JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 34, 1063-1077 (2018)
DOI: 10.6688/JISE.201807_34(4).0016

2-OptACO: An Improvement of Ant Colony Optimization for UAV Path in Disaster Rescue*

XIANG JI, QING-YI HUA, AN-WEN WANG+, JUN-SONG TANG,
CHUN-YU LI, FENG CHEN AND DING-YI FANG
School of Information Science and Technology
Northwest University
Xi'an, 710127 P.R. China
E-mail: {jixiang; huaqy}@nwu.edu.cn; {wang_anwen; t_junsong}@163.com;
amylcy@stumail.nwu.edu.cn; {xdcf; dyf}@nwu.edu.cn

Unmanned aerial vehicles (UAVs) are favored by the industry to search and locate lost persons in mountains and trapped persons in earthquakes, fires and other disasters because it is not limited by the obstructions on the ground. Currently, however, a UAV always searches and locates targets along a fixed flight path, which consumes more time and has lower accuracy. This kind of method can only provide a rough position estimation. GuideLoc takes the UAV's GPS coordinates as the location information of a target and the genetic algorithm (GA) is used for path planning in order to shorten the flight path to improve the search efficiency and obtain a good result. But its performance still has room for improvement. In this paper, the path optimization algorithm used in GuideLoc is further discussed and studied, and then a method, 2-OptACO, is proposed. The method uses the 2-optimization (2-opt) algorithm to improve the ant colony optimization algorithm (ACO) and is applied to optimize the UAV's path for search and rescue. The simulation results show that the 2-OptACO method has a faster convergence rate than the GA and ACO. It can obtain a better global optimal solution.

Keywords: ACO, UAV, path planning, 2-OptACO, GA

1. INTRODUCTION

Do you know how many trapped people died in disasters because rescue workers were unable to obtain accurate rescue information? In rescuing, determining the location of trapped persons quickly and accurately is the key to saving more lives. At present, the industry is looking for ways to improve the positioning accuracy and efficiency in disaster sites to solve the problem [1]. Unmanned aerial vehicles (UAVs) [2] are favored by the industry because of the fact that it is not restricted by ground obstacles during searching trapped persons. Currently, however, a UAV always uses a fixed flight path to search and rescue trapped persons, which not only consumes more time, but also has lower accuracy. It can only provide a rough location estimate. But a rescue is urgently required to locate trapped targets quickly, accurately and efficiently. For example, in an earthquake, quickly and accurately positioning trapped persons will help rescue workers gain more opportunities to save more people's lives. Otherwise, the survivors may lose their lives due to the aftershocks.

Received September 4, 2017; revised November 6, 2017; accepted December 15, 2017
Communicated by Xiaohong Jiang.
+ Corresponding author.
* A preliminary version of this paper was presented in 2017 International Conference on Networking and Network Applications.

The most effective way to locate trapped people is to use wireless signals sent by the radio equipment (such as a cellphone) carried by the trapped people. There are three types of the methods: The first is ground detection [3, 4]. The influence of roads and obstacles makes the efficiency low. The second is the rough detection of aircrafts [1, 4-6]. The current methods are high in cost and low in accuracy. The third is the accurate detection that UAVs traverse trapped targets [7]. Cost and energy consumption are the key issues.

In this paper, the third method is further discussed. In order to make the scheme faster and save energy for locating trapped people, a new method for planning UAV's path is proposed. The method is based on the 2-optimization (2-opt) algorithm [8, 9] to improve the ant colony optimization algorithm (ACO) [10, 11], which is applied to the search and rescue path optimization of UAV. The simulation results show that the 2-OptACO method has a faster convergence rate than the genetic algorithm (GA) and can obtain a better global optimal solution. In accomplishing the same task, the flight path planned by this method is shorter. Because UAV's energy consumption has a substantially positive correlation with its flight distance, this method can save UAV's energy consumption in completing the same locating task.

Contribution:

- This paper designs a simple and efficient partition traversal strategy. It is a deterministic solution, not an approximate algorithm.
- It proposes a path optimization method combining 2-opt with ACO. Compared with GA and the traditional ACO, its speed of convergence is fast, its calculation time is short, and its optimized path is short.
- The proposed method can reduce the energy consumption of UAV through effectively shortening the flight path.

The next section describes the related work. Section 3 gives an overview of the article. Section 4 introduces the partition traversal strategy. Section 5 describes the design and implementation of the 2-OptACO algorithm. Section 6 shows the performance comparison among 2-OptACO, GA and ACO.

2. RELATED WORK

After wireless signals were introduced into disaster rescue, early techniques used war-driving [3, 4] to locate trapped targets. However, war-driving is limited by the road conditions. For instance, after an earthquake, there may be no way in the scene of the disaster. At that time, the war-driving methods become less effective and less convenient.

With the maturing of UAV technology, using UAVs to locate trapped targets has entered the sight of the researchers. War-flying [12-14] makes a UAV fly in an open space, detect and accurately locate targets equipped with wireless devices [4, 15]. But it needs to traverse the target area according to a predetermined space filling curve to determine whether or not there is a target, so the flight path is relatively longer.

GuideLoc [7] is a UAV-assisted method to search and locate trapped persons, which can locate multiple targets in a target area one by one. Its accuracy is equivalent to that of GPS. GuideLoc locates a target through two steps. Firstly, UAV receives the angles of arrival (AOAs) and the received signal strength indicators (RSSIs) [16-26] sent by wireless devices that trapped targets carry, and estimates the positions of the trapped targets. Then it plans an optimal path according to the estimated result. Along the planned path, GuideLoc controls the UAV to fly towards the next target based on the AOA of the next target, and uses the RSSI of the next target to determine whether the UAV is just over the target. If the UAV is over the target, GuideLoc will take the UAV's GPS coordinates as the target location information to obtain accurate positioning. It is worth noting that GuideLoc can estimate the location of a target depending on the received signals sent by the target. Hence, the area in which GuideLoc can directly locate targets is limited (*i.e.*, a single-hop range). When the target area is larger than the range, the method adopts the solution of regional division to divide the target area into several unit partitions. Each unit partition is not larger than a single-hop range. Then, GuideLoc controls the UAV to visit the unit partitions one after the other and locates the targets in each partition. Obviously, the path of traversing unit partitions and the path of traversing multiple targets in a unit partition will have a direct impact on the efficiency of search and rescue and the energy consumption of UAV. Although the method uses the genetic algorithm for path planning and achieves a good result, there is still a possibility to improve its performance.

In this paper, we further discuss and study the path optimization algorithm used in GuideLoc, and propose a 2-OptACO method which is better than that used in GuideLoc.

3. OVERVIEW

In this paper, the path planning method in GuideLoc is improved. Firstly, we give a concise method to plan the path for traversing all the partitions in GuideLoc.

Then, a 2-OptACO method is proposed to plan the traversal path for multiple targets in a single partition.

The 2-OptACO method is based on the 2-opt algorithm to improve ACO, which is applied to the path optimization in using a UAV to search and rescue. The simulation results show that the 2-OptACO method has a faster convergence speed than GA and can obtain a better optimal solution.

In order to verify the performance of 2-OptACO, the method is compared with GA and ACO in this paper. To get a faster converge speed and a better path, 2-OptACO requires the following steps:

Step 1: Randomly generate simulation data to initialize the coordinates of all the nodes to be traversed.
Step 2: Set the number of ants, the number of iterations, the heuristic value, and other parameters related with the pheromone (*e.g.*, the evaporation factor). Initialize the pheromone value of each edge and each ant's tabu list.
Step 3: In accordance with the probability rules, each ant chooses the next node under the control of its tabu list until a corresponding path is formed.
Step 4: Use the 2-opt algorithm to respectively optimize the path generated by each ant and get a new path for each ant. Calculate the length of each new path. Path length is the sum of the length of each edge in the path.

Step 5: Update the pheromone value of each edge in each new path. The pheromone evaporation operation is firstly carried out for each edge, and then the deposited pheromone value of each edge is obtained according to the length of each new path.
Step 6: After the pheromone values of all the edges have been updated, record the current shortest new path and update the related parameters (*e.g.*, the tabu list of each ant), and then go to Step 3. Repeat the above steps until the termination condition (*e.g.*, the solution cannot be further improved or the predetermined iterations has been reached.) of the algorithm is satisfied.

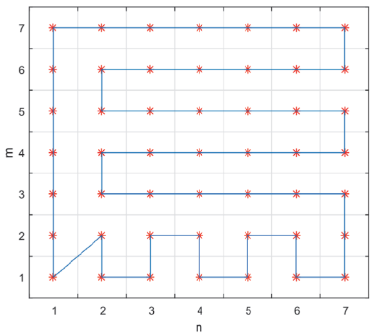
4. A SIMPLE STRATEGY FOR PARTITION TRAVERSAL

In order to solve the problem that the region of search is larger than the signal coverage of the device carried by a target, GuideLoc first divides the region of search into many unit partitions whose area is as large as the signal coverage of a device. Then, it locates every target in each partition one by one. Obviously, the sequence (*i.e.* the path) of traversing the partitions will affect the speed of localization and the energy consumption of the UAV. GuideLoc uses GA to optimize the path of traversing all the partitions. Although the optimal path can be found, the algorithm complexity of GA is relatively high. Therefore, a simple strategy is proposed in this paper, and it can give the optimal path directly.

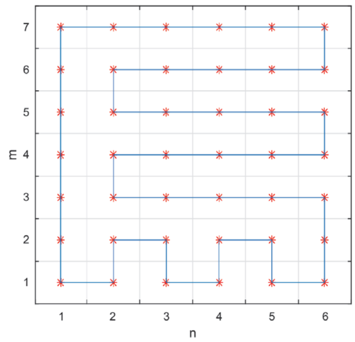
We assume that the region is divided into $m \times n$ same square partitions. For a square partition, it is obvious that the shortest path from one partition center to another is the length of a partition side when the two partitions are adjacent with a common side.

You might as well set partition (1, 1) as the starting position. Firstly, the UAV traverses partitions (j, 1) (j = 1, 2, 3, ..., m) in turn; Secondly, the UAV traverses partitions (m, i) (i = 2, 3, 4, ..., n) in turn; Thirdly, the UAV traverses partitions ($m-1$, i) (i = n, $n-1$, $n-2$, ..., 2) in turn; Fourthly, the UAV traverses partitions ($m-2$, i) (i = 2, 3, 4, ..., n), and so on. If m is even, whatever n is odd (Fig. 1 (c)) or even (Fig. 1 (d)), the last traversed partition will be (1, 2) which is just adjacent to partition (1, 1) with a common side, so that the UAV can return to partition (1, 1) with the shortest distance. If m is odd, when the third row has just been traversed and the UAV is being at partition (3, n), then the UAV will visit partitions (2, n), (1, n), (1, $n-1$), (2, $n-1$), (2, $n-2$), (1, $n-2$), (1, $n-3$), *etc.* one by one. If n is even, as shown in Fig. 1 (b), the last traversed partition will be also (1, 2), and the case is as same as that m is even. If n is odd, as shown in Fig. 1 (a), the last traversed partition will be (2, 2) which is just adjacent to partition (1, 1) with a common vertex, so that the UAV can only return to partition (1, 1) with a diagonal distance.

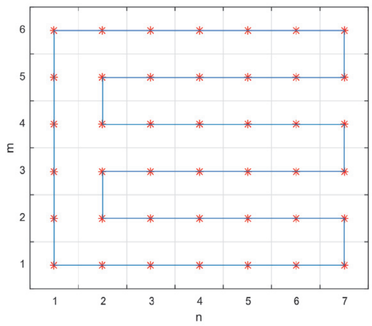
Using the above strategy to visit each partition, we can get the determinate shortest path no matter how large the search region is. But for GA, it is only to look for an approximate optimal path, and its efficiency is closely associated with the related parameters setting. The quality of the solution given by GA is instability. However, with the area of the search region increasing, the quality of our solution will not decrease. We have written a loop program to simulate the traversal with our strategy. Specific implementations are shown in Fig. 2.



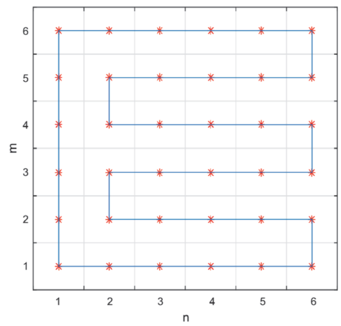
(a) m and n are all odd numbers (m=7, n=7).



(b) m is an odd number and n is an even number (m=7, n=6).



(c) m is an even number and n is an odd number (m=6, n=7).



(d) m and n are all even numbers (m=6, n=6).

Fig. 1. Influence of the parity of m and n on partition traversal.

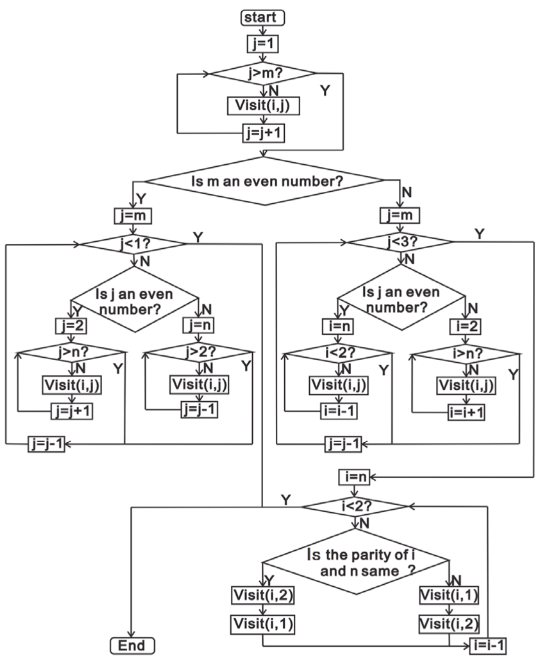


Fig. 2. Flow chart of the strategy of partition traversal.

We use this program to simulate the cases that m and n are respectively odd or even. Specifically, m and n are set in turn as following: $m = n = 7$, $m = n = 6$, $m = 7$ and $n = 6$, $m = 6$ and $n = 7$. In order to make the size of the figure well suited and the display of the figure clear in this paper, we use 6 as the example of even number and 7 as the example of odd number. The results are shown in Figs. 1 (a)-(d). In those experiments using other different even and odd numbers (even the numbers are very large), this strategy is also ok.

5. 2-OPTACO

5.1 Problem Analysis

In GuideLoc, the path planning scheme is mainly used in a unit partition which can be completely covered by the signal of the device carried by a target. According to the task requirements, the UAV will start from the center of a unit partition to visit all the target nodes in this partition, and finally arrive at the next unit partition [7]. This problem is similar to the classic traveling salesman problem (TSP) [27-30]. The difference between TSP and this problem is that TSP needs to return to the starting point, but this problem does not need to return. However, this problem can be transformed into a TSP problem by establishing an edge between the final point (*i.e.*, the center of the next unit partition) and the starting point (*i.e.*, the center of the current unit partition). It is worth to note that the path to be planned must include the added edge. So a negative weight would be set for this edge. Then the path can be found by solving the TSP problem and the edge having the negative weight will be removed from the path. At last, the remaining path is the solution of the problem.

Therefore, the solution of the problem boils down to resolving a TSP problem. It also belongs to a kind of NP complete problems (NPC) [31, 32]. The purpose of the TSP problem is to search for an optimal circle that traverses n points or an arrangement $\pi(X) = \{V_1, V_2, V_3, \ldots, V_n\}$ of a natural number subset $X = \{1, 2, 3, \ldots, n\}$ (the elements of the subset X refer to the serial numbers of the n points), which makes

$$\min T_d = \min\left\{\sum_{i=1}^{n-1} d\left(V_i, V_{i+1}\right) + d\left(V_n, V_1\right)\right\} \tag{1}$$

where $d(V_i, V_{i+1})$ indicates the distance between point V_i and point V_{i+1} and V_1 is the starting point.

5.2 The Theory of 2-OPTACO

From the previous section, we know that the multi-targets traversal problem in a unit partition can be easily transformed as a TSP problem to resolve. We simply replace every city in the TSP problem with the search target that is the trapped person. ACO is often used to solve TSP. In order to simulate the actual behavior of ants, we define these signs: m is the number of ants in the ant colony, n is the number of targets in a unit partition, d_{ij} $(i, j = 1, 2, 3, \ldots, n)$ is the distance between target i and target j, $b_i(t)$ is the number of the ants being at the target i at time t, then

$$m = \sum_{i=1}^{n} b_i(t) \tag{2}$$

$\tau_{ij}(t)$ is the amount of pheromones on the path between target i and target j at time t. At the beginning, the amount of pheromones on each path is equal, and $\tau_{ij}(0) = C(i, j = 1, 2, 3, \ldots, n, C$ is a constant). In the movement, ant $k(k = 1, 2, 3, \ldots, m)$ determines the movement direction according to the amount of pheromones on each path. $P^k_{ij}(t)$ is the probability that ant k transfers from target i to target j at time t:

$$P^k_{ij} = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}, j \notin tabu_k \\ 0, j \in tabu_k \end{cases} \tag{3}$$

where η_{ij} is the heuristic information for the ants transferring from target i to target j. And it is defined as $\eta_{ij} = \dfrac{1}{d_{ij}}$. α and β respectively indicate how important the residual pheromone and the heuristic information are for the path selection on the path $i\,j$. Different from the actual ant colony, the artificial ant colony system has a memory function, tabu list $tabu_k$ ($k = 1, 2, 3, \ldots, m$), to record the targets that ant k has already traversed until now. In fact, the tabu list records the targets that are not allowed to select in the next step. $tabu_k$ is dynamically adjusted with ant k advancing.

When all the ants have completed one traversal, their current tabu lists should be all full. Then the tabu list of each ant will be emptied, and the target where ant k stays will be filled into $tabu_k$ to be ready for the next traversal. The length of the path that ant k traveled, L_k, is calculated, and the shortest path L_{kmin} ($L_{kmin} = minL_k$, $k = 1, 2, 3, \ldots, m$) is saved.

In order to make 2-optACO converge faster than the classic ACO, we added local search optimization, 2-opt, into the classic ACO. The principle is as follows:

If the current shortest path L_{kmin} is $A \to B \to C \to D \to E \to F \to G \to H \to A$, we randomly divide the path into three segments, such as ($A \to B \to C$), ($D \to E \to F \to G$) and ($H \to A$). Then we turn the direction of the middle segment ($D \to E \to F \to G$) and get a new path segment ($G \to F \to E \to D$). We use it to replace the old middle segment of the path and get a new path $A \to B \to C \to G \to F \to E \to D \to H \to A$.

Lastly, we calculate the length of the new path, compare it with the length of the path L_{kmin}, and take the shorter path as the new path L_{kmin}. The procedure is repeated until that a path shorter than the current L_{kmin} cannot be found.

As time went by, the pheromones become less and less due to evaporation. Parameter ρ indicates the degree of pheromone evaporation. Once the ants have completed a cycle, the amount of pheromones on each path should be adjusted according to Eq. (4):

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

where $\Delta\tau^k_{ij}(t)$ is the pheromones delivered by ant k on the path ij in cycle t.

5.3 2-OptACO Algorithm

According to the theory in the previous section, we design the 2-OptACO algorithm:

Step 1: Initialize the pheromone value of each edge to a small same constant.
Step 2: Allocate m ants to n target nodes randomly, and correspondingly fill these target nodes into the ants' tabu lists.
Step 3: For each ant, calculate the selection probability of each optional target node according to Eq. (3), select the next target node in accordance with the calculated result and modify its tabu list.
Step 4: Choose the next target node and update the pheromone on the edge for each ant. And record the path that each ant has traveled.
Step 5: Determine whether every ant has visited all the target nodes? If yes, go to Step 6; if not, go to Step 3.
Step 6: When all the ants have visited all the target nodes, calculate the length of the path that each ant traveled and retain the shortest path.
Step 7: Use the 2-opt algorithm to optimize the retained shortest path.
Step 8: Update the pheromone values on the optimized shortest path.
Step 9: Are all the iterations finished? If yes, the algorithm ends; otherwise, empty the tabu list of each ant and return to Step 2.

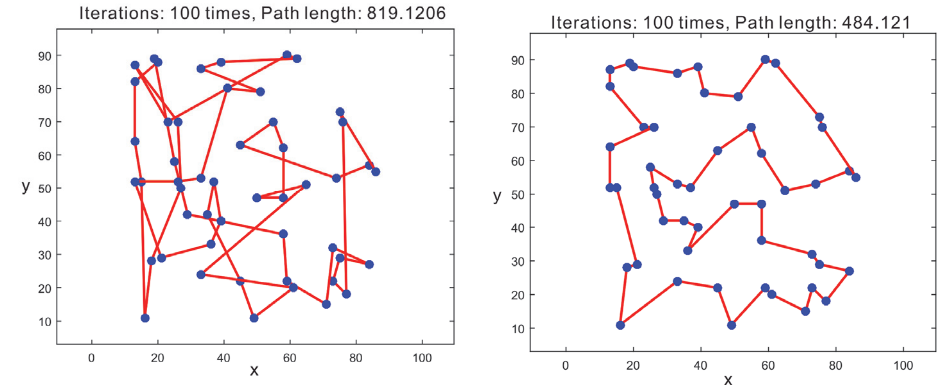
6. SIMULATION EXPERIMENT AND RESULT ANALYSIS

In this section, the simulation experiments were carried out with MATLAB. In the simulations, a number of coordinates were randomly generated as the search and rescue targets (also called "nodes" in the simulations). Then, the GA, classic ACO and 2-Opt-ACO algorithms were used respectively for path planning. We compared the lengths of the paths planned by the three algorithms and the algorithms' performance in convergence speed and in calculation time.

6.1 Comparison of GA and ACO in GuideLoc

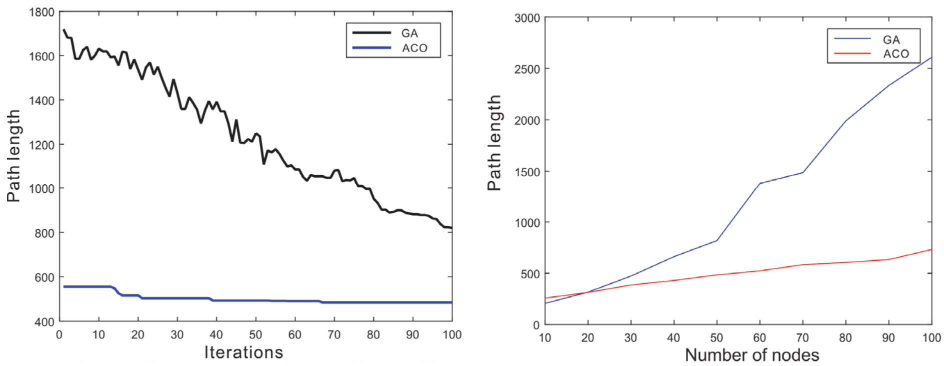
To verify that ACO is better than GA in path planning for GuideLoc, the same nodes were used by GA and ACO to plan the path in each simulation. And we simulated many times by using different numbers of nodes. The simulation results are explained as below.

In Fig. 3, we show the case that there are 50 target nodes. It is obviously that the path planed by ACO is better than that planed by GA. In the simulation, each algorithm performed 100 iterations. In terms of the computation speed, GA spent 22 seconds for the 100 iterations, and ACO only spent 12 seconds for the 100 iterations. The optimal path lengths gotten respectively by the two algorithms are 662.0707 (GA) and 429.9885 (ACO). In Fig. 4 (a), it can be seen that with the number of iterations increasing, the length of the path optimized by GA is shortened slowly, and the length of the path optimized by ACO is reduced quickly to a stable value. Moreover, the path length obtained by ACO is much shorter than that gotten by GA under the same iterations. Without loss of generality, we changed the number of nodes (from 10 to 100) and the corresponding coordinates, and then compared the performance of the two algorithms. The simulation result is shown in Fig. 4 (b). The comparison shows that with the increase of the number of target nodes, the difference between the lengths of the two paths planned by ACO and GA is getting larger and larger, and the path planned by ACO is becoming more and more shorter than that planned by GA. Therefore, the ACO algorithm is chosen for path planning. Combined with the 2-Opt algorithm, the ACO algorithm is improved, and a better algorithm, 2-OptACO, is designed.



(a) Path planned by GA. (b) Path planned by ACO.

Fig. 3. Paths planned by ACO and GA under 50 nodes.

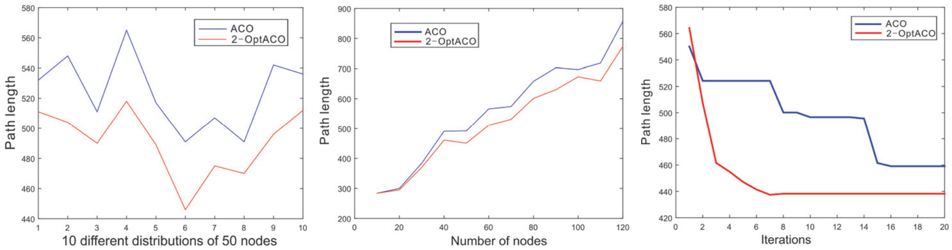


(a) Comparison of path lengths under different iterations. (b) Comparison of path lengths under different numbers of nodes.

Fig. 4. Comparisons of path lengths gotten by GA and ACO under different conditions.

6.2 Comparison of ACO and 2-OptACO in GuideLoc

In this subsection, we compared the lengths of the two paths planned by the ACO and 2-OptACO algorithms, the convergence speed and the calculation time of the two algorithms.



(a) Comparison of path lengths under 10 different distributions of 50 nodes.

(b) Comparison of path lengths under different numbers of nodes.

(c) Comparison of path lengths under different iterations.

Fig. 5. Comparisons of path lengths gotten by 2-OptACO and ACO under different conditions.

In Fig. 5 (a), we also show the case that there are 50 target nodes. Under the 10 different distributions of nodes, the lengths of the two paths planned respectively by 2-OptACO and ACO are compared. It can be seen that the length of the path planned by 2-OptACO is always shorter than that planned by ACO. Without loss of generality, we changed the number of nodes (from 10 to 120), and then compared the performance of the two algorithms. The simulation result is shown in Fig. 5 (b). The comparison shows that with the increase of the number of nodes, the path planned by 2-OptACO is always shorter than that planned by ACO. In Fig. 5 (c), we compared the lengths of the two paths planned by 2-OptACO and ACO under the different iterations. As shown in the Figure, through 7 iterations, 2-OptACO almost converges to the optimum value. But ACO achieves a stable value when the iterations is about 16, and the optimal path of ACO is not better than that of 2-OptACO. So the convergence speed of 2-OptACO is faster.

Fig. 6 shows the computation time spent by ACO and 2-OptACO in path planning under the different number of target nodes and the same iterations. It can be seen that the time spent by 2-OptACO is less than that spent by ACO in path planning.

In summary, according to the simulation experiments, the path planned by 2-OptACO is shorter than that planned by GA or ACO. The convergence speed of 2-OptACO is faster than that of GA or ACO, and the computation time of 2-OptACO is shorter than that of GA or ACO.

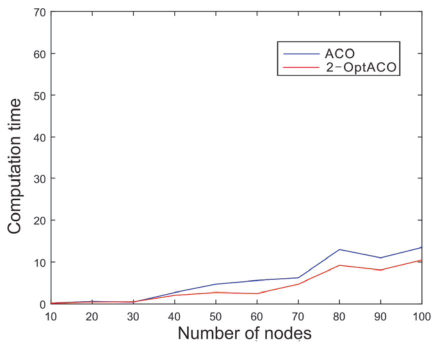


Fig. 6. Comparison of computation time spent by 2-OptACO and ACO under different numbers of nodes.

7. CONCLUDING REMARKS

In this paper, we further discussed and studied the path optimization algorithm used in GuideLoc and proposed a 2-OptACO method. The simulation results show that2-OptACO has a faster convergence speed and can get a better flight path than GA in GuideLoc and ACO.

ACKNOWLEDGMENT

This work was Supported in part by Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No. 17JK0775) and the National Natural Science Foundation of China (No. 61602382, No. 61501372) and the Natural Science Basic Research Plan in Shaanxi Province of China (No. 2017GY-191) and the Science Foundation of Northwest University (No. 15NW31 and No. 15NW32).

REFERENCES

1. A. Purohit and P. Zhang, “Sensorfly: a controlled-mobile aerial sensor network,” in *Proceedings of ACM Conference on Embedded Networked Sensor Systems*, 2009, pp. 327-328.
2. S. Verykokou, A. Doulamis, G. Athanasiou, C. Ioannidis, and A. Amditis, “UAV-based 3D modelling of disaster scenes for urban search and rescue,” in *Proceedings of IEEE International Conference on Imaging Systems and Techniques*, 2016, pp. 106-111.
3. Wikipedia, “Wardriving and warwalking,” http://en.wikipedia.org/wiki/Wardriving, 2013-03-06.
4. A. Subramanian, P. Deshpande, J. Gaojgao, and S. Das, “Drive-by localization of roadside wifi networks,” in *Proceedings of the 27th IEEE Conference on Computer Communications*, 2008, pp. 718-725.
5. A. Purohit, Z. Sun, F. Mokaya, and P. Zhang, “Sensorfly: Controlled-mobile sensing platform for indoor emergency response applications,” in *Proceedings of International Conference on Information Processing in Sensor Networks*, 2011, pp. 2554-2584.
6. N. H. Motlagh, T. Taleb, and O. Arouk, “Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives,” *IEEE Internet of Things Journal*, Vol. 3, 2016, pp. 899-922.
7. A. Wang, X. Ji, D. Wu, X. Bai, N. Ding, J. Pang, S. Chen, X. Chen, and D. Fang, “Guideloc: UAV-assisted multitarget localization system for disaster rescue,” *Mobile Information Systems*, Vol. 2017, 2017, pp. 1-13.
8. M. Englert, H. Roglin, and B. Vocking, “Smoothed analysis of the 2-Opt algorithm for the general TSP,” *ACM Transactions on Algorithms*, Vol. 13, 2016, pp. 1-15.
9. C. F. Tsai and C. Yao, “Enhancement of traveling salesman problem by hybridization of greedy and 2-opt techniques,” in *Proceedings of International Conference on Machine Learning and Cybernetics*, 2017, pp. 564-569.

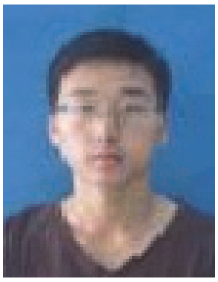
10. I. Alaya, C. Solnon, and K. Ghedira, "Ant colony optimization for multi-objective optimization problems," in *Proceedings of IEEE International Conference on TOOLS with Artificial Intelligence*, 2017, pp. 450-457.
11. O. Deepa and A. Senthilkumar, "Swarm intelligence from natural to artificial systems: Ant colony optimization warflying," *International Journal on Applications of Graph Theory in Wireless Ad Hoc Networks and Sensor*, Vol. 8, 2016, pp. 9-17.
12. Wikipedia, "Warflying," http://en.wikipedia.org/wiki/Warflying, 2013-06-12.
13. R. Queen, "Wireless aerial surveillance platform," https://rabbit-hole.org/, 2014-01-12.
14. Z. Liu, Y. Chen, B. Liu, C. Cao, and X. Fu, "Hawk: An unmanned mini-helicopter-based aerial wireless kit for localization," *IEEE Transactions on Mobile Computing*, Vol. 13, 2014, pp. 287-298.
15. F. Comtech, "802.11 phocus array antenna system by fidelity comtech," http://www.fidelity-comtech.com/, 2014-01-04.
16. P. Gimenez-Febrer, A. Pages-Zamora, S. S. Pereira, and R. Lopez-Valcarce, "Distributed AOA-based source positioning in NLOS with sensor networks," in *Proceedings of IEEE International Conference on Acoustics*, *Speech and Signal Processing*, 2015, pp. 3197-3201.
17. J. Wang, X. Chen, D. Fang, C. Q. Wu, Z. Yang, and T. Xing, "Transferring compressive-sensing-based device-free localization across target diversity," *IEEE Transactions on Industrial Electronics*, Vol. 62, 2015, pp. 2397-2409.
18. L. Chen, X. Chen, L. Ni, Y. Peng, and D. Fang, "Human behavior recognition using Wi-Fi CSI: Challenges and opportunities," *IEEE Communications Magazine*, Vol. 55, 2017, pp. 112-117.
19. J. Wang, D. Fang, Z. Yang, H. Jiang, X. Chen, T. Xing, and L. Cai, "E-HIPA: An energy-efficient framework for high-precision multi-target-adaptive device-free localization," *IEEE Transactions on Mobile Computing*, Vol. 16, 2017, pp. 716-729.
20. J. Wang, J. Xiong, H. Jiang, X. Chen, and D. Fang, "D-Watch: Embracing "bad" multipaths for device-free localization with COTS RFID devices," *IEEE/ACM Transactions on Networking*, Vol. 25, 2016, pp. 3559-3527.
21. L. Chang, X. Chen, Y. Wang, D. Fang, J. Wang, T. Xing, and Z. Tang, "FitLoc: Fine-grained and low-cost device-free localization for multiple targets over various areas," *IEEE/ACM Transactions on Networking*, Vol. 25, 2017, pp. 1994-2007.
22. M. Tanbo, R. Nojiri, Y. Kawakita, and H. Ichikawa, "Active RFID attached object clustering method based on RSSI series for finding lost objects," in *Proceedings of IEEE 2nd World Forum on Internet of Things*, 2016, pp. 363-368.
23. C. Liu, D. Fang, Z. Yang, H. Jiang, X. Chen, W. Wang, T. Xing, and L. Cai, "Rss distribution-based passive localization and its application in sensor networks," *IEEE Transactions on Wireless Communications*, Vol. 15, 2016, pp. 2883-2895.
24. Y. Zou, W. Liu, K. Wu, and L. M. Ni, "Wi-Fi radar: Recognizing human behavior with commodity Wi-Fi," *IEEE Communications Magazine*, Vol. 55, 2017, pp. 105-111.
25. Y. Zou, J. Xiao, J. Han, K. Wu, Y. Li, and L. M. Ni, "GRfid: A device-free RFID-based gesture recognition system," *IEEE Transactions on Mobile Computing*, Vol. 16, 2017, pp. 381-393.

26. G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni, "We can hear you with Wi-Fi!," *IEEE Transactions on Mobile Computing*, Vol. 15, 2016, pp. 2907-2920.
27. J. Yang, R. Ding, Y. Zhang, M. Cong, F. Wang, and G. Tang, "An improved ant colony optimization (I-ACO) method for the quasi travelling salesman problem (Quasi-TSP)," *International Journal of Geographical Information Science*, Vol. 29, 2015, pp. 1534-1551.
28. N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Social Science Electronic Publishing*, 2016, pp. 1-40.
29. L. Liu, Y. Song, H. Zhang, H. Ma, and A. V. Vasilakos, "Physarum optimization: A biology-inspired algorithm for the Steiner tree problem in networks," *IEEE Transactions on Computers*, Vol. 64, 2015, pp. 818-831.
30. Y. Song, L. Liu, H. Ma, and A. V. Vasilakos, "A biology-based algorithm to minimal exposure problem of wireless sensor networks," *IEEE Transactions on Network and Service Management*, Vol. 11, 2014, pp. 417-430.
31. B. Reus, "How to solve NP-complete problems," *Limits of Computation*, Springer, Switzerland, 2016, pp. 275-297.
32. Y. Liu, C. Gao, Z. Zhang, Y. Lu, S. Chen, M. Liang, and T. Li, "Solving NP-hard problems with physarum-based ant colony system," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 14, 2017, p. 108.

Xiang Ji (姬翔) received the B.S. degree in Computer Application from Northwest University, Xi'an, China, in 2001, received the M.S. degree in Computer Application Technology from Xidian University, Xi'an, China, in 2005. She is studying for a Ph.D. in Northwest University. Her main research interests include wireless sensor network and human-computer interaction.

Qing-Yi Hua (华庆一) received the Ph.D. degree in Computer Application Technology from Northwestern Polytechnical University, Xi'an, China. He is currently a Professor with the School of Information Science and Technology, Northwest University, Xi'an. His current research interest is human-computer interaction.

An-Wen Wang (王安文) received the B.S. degree in Computer Science and Technology from Northwest University, Xi'an, China, in 2001. He received the M.S. degree in Computer Software and Theory from Shannxi Normal University, Xi'an, China, in 2006. He is currently a Lecturer with the School of Information Science and Technology, Northwest University. His research interests include wireless sensor networks and embedded system.



Jun-Song Tang (汤俊松) received the B.S. degree in Computer Science and Technology from Northwest University, Xi'an, China, in 2016. His current research interests include network and information security.

Chun-Yu Li (李春雨) is a student in Computer Science and Technology from Northwest University, Xi'an, China, in 2016.

Feng Chen (陈峰) received the M.S. degree in Computer Science and Technology from Northwest University, Xi'an, China. His current research interests include network and information security.

Ding-Yi Fang (房鼎益) received the Ph.D. degree in Computer Application Technology from Northwestern Polytechnical University, Xi'an, China, in 2001. He is currently a Professor with the School of Information Science and Technology, Northwest University, Xi'an. His current research interests include mobile computing and distributed computing systems, network and information security, and wireless sensor networks.