

A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization

Hui Yang, Jie Qi, Yongchun Miao, Haixin Sun, *Member, IEEE*, and Jianghui Li, *Member, IEEE*

Abstract—This paper presents an efficient double-layer ant colony algorithm, called DL-ACO, for autonomous robot navigation. This DL-ACO consists of two ant colony algorithms which run independently and successively. First, a parallel elite ant colony optimization (PEACO) method is proposed to generate an initial collision-free path in a complex map, and then we apply a path improvement algorithm called turning point optimization algorithm (TPOA), in which the initial path is optimized in terms of length, smoothness and safety. Besides, a piecewise B-spline path smoother is presented for easier tracking control of the mobile robot. Our method is tested by simulations and compared with other path planning algorithms. The results show that our method can generate better collision-free path efficiently and consistently, which demonstrates the effectiveness of the proposed algorithm. Furthermore, its performance is validated by experiments in indoor and outdoor environments.

Index Terms—Double-layer ant colony optimization (DL-ACO), path planning, trajectory optimization, piecewise B-spline curve.

I. INTRODUCTION

GLOBAL path planning is a significant issue in robotics because of its various applications, such as autonomous driving, civilian search, emergency rescue, resource exploitation and so forth [1]–[3]. This problem is concerned about how a mobile robot can find a path from a start position to a target position, avoiding collision against obstacles and satisfying constraints. Traditional approaches, such as artificial potential field [4] and cell decomposition [5], are widely applied in various tasks and demonstrated remarkable performance thanks to their easy execution. Nevertheless, they tend to consume expensive computation and easily fall into

traps in complex problems. Comparing to traditional methods, heuristic algorithms have been proven to be efficient in robot path planning [6], including neural network [7], fuzzy logic technique [8] and nature spired algorithms such as genetic algorithm (GA) [9], particle swarm optimization (PSO) [10] and ant colony algorithm (ACO) [11]. The global searching ability of a good path planning method should be strong, as well as characterized by stability. However, in some heuristic algorithms such as PSO and GA, the initial values are all obtained randomly, this randomness sometimes leads to a great variation of results. Fig. 1 depicts three widely different routes generated by the same algorithm. This uncertainty seriously affects the robustness of path planner in practical applications.

Ant colony optimization has been recognized as an efficient and robust optimization technique and has been used frequently in different applications [12]–[15]. Studies using ACO to solve the global path planning problem have been carried out in recent years. Duan *et al.* [16] proposed a path planning method based on ACO and differential evolution for three-dimension path planning of unmanned combat aerial vehicle (UCAV). An improved ACO for robot path planning in a dynamic environment by adding a fuzzy cost function in path evaluation was introduced in [17]. Zhu *et al.* [18] presented an algorithm based on dynamic path re-computation and an improved scout ant algorithm for robot navigation in unknown environments. Although ant colony algorithm reflect good search feature, it has shortcomings of low convergence rate and premature convergence. Furthermore, the maximum distance an ant can travel in a single movement is limited, which results in twists and turns on the path and affects the path length and smoothness.

Researchers put forward some improved methods to modify the shortcomings of ant colony optimization. Abdulkader *et al.* [19] put forward a hybridized ant colony algorithm, which combines local search with an existent ant colony algorithm for less computational time and high performance in larger problems. Jiao *et al.* [20] designed an adaptive polymorphic ant colony algorithm by employing the adaptive state transition strategy and the adaptive information updating strategy, which determines the optimal combination parameters in accordance to actual situation. Chen *et al.* [21] introduced a two-stage ant algorithm. With the raid of this method, the heuristic search is split into preprocess stage and path planning stage to avoid the algorithm falling into local minimum. Parallel computing and elitist strategy have been widely used to improve the

Manuscript received June 27, 2018; revised Aug 27, 2018 and Nov 11 2018; accepted Dec 2, 2018. This work was supported by the National Natural Science Foundation of China (61671394, 61471309), the Fundamental Research Funds for the Central Universities (20720170044) and Science and Technology Program of Shenzhen, China (JSGG20170414090428464). (Corresponding author: Haixin Sun)

Hui Yang and Jie Qi are with the College of Electronic Science and Technology, Xiamen University, Xiamen, Fujian, 361005 China (e-mail: hyang@stu.xmu.edu.cn and qjie@xmu.edu.cn)

Yongchun Miao and Haixin Sun are with the School of Information Science and Engineering, Xiamen University, Xiamen, Fujian, 361005 China (e-mail: ycmiao@stu.xmu.edu.cn and hxsun@xmu.edu.cn).

Jianghui Li is with the Engineering, University of Southampton, Highfield, Southampton (e-mail: J.Li@soton.ac.uk).

performance of algorithms [22]–[25]. Two parallel ant populations can widen the searching diversity, and strengthening the influence of elite ant can speed up the convergence rate.

In this work, the path planning problem is divided into two parts, path generation and trajectory optimization. We propose a double-layer ant colony algorithm which contains two ant algorithms to solve these two parts respectively. First the parallel elite ant colony optimization is presented for generating an initial feasible path. According to the twists and turns, the turning point optimization algorithm is proposed to make further improvement on the initial path. This type of double-layer computing structure improves the quality of solutions and strengthens the stability.

When a mobile robot tracking a path composed by a sequence of line segments, it must stop at each corner to reorient its steering wheels. This jerky motion prolongs the travelling time and shortens the service life of its mechanical devices. Besides, running on a smooth path reduces the robot's energy consumption [26]. Hence these discontinuous segments must be smoothed. B-spline curve is one of the most efficient curve interpolations and has been widely applied in many disciplines, such as medical imaging [27], geometric modeling [28], surface reconstruction [29] and position control [30]. With the properties of the B-spline curve, this interpolation scheme is practically useful for path smoothing [31], [32]. However, the fitting degree between the B-spline curve and the original segments does not meet the requirements in some cases, which may affect the path safety. Thus, we improve the conventional B-spline curve and design a piecewise B-spline curve, which only smooths the path around the corner. Finally, the DL-ACO and the piecewise B-spline path smoother are combined to form a complete path planning scheme for robot navigation. The main contributions of this paper can be summarized as follows.

- 1) An enhancement function and a parallel computing structure is designed for ant colony optimization to widen the searching diversity and avoid premature convergence.
- 2) A trajectory optimization method is put forward to improve the route in terms of both length and smoothness.
- 3) A piecewise B-spline path smoother is proposed to smooth the path around the corner without affecting the safety.
- 4) The proposed DL-ACO generates a shorter and smoother path in comparison to other heuristic algorithms. Furthermore, the double-layer structure strengthens the stability of our algorithm.

The rest of this paper is organized as follows. The path planning problem in the grid-based map is defined in Section

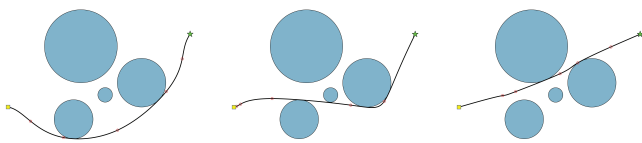


Fig. 1. Three routes generated by the same algorithm. (Square) Start position. (Star) goal position.

II. Section III describes the execution steps of PEACO. Section IV presents the trajectory optimization method including a turning point optimization algorithm and a piecewise B-spline path smoother. Computer simulations and results of real experiments are presented in Section V. Section VI concludes the paper.

II. ENVIRONMENT DESCRIPTION AND PROBLEM STATEMENTS

A. Environment Description

The working environment for mobile robot is a grid-based two-dimensional field with several static obstacles located in it. Obstacles have no influential negative attributes to free space and their location is completely known. Hence the grids are divided into free space grids \mathcal{T}^f and obstacle grids \mathcal{T}^o . The mobile robot is regarded as a mass point R and moves at a fixed speed.

High-risk area: In view of the safety, robot should keep a safe distance with obstacles to avoid accidents. As shown in Fig. 2, given an obstacle grid \mathcal{T}^o , the grids in four directions of \mathcal{T}^o are defined as high-risk grids \mathcal{T}^h . Robots will try to avoid this area, but if necessary (e.g. narrow channel), they are allowed to pass through it.

B. Problem Statements

Here we present the problem statements considered in this work. The grid-based path planning problem is defined as follows.

Given a start grid S and a goal grid E , find a set of grids $\pi(t) : [0, t] \rightarrow \mathcal{T}^f$ such that $\pi(0) = S$ and $\pi(t) = E$, the path is represented with the grids in $\pi(t)$ that are sequentially connected to each other from S to E , and these segments cannot collide with any \mathcal{T}^o . Then optimize the path in terms of length, smoothness and safety. Since there are several criteria to be optimized, the global path planning problem is categorized as a multi-objective optimization problem.

III. PARALLEL ELITE ANT COLONY OPTIMIZATION

To generate an initial collision-free path in a complex environment, a Parallel Elite Ant Colony Optimization (PEACO)

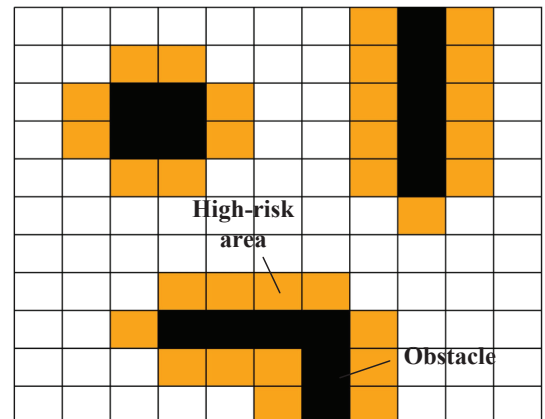


Fig. 2. Illustration of High-risk area.

is proposed. Different from conventional ACOs, the ant colony in PEACO is divided into two subpopulations and each subpopulation iterates independently. An enhancement function is proposed to strengthen the elitist ant at the step of pheromone updating. In addition, when the iteration process of each subpopulation ends, a pheromone interaction step between these two subpopulations is then added. The pseudo code for PEACO is given in Algorithm 1.

Algorithm 1 Parallel Elite Ant Colony Optimization

```

1: procedure PEACO
2:   Divide the ant colony into two subpopulation
3:   for each subpopulation do
4:     initialize number of ants  $m$ , maximum iteration
       number  $N_{max}$ , weights  $\alpha, \beta, \rho$ , start  $S$ , goal  $E$ 
5:     calculate heuristic information  $\tau_j$ 
6:     for  $N = 1$  to  $N_{max}$  do
7:       Put all ants into  $S$ 
8:       while ant  $k$  is not in  $E$  do
9:          $\psi \leftarrow$  the set of reachable grids for  $k$ 
10:        choose the next grid by (2)
11:      end while
12:      if all ants have arrived  $E$  then
13:         $L_k \leftarrow$  length of ant  $k$ 's path
14:         $L \leftarrow$  the best path in all iterations
15:        select the best ant  $b$ 
16:        while  $L_b < L$  do
17:           $q \leftarrow$  enhancement factor by (1)
18:           $\tau_b = q \cdot \tau_b$ 
19:           $L = L_b$ 
20:        end while
21:        update the pheromone by (4), (5), (6), (7)
22:      end if
23:    end for
24:  end for
25:  output the best path
26: end procedure

```

A. Enhancement function

In foraging, as soon as an ant finds the food source, it starts the return trip and deposits pheromones on the path that they have passed to guide the following ants. This forms a positive feedback loop that all the ants in the colony ultimately follow the optimal path to the food source. Meanwhile, the pheromone intensity reduces over the time which is called the pheromone evaporation, it raises the possibility that ants can find a better path instead of insisting on the existing path.

However, when a large number of ants run on the same route, the accumulation rate of pheromone will be much higher than the evaporation rate. This results in high pheromone concentrate on the existing path. Under this circumstance, even though an ant finds a new path, the pheromone that it releases on this new path is still far less than these on the existing path. Thus, we propose an enhancement factor as following:

$$q = \begin{cases} e^{\frac{t-1}{n}} - 1, & \text{if } L_b < L \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

TABLE I
VARIABLES OF PEACO.

Variable	Description
m	Number of ants
N_{max}	Maximum number of iterations
α	Weight of pheromone
β	Weight of heuristic information
ρ	Pheromone evaporation ratio
q	Enhancement factor
τ_{ij}	Pheromone on the path between i and j
η_j	Heuristic information on j

where t is the current iteration number, n is a constant that depends on the maximum iteration number to ensure that q is not overly large. Such that q is approximately 0 in early iterations, and the optimal solution in early iterations will not be strengthened. L_b is the length of the best path in the current iteration while L is the optimal solution up to the last iteration. That is, we enhance it only if a new optimal solution appears.

B. Process of PEACO

PEACO is comprised of three steps: initialization, selection and pheromone update. Each of these steps is discussed below:

- 1) *Initialization*: At this step, ant colony is divided into two subpopulations. Several parameters in each subpopulation are initialized and all of them are explained in Table I. The value of these weights should be assigned via experiments. Besides, the start S and the goal E are decided and all ants are located in the start grid. Two subpopulations carry out each iteration independently.
- 2) *Selection*: During their tour, ants use the roulette wheel selection to choose the next grid. The transition probability P_{ij}^k for ant k moving from grid i to grid j is calculated as

$$P_{ij}^k = \frac{\tau_{ij}^\alpha \eta_j^\beta}{\sum_{l \in \psi} \tau_{il}^\alpha \eta_l^\beta}, \quad l \in \psi. \quad (2)$$

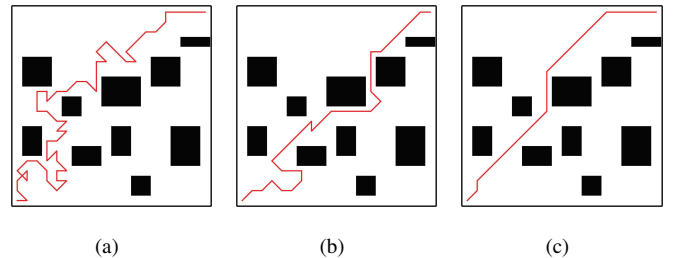


Fig. 3. Path generated by PEACO. (a) 5 iterations. (b) 15 iterations. (c) 30 iterations.

While ψ is the set of reachable grids, η_j is the heuristic information:

$$\eta_j = \frac{1}{d_{je}}, \quad (3)$$

where d_{je} is the distance between grid j and goal grid E . In addition, if an ant has passed the predefined maximum number of grids but still does not reach the goal, we let the ant drop.

- 3) *Pheromone update*: This step starts after all ants reach the goal. The best ant in two subpopulations called ant b is selected and two operations are conducted at the same time, pheromone reinforcement and pheromone evaporation. The rule is given by

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}(t) + q \cdot \Delta\tau_{ij}^b(t), \quad (4)$$

and

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t), \quad (5)$$

where $\Delta\tau_{ij}^k(t)$ is the quantity of pheromone deposited by ant k on the path between i and j at time t . $\Delta\tau_{ij}^b$ is the quantity of the extra pheromone deposited by ant b , $\Delta\tau_{ij}^k(t)$ is given by

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{if ant } k \text{ passes grid } i \text{ and } j \\ 0, & \text{otherwise} \end{cases}, \quad (6)$$

where Q is a constant. L_k is the total length of the path that ant k travels as

$$L_k = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \quad (7)$$

where x, y are the coordinates.

When the iteration number reaches the predefined N_{max} , the best path in two subpopulations will be chosen and outputted. Fig. 3 depicts the optimization results after different iterations in PEACO.

C. Sensitivity Analysis

In this subsection, we conduct a sensitivity analysis of α and β , which may affect the performance of our algorithm. We keep one parameter fixed, change another and record the changes in path length. The map in Fig. 3 is used for testing. Fig. 4 depicts the results. It can be seen that our algorithm performs good when $\alpha \in (1, 3)$, $\beta = 5$, and $\alpha = 1, \beta \in (2, 5)$. To sum up, when β is 2 to 5 times α , our algorithm can provide a good solution, and the solution quality is stable when the parameters are in this certain range. However, when the parameters are out of range, the solution quality will be affected.

IV. TRAJECTORY OPTIMIZATION

This section aims to improving the initial path generated by PEACO, not only optimize the path itself, but also reduce the difficulty that robot steering to follow the predetermined path. The Turning Point Optimization Algorithm (TPOA) is presented to reduce the number of turns and shorten the overall path length, while the piecewise B-spline path smoother is for smoothing the path around the turns. These two steps are conducted successively.

A. Turning Point Optimization Algorithm

As shown in Fig. 5, $S \rightarrow I_1 \rightarrow A \rightarrow B \rightarrow I_2 \rightarrow E$ is a feasible path from start to goal. It is obviously that this path is not the best one. For instance, an ant on S can go straight to A without passing through I_1 , similarly B to E without I_2 . If we remove I_1 and I_2 , the path $S \rightarrow A \rightarrow B \rightarrow E$ is shorter in length, as well as it has less turns. These points I_1, I_2 are defined as unnecessary turning points. For a complicated route with numerous turns, removing these unnecessary turning points can effectively improve it in terms of length and smoothness. Hence the Turning Point Optimization Algorithm is proposed as follows.

For the initial path, the set of nodes are comprised of the start, the goal and all turning points as

$$T = \{S, T_1, T_2, \dots, T_n, E\}. \quad (8)$$

In view of the orientation of the route, ants are only allowed to travel in a single direction. Thus, when an ant reaches node T_i , the set of following nodes is

$$T(i) = \{T_{i+1}, T_{i+2}, \dots, T_n, E\}. \quad (9)$$

We add an extra step that judges whether the ant on one node can reach another directly. If the straight line between node T_a and T_b does not pass through any obstacle grids \mathcal{T}^o , we believe that the ant on T_a can go straight to T_b . As following

$$con(T_a, T_b) = \begin{cases} 1, & \text{if } T_a \text{ can go straight to } T_b \\ 0, & \text{otherwise} \end{cases}, \quad (10)$$

where T_a and T_b are any two nodes on the path, To sum up, the set of reachable nodes $T_r(i)$ of an ant on node T_i is defined as

$$T_r(i) = \{T | T \in T(i) \cap con(T_i, T) = 1\}. \quad (11)$$

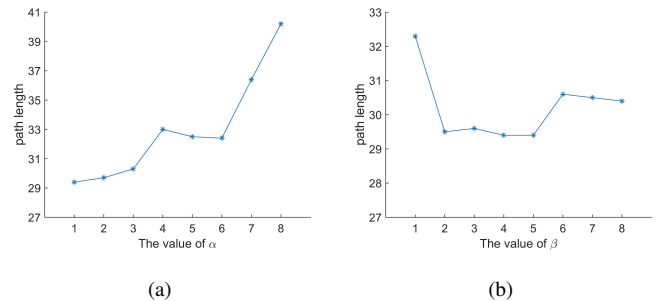


Fig. 4. Sensitivity Analysis of α and β . (a) α is variable, $\beta = 5$. (b) $\alpha = 1$, β is variable.

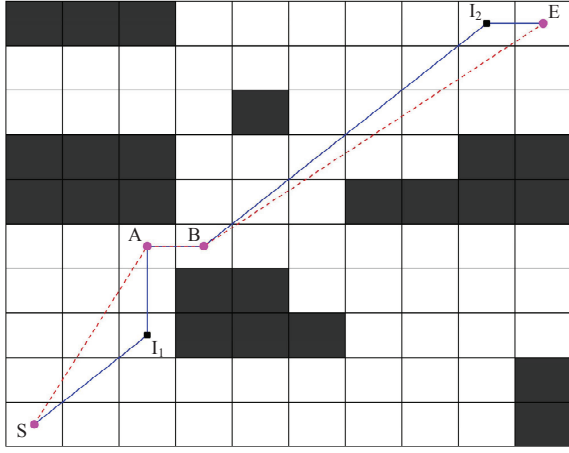


Fig. 5. The original collision-free path and the refined one on which the unnecessary turning points are removed.

Algorithm 2 Turning Point Optimization Algorithm

```

1: procedure TPOA
2:   input the coordinates of nodes T
3:    $n \leftarrow$  number of nodes on the path
4:   initialize parameters
5:   for  $a = 1$  to  $n - 1$  do
6:     for  $b = i + 1$  to  $n$  do
7:       calculate  $con(a, b)$  by (10)
8:     end for
9:   end for
10:  for  $N = 1$  to  $N_{max}$  do
11:    put all ants into  $S$ 
12:    while ant  $k$  is on node  $T_i$  do
13:      select the set of reachable nodes  $T_r(i)$  by (11)
14:      choose the next grid by (12)
15:    end while
16:    if all ants have arrived  $E$  then
17:      update the pheromone by (14), (15)
18:    end if
19:  end for
20:  select and output the best path
21: end procedure
    
```

The probability that an ant in node T_a goes to node T_b is given by

$$P_{ab}^k = \frac{\tau_{ab}^\alpha \eta_{ab}^\beta}{\sum_{l \in T_r(i)} \tau_{al}^\alpha \eta_{al}^\beta}, \quad l \in T_r(i), \quad (12)$$

where τ_{ab} is the pheromone concentration, η_{ab} is the heuristic information which is given by

$$\eta_{ab} = d_{ab}, \quad (13)$$

where d_{ab} is the distance between T_a and T_b . In the pheromone updating, we consider four parameters to evaluate a path including path length, number of turning points, path smoothness and path safety as

$$\tau_{ab}(t+1) = (1-\rho)\tau_{ab}(t) + \rho \cdot \sum_{k=1}^m \tau_{ab}^k(t), \quad (14)$$

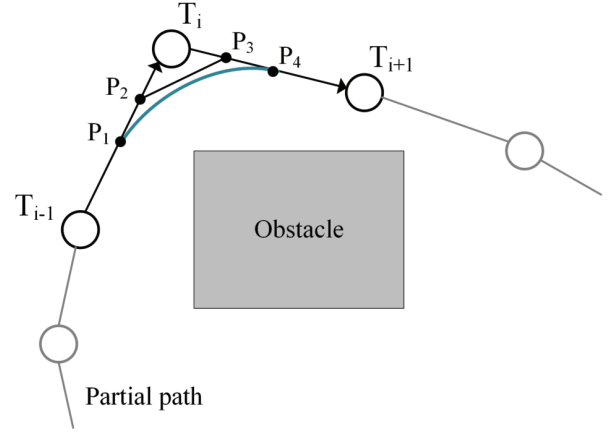


Fig. 6. Illustration of $T_{i-1} \rightarrow T_i \rightarrow T_{i+1}$ smoothed by piecewise B-spline curve.

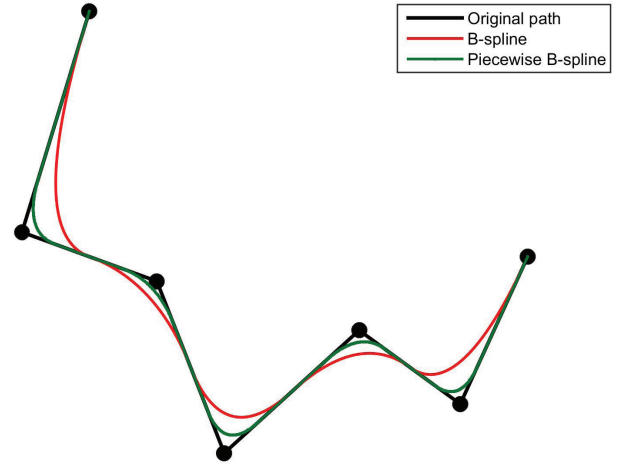


Fig. 7. Illustration of the same path smoothed by conventional B-spline curve and piecewise B-spline curve separately.

$$\Delta\tau_{ab}^k(t) = \frac{Q}{L_k + n + sm_k + sf_k}, \quad (15)$$

where n is the number of turning points, L_k is the length of the path. sf_k is the number of high-risk area grids the ant has passed. sm_k is the path smoothness:

$$sm_k = \sum_{i=2}^{n-1} \text{abs}(\theta_{i+1} - \theta_i) * 180/\pi, \quad (16)$$

with

$$\theta_{i+1} = \text{atan}[(y_{i+1} - y_i) / (x_{i+1} - x_i)], \quad (17)$$

$$\theta_i = \text{atan}[(y_i - y_{i-1}) / (x_i - x_{i-1})]. \quad (18)$$

The pseudo code for TPOA is given in Algorithm 2.

B. Piecewise B-spline Path Smoother

This subsection aims to smooth the path around the turning points. B-spline curve has been the most commonly used in the smoothness of polyline. However, in some cases (e.g. high steering angle), the path smoothed by B-spline does not fit the

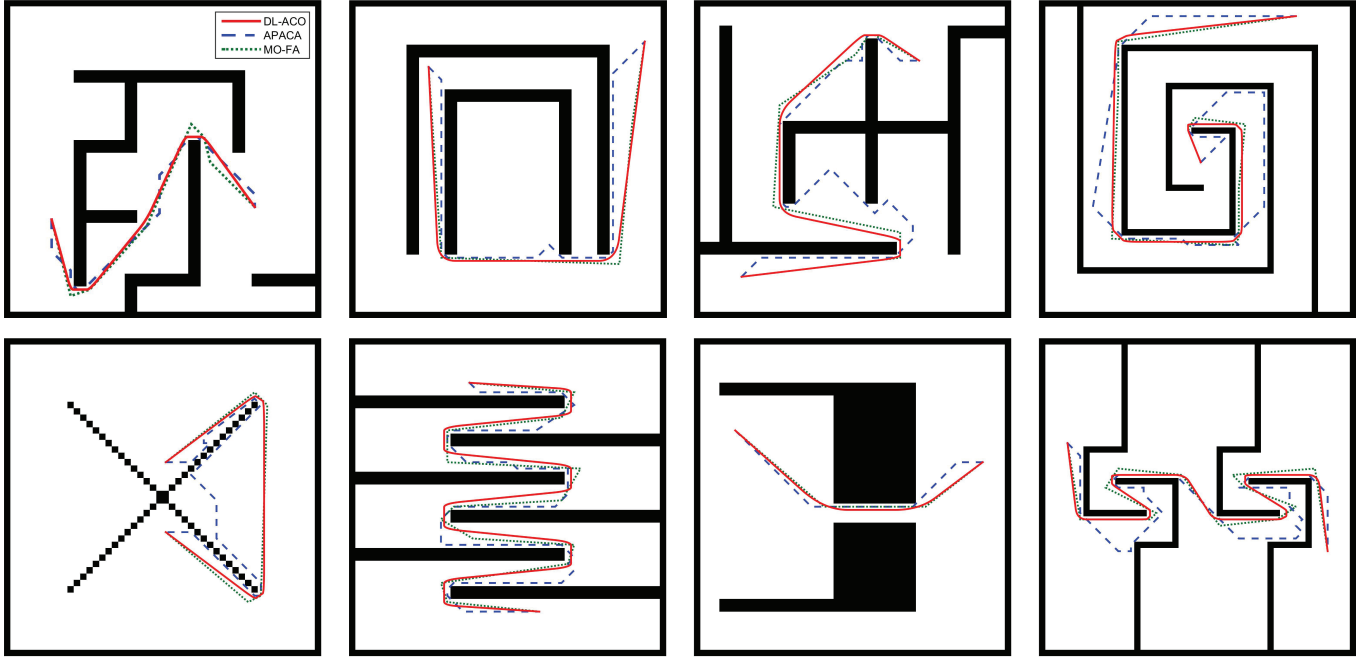


Fig. 8. The best path generated by DL-ACO, APACA and MO-FA.

original path very well. Under this circumstance, the risk of collision will increase. Thus, we propose a piecewise B-spline path smoother to achieve higher degree of fitting.

A B-spline curve is defined by $n + 1$ control points B_i and a knot vector u , the formula is given by

$$C(u) = \sum_{i=1}^{n+1} N_{i,k}(u)B_i, \quad (19)$$

where $N_{i,k}(u)$ is the B-spline based function defined by the following DeBoor-Cox recursion formulas

$$N_{i,1}(u) = \begin{cases} 1, & x_i \leq u < x_{i+1} \\ 0, & \text{otherwise} \end{cases}, \quad (20)$$

$$N_{i,k}(u) = \frac{(u - x_i)N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}, \quad (21)$$

where x_i is called knot values.

Different from conventional B-spline, the piecewise B-spline smooths the path around each corner respectively. As shown in Fig. 6, $T_{i-1} \rightarrow T_i \rightarrow T_{i+1}$ is the original path, T_i is the turning point and path around it should be smoothed. We add P_1, P_2 in $T_{i-1}T_i$, P_3, P_4 on T_iT_{i+1} , and their locations are calculated as:

$$P_1T_i = P_4T_i = X_{safe} \frac{\theta_i}{\pi}, \quad (22)$$

$$\frac{P_2T_i}{P_1T_i} = \frac{P_3T_i}{P_4T_i} = 0.5, \quad (23)$$

where X_{safe} is the safe distance which is calculated according to the security requirements. Then the B-spline curve is defined by P_1, P_3, P_2, P_4 using Eq.(19)-(21). For a complicated route

 TABLE II
DESCRIPTION OF SIMULATION ENVIRONMENT

Map NO.	Map name	Start position	End position
1	Complex1	(8,35)	(40,33)
2	Double U-shape	(12,10)	(46,6)
3	Complex2	(7,43)	(35,9)
4	Spiral	(25,25)	(41,2)
5	X-shape	(25,19)	(25,31)
6	Z-type	(19,7)	(30,43)
7	clasps	(7,15)	(46,20)
8	Corridor	(5,17)	(46,33)

with many turns, we do this at each corner such that the whole path will be smoothed.

Fig. 7 displays the same path smoothed by conventional B-spline and piecewise B-spline respectively. Obviously the path smoothed by piecewise B-spline curve is more fitted with the original path than that smoothed by conventional B-spline curve. The overall advantages of the piecewise B-spline can be summarized as:

- 1) The smoothed path is tangent to the original path, namely that the robot does not have to swerve during the tracking of the path.
- 2) Only the two lines around the corner have an effect on the curve, any other paths can be changed without transforming the smoothed path.
- 3) When the steering angle is large, the fitting between the original path and the smoothed path will still be high.

TABLE III
SIMULATION RESULTS OF DL-ACO, APACA AND MO-FA

Name	Map NO.	Length	Smoothness	Rate _g
DL-ACO	1	58.72	4.52	1.00
	2	91.06	2.96	1.00
	3	89.22	7.17	1.00
	4	110.54	13.61	0.98
	5	67.82	5.60	1.00
	6	145.32	18.92	0.99
	7	45.67	1.37	1.00
	8	96.54	19.74	0.98
APACA	1	60.43	9.42	0.75
	2	94.39	7.85	0.63
	3	104.00	19.63	0.70
	4	133.93	16.50	0.35
	5	79.91	14.13	0.95
	6	153.32	30.24	0.42
	7	46.87	2.35	0.90
	8	117.74	27.48	0.46
MO-FA	1	63.30	5.51	0.86
	2	93.33	4.05	0.99
	3	92.70	9.74	0.95
	4	113.48	15.41	0.74
	5	69.92	8.01	1.00
	6	153.85	23.93	0.81
	7	45.84	1.37	1.00
	8	107.40	22.04	0.78

V. EXPERIMENT AND DISCUSSION

In this section, two experiments are conducted to illustrate the feasibility and merit of the proposed algorithm. In the first experiment, simulations under different maps are performed and the results are compared to that of other path planning algorithms. While in the second experiment we apply this algorithm on a robot for navigation in real environments.

A. Simulation and Comparison

To verify the adaptability of our algorithm in different environments, eight maps are chosen for simulation, which are all a square area within the size of 50×50 and different in terms of number of obstacles, shape of obstacles and width of roads. Start and goal positions are randomly positioned and their locations are depicted in Table II. Parameters of PEACO are set as the following: $n = 100$, $m = 20$, $\alpha = 1$, $\beta = 3$, $\rho = 0.03$, $N_{max} = 100$. Parameters of TPOA are set as: $m = 10$, $\alpha = 0.3$, $\beta = 0.8$, $\rho = 0.1$, $N_{max} = 100$. The safe distance $X_{safe} = 1$.

Two other algorithms are selected for comparison: the adaptive polymorphic ant colony algorithm (APACA) [33] and the multi-objective firefly algorithm (MO-FA) [34]. APACA is also a path planning algorithm based on ant algorithm, which uses the adaptive strategy to guarantee the relative importance of pheromone intensity and desirability. Besides, the direction

TABLE IV
PERCENTAGE OF IMPROVEMENT

Map NO.	APACA			MO-FA		
	Len _i	Sm _i	Rate _{gi}	Len _i	Sm _i	Rate _{gi}
1	0.03	0.52	0.25	0.07	0.18	0.14
2	0.04	0.37	0.37	0.02	0.27	0.01
3	0.14	0.63	0.30	0.04	0.26	0.05
4	0.17	0.18	0.63	0.03	0.12	0.24
5	0.15	0.60	0.05	0.03	0.30	0.00
6	0.05	0.37	0.57	0.06	0.21	0.18
7	0.02	0.41	0.10	0.01	0.00	0.00
8	0.18	0.28	0.52	0.10	0.10	0.20
Avg.	0.10	0.42	0.34	0.05	0.18	0.10

determining method is employed to accelerate convergence. MO-FA is a swarm intelligence algorithm for multi-objective path planning, which has a new evolutionary operator to obtain accurate and effective paths. For parameters of APACA, information heuristic factor is 1, desired heuristic factor is 5, pheromone evaporation coefficient is 0.9, ant number is 120 and cycle number is 200. Parameters of MO-FA are set as: the population size is 50, the generations is 150. Each method runs 100 times in a map.

Fig. 8 displays the best path generated by each method. Clearly that the path produced by our algorithm has less number of turns and shorter length. For instance, our algorithm can avoid U-type traps efficiently, while the APACA always falls into the trap (see Map 3 and 5), and the difference of path quality is more obvious in complex environments (see Map 3, 4 and 8). Besides, it is also revealed that the piecewise B-spline smooths the path around the corner effectively without affecting the safety.

Table III and Fig. 9 summarizes the qualitative comparison between the features of our algorithm and other two algorithms, including length, smoothness and rate of good solution (Rate_g). Length is calculated by Eq.(7), the smoothness of a path is the sum of the angles between line segments for all inside points on the path and is calculated by Eq.(16). Rate_g is an indicator denotes the stability of an algorithm, which is the probability of finding the nearly optimal solution. We define the quality of a route as

$$R_q = L + 5 \times sm. \quad (24)$$

The lowest R_q represents the best solution, and other solutions whose R_q is less than 5% difference comparing with the best solution are considered as nearly optimal solution. According to the results of length and smoothness, our algorithm produces the shortest and smoothest route in all maps. The path generated by APACA has large differences from our method, especially in the presence of traps. This is because of the heuristic information makes the ants go straight to the end point, and the ants do not change their direction until they face the obstacle. MO-FA provides better results. However, due to the particles are randomly positioned, paths around the corner often has some twists and turns, which affects the quality of the route. With regard to (Rate_g), our algorithm can

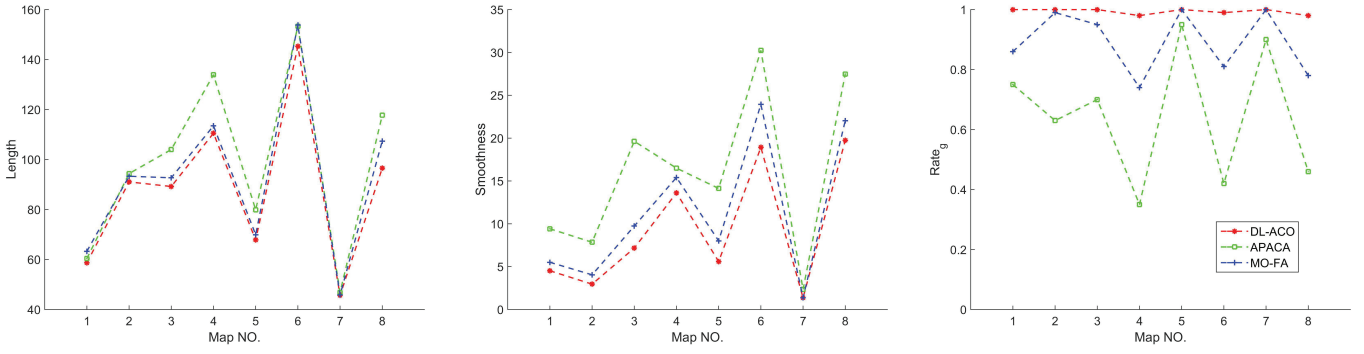


Fig. 9. Algorithm performance by maps.

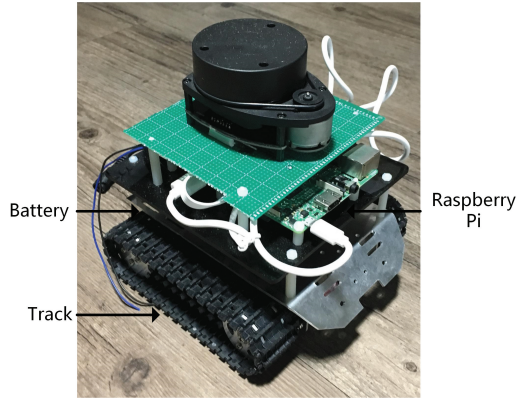


Fig. 10. The experimental Rikibot.

produce a nearly optimal solution almost every time. MO-FA can keep stable in simple environments (see Map 2, 5 and 7) while APACA has poor stability. The superiority of our method is more obvious in complex maps. Table IV depicts the percentage of improvement in all maps.

In conclusion, although the APACA can generate a feasible path, its ability of avoiding traps is poor, which affects its solution quality and stability. Even if the MO-FA can provide similar solutions some time, its stability in complex maps decreases quickly. These results can be interpreted to that our proposed algorithm can generate a better and more robust route consistently in terms of length, smoothness and safety.

B. Path Planning under Real Environment

In this subsection, real experiments are conducted to further validate the effectiveness in practical applications of our algorithm. As shown in Fig. 10, we use a Rikibot for the experiment, which is driven by a Raspberry Pi and uses the laser radar Rplidar A1 to detect obstacles. One indoor and six outdoor environments are selected as the experimental area. Fig. 11 depicts the process of path generation, first the free space area is extracted from the real map, and then the real map is modelled into a 2-D grid-based map and input into the Raspberry Pi. After the route is generated, the motion instructions will be transmitted to the robot and let it to move along the route until reaching the goal. The motion control

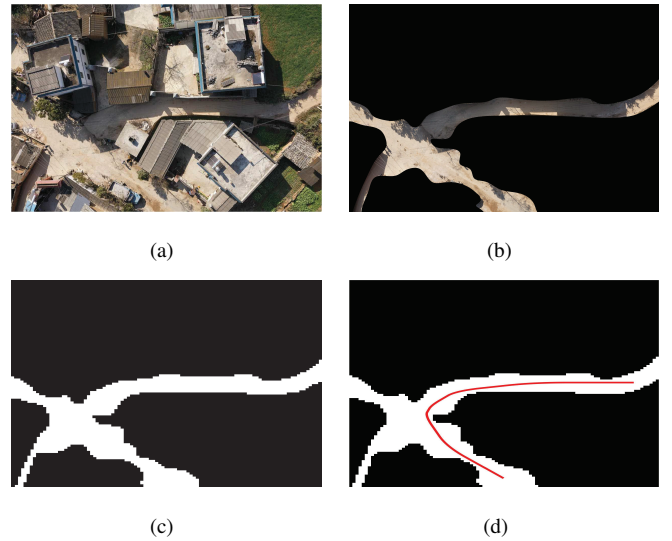


Fig. 11. Illustration of map modelling and path generation. (a) The original map. (b) Extract the free space. (c) The grid-based map. (d) Generate the path.

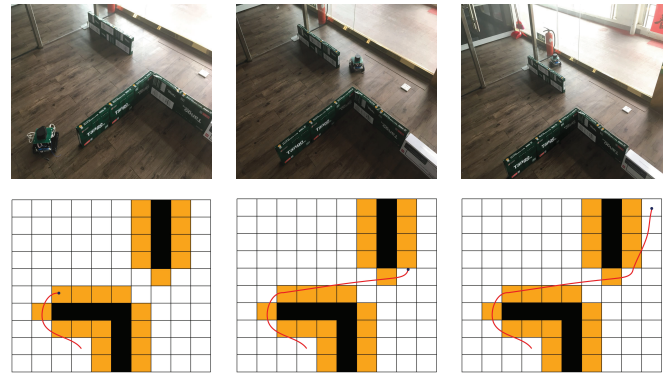


Fig. 12. Experimental results indoor.

laws that we used are in [35], [36]. Parameters of DL-ACO are the same as that in computer simulations.

Fig. 12 displays the experimental results indoor. Respectively are the running pictures and the corresponding trajectory



Fig. 13. Trajectory in outdoor environments. (Star) Start position. (Triangle) goal position.

ries. The real environment is modelled into a 10×10 square. It is clear that the robot successfully avoids and maintains a certain distance with obstacles. Besides, the robot passes through five high-risk grids during the whole navigation, which demonstrates that our algorithm can effectively take the tradeoff between the path length and the safety. Trajectories in outdoor environments are shown in Fig. 13. The robot runs 5 times in each map and the trajectories are almost identical. Paths around the corner are also well smoothed. These results prove the effectiveness in practical applications of the proposed algorithm.

Furthermore, we find that the runtime is directly related to the resolution of the grid map. For instance, the runtime is less than 0.5s under the 10×10 indoor map, while we model the outdoor map into 20×25 , the runtime rises to 3s. Nevertheless, high resolution can improve the accuracy of path planning. Hence we need to keep a balance between the runtime and the path quality in practical applications.

VI. CONCLUSION

In this paper, we have proposed a path planning approach for autonomous mobile robot, i.e. DL-ACO, which utilizes the concept of ACO. The goals are to benefit from its strong searching ability and overcome its weakness. By using the double-layer computing structure, our algorithm has good stability showing that it can provide a good feasible solution every time. Another advantage is that, when robots track on the smoothed path, it can avoid jerky motion and simplify control activity. Simulations show that DL-ACO is superior to other algorithms in generating a shorter and smoother path with good stability, as well as adapting to different maps. Real ex-

periments also verify its effectiveness for challenging practical applications. However, our algorithm has a high requirement in the accuracy of a map, which results in limited scenarios for application. Future work will be directed towards multi-robot path planning and trajectory optimization in unknown dynamic environments.

REFERENCES

- [1] B. Zhang, L. Tang, J. DeCastro, M. J. Roemer, and K. Goebel, "A recursive receding horizon planning for unmanned vehicles," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2912–2920, May. 2015.
- [2] A. Macwan, J. Vilela, G. Nejat, and B. Benhabib, "A multirobot path-planning strategy for autonomous wilderness search and rescue," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1784–1797, Sep. 2015.
- [3] Y. Wang, S. Wang, and M. Tan, "Path generation of autonomous approach to a moving ship for unmanned vehicles," *IEEE Trans. Ind. Electron.*, vol. 62, no. 9, pp. 5619–5629, Sep. 2015.
- [4] J. Barraquand, B. Langlois, and J. C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 22, no. 2, pp. 224–241, Mar. 1992.
- [5] A. Hourtash and M. Tarokh, "Manipulator path planning by decomposition: algorithm and analysis," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 842–856, Dec. 2001.
- [6] T. T. Mac, C. Copot, D. T. Tran, and R. D. Keyser, "Heuristic approaches in robot path planning: A survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13 – 28, Dec. 2016.
- [7] U. A. Syed, F. Kunwar, and M. Iqbal, "Guided autowave pulse coupled neural network (gapcnn) based real time path planning and an obstacle avoidance scheme for mobile robots," *Robot. Auton. Syst.*, vol. 62, no. 4, pp. 474 – 486, Apr. 2014.
- [8] A. Bakdi, A. Hentout, H. Boutami, A. Maoudj, O. Hachour, and B. Bouzouia, "Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control," *Robot. Auton. Syst.*, vol. 89, pp. 95 – 109, Mar. 2017.
- [9] X. Zeng, Y. Li, and J. Qin, "A dynamic chain-like agent genetic algorithm for global numerical optimization and feature selection," *Neurocomputing*, vol. 72, no. 4, pp. 1214–1228, Jan. 2009.
- [10] Y. Fu, M. Ding, C. Zhou, and H. Hu, "Route planning for unmanned aerial vehicle (uav) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization," *IEEE Trans. Syst., Man, Cybern., Sys.*, vol. 43, no. 6, pp. 1451–1465, Nov. 2013.
- [11] C. Mou, W. Qing-xian, and J. Chang-sheng, "A modified ant optimization algorithm for path planning of ucav," *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1712 – 1718, 2008.
- [12] M. Ma, C. Sun, and X. Chen, "Discriminative deep belief networks with ant colony optimization for health status assessment of machine," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 12, pp. 3115–3125, Dec. 2017.
- [13] T. Liao, K. Socha, M. A. M. de Oca, T. Stützle, and M. Dorigo, "Ant colony optimization for mixed-variable optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 503–518, Aug. 2014.
- [14] M. Mavrouniotis, F. M. Müller, and S. Yang, "Ant colony optimization with local search for dynamic traveling salesman problems," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1743–1756, Jul. 2017.
- [15] D. Z. Zhu, P. L. Werner, and D. H. Werner, "Design and optimization of 3-d frequency-selective surfaces based on a multiobjective lazy ant colony optimization algorithm," *IEEE Trans. Antenn. Propag.*, vol. 65, no. 12, pp. 7137–7149, Dec. 2017.
- [16] H. Duan, Y. Yu, X. Zhang, and S. Shao, "Three-dimension path planning for ucav using hybrid meta-heuristic aco-de algorithm," *Simul. Model. Pract. Th.*, vol. 18, no. 8, pp. 1104 – 1115, Sep. 2010.
- [17] M. P. Garcia, O. Montiel, O. Castillo, R. Sepúlveda, and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Appl. Soft Comput.*, vol. 9, no. 3, pp. 1102 – 1110, Jun. 2009.
- [18] Q. Zhu, J. Hu, W. Cai, and L. Henschen, "A new robot navigation algorithm for dynamic unknown environments based on dynamic path re-computation and an improved scout ant algorithm," *Appl. Soft Comput.*, vol. 11, no. 8, pp. 4667 – 4676, Dec. 2011.
- [19] M. M. Abdulkader, Y. Gajpal, and T. Y. ElMekkawy, "Hybridized ant colony algorithm for the multi compartment vehicle routing problem," *Appl. Soft Comput.*, vol. 37, pp. 196 – 203, Dec. 2015.

- [20] Z. Jiao, K. Ma, Y. Rong, P. Wang, H. Zhang, and S. Wang, "A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs," *J. Comput. Sci-neth.*, vol. 25, pp. 50–57, Mar. 2018.
- [21] X. Chen, Y. Kong, X. Fang, and Q. Wu, "A fast two-stage aco algorithm for robotic path planning," *Neural Comput. Appl.*, vol. 22, no. 2, pp. 313–319, Feb. 2013.
- [22] X. Ye, W. Dong, P. Li, and S. Nassif, "Hierarchical multialgorithm parallel circuit simulation," *IEEE Trans. Comput.-Aided Design of Integr Circuits and Syst.*, vol. 30, no. 1, pp. 45–58, Jan. 2011.
- [23] C. C. Tsai, H. C. Huang, and C. K. Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4813–4821, Oct. 2011.
- [24] Y. H. Cheng, C. N. Kuo, and C. M. Lai, "Effective natural pcr-rflp primer design for snp genotyping using teaching-learning-based optimization with elite strategy," *IEEE Trans. NanoBiosci.*, vol. 15, no. 7, pp. 657–665, Oct. 2016.
- [25] P. H. Chen, "Two-level hierarchical approach to unit commitment using expert system and elite pso," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 780–789, May. 2012.
- [26] Y. Wang, Y. Zhao, S. A. Bortoff, and K. Ueda, "A real-time energy-optimal trajectory generation method for a servomotor system," *IEEE Trans. Ind. Electron.*, vol. 62, no. 2, pp. 1175–1188, Feb. 2015.
- [27] D. Shulga, O. Morozov, and P. Hunziker, "A tensor b-spline approach for solving the diffusion pde with application to optical diffusion tomography," *IEEE Trans. Med. Imaging*, vol. 36, no. 4, pp. 972–982, Apr. 2017.
- [28] J. Liu and L. X. Guo, "Evaluation of physical optics integrals from b-spline surfaces by means of a fast locating algorithm of stationary points," *IEEE Trans. Antenn. Propag.*, vol. 65, no. 3, pp. 1495–1499, Mar. 2017.
- [29] M. Rouhani, A. D. Sappa, and E. Boyer, "Implicit b-spline surface reconstruction," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 22–32, Jan. 2015.
- [30] C. F. Hsu and Y. C. Chen, "Microcontroller-based b-spline neural position control for voice coil motors," *IEEE Trans. Ind. Electron.*, vol. 62, no. 9, pp. 5644–5654, Sep. 2015.
- [31] M. Elbanhawi, M. Simic, and R. N. Jazar, "Continuous path smoothing for car-like robots using b-spline curves," *J. Intell. & Robot. Syst.*, vol. 80, no. 1, pp. 23–56, Dec. 2015.
- [32] M. Elbanhawi, M. Simic, and R. Jazar, "Randomized bidirectional b-spline parameterization motion planning," *IEEE Trans. Intell. Transp.*, vol. 17, no. 2, pp. 406–419, Feb. 2016.
- [33] Z. Jiao, K. Ma, Y. Rong, P. Wang, H. Zhang, and S. Wang, "A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs," *J. Comput. Sci-neth.*, vol. 25, pp. 50–57, Mar. 2018.
- [34] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach," *Soft Comput.*, vol. 21, no. 4, pp. 949–964, Feb. 2017.
- [35] L. L. Menegaldo, G. A. N. Ferreira, M. F. Santos, and R. S. Guerato, "Development and navigation of a mobile robot for floating production storage and offloading ship hull inspection," *IEEE Trans. Ind. Electron.*, vol. 56, no. 9, pp. 3717–3722, Sep. 2009.
- [36] C. F. Juang and C. H. Hsu, "Reinforcement ant optimized fuzzy controller for mobile-robot wall-following control," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 3931–3940, Oct. 2009.



Jie Qi received her B.S. degree in automatic control and M.S. degree in electronic engineering from Northwestern Polytechnical University, China, in 1995 and 1999, respectively. And she received her Ph. D. degree in School of Aeronautics and Astronautics from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2011. She is a Lecturer in School of Information Science and Engineering at Xiamen University. Her research is focused on optical sensing and optical communication.



Yongchun Miao received her B.S. degree in 2013 from Jiangxi Normal University in software engineering, Nanchang, China. She is currently pursuing the Ph.D. degree in communication engineering at Xiamen University, Xiamen, China. Her general research interests lie in the areas of underwater acoustic communication and signal processing, and her recent focus on the identification of underwater radiation source based on time frequency analysis.



Haixin Sun received his B.S. degree and M.S. degree in electronic engineering from Shandong University of Science and Technology, Shandong, China, in 1999 and 2003, respectively. Then He received the Ph.D. degree in communication engineering from Institute of Acoustic, Chinese Academy of Science, Shanghai, China, in 2006. He visited the Department of Electrical and Computer Engineering at the University of Connecticut, Storrs, USA, March 2012–April 2013. He is currently an associate professor and

doctoral tutor in School of Information Science and Engineering at Xiamen University.



Jianghui Li received the B.S. degree in communications engineering from Huazhong University of Science and Technology, Wuhan, China in 2011, the M.Sc. degree in communications engineering, and the Ph.D. degree with receiving the K. M. Stott Prize for excellent research in electronics engineering from the University of York, U.K., in 2013 and 2017, respectively. He has been the first researcher receiving the IEEE OES scholarship in U.K. From 2011 to 2012, he served as a Research Assistant with the Chinese Academy of Sciences, Beijing, China. Since 2017, he has been a Research Fellow with the University of Southampton, U.K. His current research interests include adaptive signal processing, wireless communications, underwater acoustics, and ocean engineering.



Hui Yang received his B.S. degree in electronic and information engineering from Soochow University, China, in 2017. He is currently pursuing the M.S. degree in electronic and communication engineering at Xiamen University, China. His main research interests are robot location and navigation.