Situation. The International Journal on Marine Navigation and Safety of Sea Transportation, 2013, 7(2), 293-300. https://doi.org/10.12716/1001.07.02.17
18. Tam, C. K., Bucknall, R. Path-Planning Algorithm for Ships in Close-Range Encounters. Journal of Marine Science and Technology, 2010, 15, 395-407. https://doi.org/10.1007/s00773-010-0094-x
19. Wang, P.-D., Tang, G.-Y., Li, Y., Yang, X.-X. Improved Ant Colony Algorithm for Traveling Salesman Problems. Chinese Control and Decision Conference, 2012, 24, 660-664.
20. Wang, P.-D., Tang, G.-Y., Li,Y., Yang, X.-X. Improved Ant Colony Algorithm for Traveling Salesman Problems. Chinese Control and Decision Conference (CCDC), 2012, 24, 660-664.
21. Yabo, Y., Yi, L., Bin, W. Solving Shortest Path Problem with Modified Ant Colony Algorithm. Computer Engineering and Applications, 2016, 52(6), 8-12.
22. Yang, J., Shi, X., Marchese, M., Liang, Y. An Ant Colony Optimization Method for Generalized TSP Problem. Progress in Natural Science 18, 2008, 1417-1422. https://doi.org/10.1016/j.pnsc.2008.03.028
23. Yen, C. T., Cheng, M. F. A Study of Fuzzy Control with Ant Colony Algorithm Used in Mobile Robot for Shortest Path Planning and Obstacle Avoidance. Microsystem Technologies, 2018, 24, 125-135. https://doi.org/10.1007/s00542-016-3192-9
24. Zhang, Z., Li, X. Based on TSP Problem the Research of Improved Ant Colony Algorithms. Electrical Engineering and Control, 2011, 827-833. https://doi.org/10.1007/978-3-642-21765-4_103

This article is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) License (http://creativecommons.org/licenses/by/4.0/).