Research Article

Mobile Robot Path Planning Using Ant Colony Algorithm and Improved Potential Field Method

Guoliang Chen [ID] **and Jie Liu**

School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, Hubei, China

Correspondence should be addressed to Guoliang Chen; glchen@whut.edu.cn

For the problem of mobile robot's path planning under the known environment, a path planning method of mixed artificial potential field (APF) and ant colony optimization (ACO) based on grid map is proposed. First, based on the grid model, APF is improved in three ways: the attraction field, the direction of resultant force, and jumping out the infinite loop. Then, the hybrid strategy combined global updating with local updating is developed to design updating method of the ACO pheromone. The process of optimization of ACO is divided into two phases. In the prophase, the direction of the resultant force obtained by the improved APF is used as the inspired factors, which leads ant colony to move in a directional manner. In the anaphase, the inspired factors are canceled, and ant colony transition is completely based on pheromone updating, which can overcome the inertia of the ant colony and force them to explore a new and better path. Finally, some simulation experiments and mobile robot environment experiments are done. The experiment results verify that the method has stronger stability and environmental adaptability.

1. Introduction

Nowadays, one of the prime concerns of mobile robot is path planning. A path planning optimization method is proposed to calculate the shortest collision free path from source to destination by avoiding static as well as dynamic obstacles. Therefore, it is necessary to select an appropriate optimization technique for optimization of paths [1, 2]. Concerning this problem, scholars proposed a lot of path planning methods, including APF [3], grid method [4], and ACO [5] and some intelligent methods, like fuzzy logic [6], neural network [7], genetic algorithm [8], artificial bee colony (ABC) [9], and particle swarm optimization (PSO) [10]. Among these methods, grid method can find an optimal path, but its efficiency is affected by environment and grid density. Neural network is good at learning, but its net structure is large, and its neuron thresholds change with time under the condition of multiobstacle and dynamic environment. Genetic algorithm has a good global search capability, but its search space is large, and its model is needed to reestablish with environment changing. ABC has the advantages of simple calculation, few parameters, and

easy implementation. But, ABC also has some disadvantages such as a slow convergence. In addition, ABC is first applied to solve the problems of function optimization, while the path planning of mobile robot is a combinatorial optimization problem. Therefore, there are some difficulties in algorithm construction. PSO has a rather fast speed of approaching the optimal solution, which is an advantage to solve optimization problems of continuous systems. But, PSO also has some deficiencies, including the premature convergence and poor local optimization capability.

APF is a popular approach for solving path planning problems because of its simplicity, fast execution time, and applicability to mobile robot, unmanned aerial vehicles. Li et al. [11] proposed an innovative yaw angle-based APF function to achieve the desired road trajectory within certain road boundary and minimize the yaw angle change rate of the autonomous electric vehicle. Rasekhipour et al. [12] developed a path planning controller based on APF for Autonomous Road Vehicles which was able to consider any PF for obstacles and road structures while calculating the optimal path. Malone et al. [13] utilized the APF framework for runtime planning and leveraged a method of stochastic

reachable sets to generate accurate potential fields for moving obstacles. APF easily traps into a local optimum, which leads the robot to be into a deadlock because of its zero resultant force. Many improved algorithms are developed to fix these defects of APF. Sun et al. [14] proposed an improvement based on a distance factor and jump strategy to solve the various situations of the target unreachable problem of the traditional APF. Liu et al. [15] introduced a critical value based on the distance between the robot and the obstacle to modify the repulsion direction for avoiding the local minimum point. Some scholars attempt to combine the intelligent algorithm with APF to eliminate the defects of APF. Pan et al. [16] used the fuzzy controller to improve APF method and safeguarded the reliability of the path planning and path smoothness.

ACO is an intelligence-optimized algorithm that simulates the heuristic mechanism of the shortest route based on pheromone in the process of ants foraging for food [17]. ACO has been used in mobile robot's path planning. Akbarimajd and Hassanzadeh [18] presented a path planning method for mobile robots based on two-dimensional cellular automata, which can be applied for environments with both concave and convex obstacles and be appropriate for multirobot problems as well as dynamic environments. Montiel-Ross et al. [19] developed the navigation software called Ant Colony Test Center. This software based on ant colonies is able to achieve path generation, path planning, and virtual path tracking at once. You et al. [20] designed a new dynamic search model for path planning problem of mobile robot, which increases the diversity of the ant population by using a bigger parameter in the prophase and accelerated convergence by using a smaller parameter to adjust the attenuation model in the anaphase. However, ACO has some problems such as the premature convergence and ant colony lost. For the inherent shortcomings of ACO, there are many improved methods that can be classified into improvements based on single classical ACO and improvements based on mixed algorithms combined ACO and other intelligent algorithm. Liu and Hu [21] proposed the multigranularity pattern ant colony algorithm, in which ants had different window sizes to enhance the diversity of the ants. The proposed algorithm was used for UAV path planning problem. Wang et al. [22] applied genetic algorithm to optimization and configuration parameters of the basic ant colony algorithm. Hsu and Juang [23] proposed a multiobjective, rule-coded, advanced, continuous-ant-colony optimization algorithm that was applied to design the fuzzy controller of wall-following control for a mobile robot.

Of course, ACO has more applications than just robot path planning, and there are other metaheuristic methods that can be used for reference or application in robot path planning. Chu and Tsai [24] proposed a new optimization algorithm of cat swarm optimization (CSO), and CSO is generated by observing the behavior of cats and composed of seeking mode and tracking mode by simulating the behavior of cats. The proposed CSO somehow belongs the swarm intelligence, which may give much better performance than PSO with a weighting factor. The studies deserve sustained attention. For numerical optimization problems, Tsai et al.

[25] proposed an enhanced ABC optimization algorithm, in which the onlooker bee is designed to move straightly to the picked coordinate indicated by the employed bee and evaluates the fitness values near it in the original ABC, and the concept of universal gravitation is introduced into the consideration of the affection between employed bees and the onlooker bees. Aiming at the weaknesses of computational swarm intelligence such as particle swarm optimization (PSO), Ever [26] utilized simplified swarm optimization (SSO) to enhance the path planning of mobile robot in working environment with irregular obstacles, which driven from an inspiration of communal behavior of birds flocking and fish schooling.

APF is a typical online algorithm for robot path planning and has a simple and beautiful mathematical description. So, APF is still very attractive in robot path planning, although it has some inherent limitations and no learning ability. ACO uses all paths of the entire ant colony to describe the solution space of an optimization problem, and obtains the best path through positive feedback based on pheromone. The idea of ACO directly maps the mobile robot's path optimization problem, which is easy to understand and has very intuitive results. Based on the above reasons, this paper combines APF and ACO to research the problem of mobile robot's path planning under the condition of known robot's motion environment described by grid map and proposes a path planning algorithm of ACO mixed with an improved APF. The proposed algorithm can provide an initial path obtained by an improved APF to be the inspired factors of ACO for avoiding its premature convergence and local optimum.

This paper is organized in the following manner: Section 2 presents the basic approach of building a grid map of robot's environment and describes the problem of robot path planning based on the grid map. Section 3 proposes some improved methods about APF after analyzing the existing defects of APF applied to the grid map. Section 4 develops some improved methods about transition algorithm of ant colony. Section 5 demonstrates results of the proposed mixed path planning method by computer simulation experiments and mobile robot's motion experiments in real environments. Finally, the contributions of this paper are summarized in Section 6.

2. Rasterizing Environment Map and Problem Description

The work space of robot is set to a limited area on a two-dimensional plane. The coordinates of obstacles edge points can be determined by processing the original environment map. If we expand the size of the obstacles to a safe distance (usually taken as mobile robot's radius) and add the size of the mobile robot to the obstacles, the robot can be seen as a scale-free particle, and the grid granularity can be calculated as

$$D_g = \phi D_r + \frac{\varphi}{D_m} + \lambda D_s, \tag{1}$$

where D_g represents the grid size; ϕ, φ, and λ are the weighing coefficients; D_m is the map size; and D_r and D_s is mobile robot's size and safe distance.

In this paper, the size of the environment map is $M \times N$ (pixel unit), and then the robot environment can be divided into $[M/D_{\mathrm{g}}] \times [N/D_{\mathrm{g}}]$ ($[\cdot]$ means integer arithmetic) grid units.

Grids are numbered from left to right and from top to bottom. Meanwhile, grids are assigned respectively to distinguish whether the grid is an obstacle space or a free space. The values of grid units are decided by the state value of these pixels. The grid is an obstacle grid if the amount of pixel with value 0 exceeds 1 in the area of the grid and defined as 0 or 1. Eventually, the typical grid model is obtained as shown in Figure 1.

The problem of mobile robot path planning can be described to find an obstacle-free shortest path from the starting grid to the target gird. In Figure 1 calculated by equation (1), the mobile robot considered as a scale-free particle adopts the movement form of linear octree and reaches the neighbor grids along the top, right, bottom, left, upper right, lower right, lower left, and upper left eight directions.

3. Improved APF

3.1. Definition of APF.
Attractive field is the effect of the target point on the robot, and it can guide the robot to move to the target point. Attractive field is radial distribution about the target point and the direction of attraction points to the target point. The traditional attractive potential function is defined as

$$U_{\mathrm{attr}}(\mathbf{X}_{\mathrm{r}}) = K \times \left(\left| \mathbf{X}_{\mathrm{r}} - \mathbf{X}_{\mathrm{g}} \right| \right)^{\alpha}, \tag{2}$$

where K is a constant, $\mathbf{X}_{\mathrm{r}}$ represents the robot's current position coordinates, $\mathbf{X}_{\mathrm{g}}$ represents the target point coordinates, and α is a set constant.

The repulsion field is the effect of the obstacle on the robot, which prevents the robot from entering the obstacle area. When the distance between the robot and the obstacle is greater than the dangerous distance, the repulsion field is zero, and the robot is moved by the attraction to the target point. When the distance between the robot and the obstacle is less than the safety distance, the repulsion field increases rapidly, and its direction is away from the obstacle. The traditional repulsion field function is defined as

$$U_{\mathrm{rep}}(\mathbf{X}_{\mathrm{r}}) = \begin{cases} L \times \left(\dfrac{D_{\mathrm{d}}}{|\mathbf{X}_{\mathrm{o}} - \mathbf{X}_{\mathrm{r}}|} \right)^{\beta}, & |\mathbf{X}_{\mathrm{o}} - \mathbf{X}_{\mathrm{r}}| \leq D_{\mathrm{d}}, \\ \\ 0, & |\mathbf{X}_{\mathrm{o}} - \mathbf{X}_{\mathrm{r}}| > D_{\mathrm{d}}, \end{cases} \tag{3}$$

where L is a constant, $\mathbf{X}_{\mathrm{o}}$ represents the obstacle coordinates, D_{d} represents dangerous distance, and β is a set constant.

D_{d} is usually double of the robot's diameter D_{r}. In the grid map, the value of D_{d} is set by the number of grids. For example, if the robot diameter D_{r} is 250 mm and the grid value D_{g} is 375 mm, then D_{d} is taken to two grids. Suppose the coordinates of the target grid $\mathbf{X}_{\mathrm{g}}$ is (a, b), the danger zone R_{d} can be calculated as

$$R_{\mathrm{d}} = \{(i, j) \mid i \geq a - 2, i \leq a + 2, j \geq b - 2, j \leq b + 2\}, \tag{4}$$

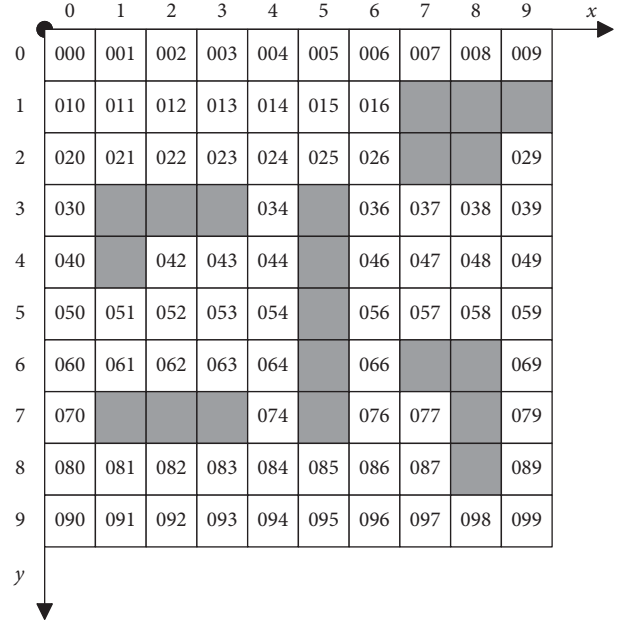|	0	1	2	3	4	5	6	7	8	9	x
0	000	001	002	003	004	005	006	007	008	009	
1	010	011	012	013	014	015	016				
2	020	021	022	023	024	025	026			029	
3	030			034		036	037	038	039		
4	040		042	043	044		046	047	048	049	
5	050	051	052	053	054		056	057	058	059	
6	060	061	062	063	064		066			069	
7	070			074		076	077		079		
8	080	081	082	083	084	085	086	087		089	
9	090	091	092	093	094	095	096	097	098	099	

y

FIGURE 1: Typical grid model.

where (i, j) are the coordinates of those grids within the hazard distance.

The total potential field $U_{\mathrm{sum}}(\mathbf{X}_{\mathrm{r}})$ is defined as

$$U_{\mathrm{sum}}(\mathbf{X}_{\mathrm{r}}) = U_{\mathrm{attr}}(\mathbf{X}_{\mathrm{r}}) + U_{\mathrm{rep}}(\mathbf{X}_{\mathrm{r}}). \tag{5}$$

According to the distribution of attractive field and repulsive field, the magnitude and direction of the force effecting on the robot in each grid can be calculated. When the robot is located in the grid far away from the obstacle, it is only subjected to the attraction pointing to the target. When the distance is closer to the obstacle, the repulsive field increases rapidly, and the robot is subjected the resultant force of attraction and repulsive force, which changes the robot's movement direction.

3.2. Improvement about APF.
It is easily fallen into a local minimum when using the traditional APF for path planning in the case of the two kinds of gird traps shown in Figure 2. In Figure 2(a), attraction and repulsive force are all along the horizontal direction, and the repulsive force is greater than the gravitational force. So, the resultant force is opposite to the target, and the robot will fall into a dead-end cycle of transverse reciprocating motion. In Figure 2(b), the robot is subjected the attraction force in the upper right of 45° and the repulsive force in the lower right of 45°, and the repulsion is greater than the attraction. Therefore, the robot will move away from the obstacle and the target point.

For these defects, the traditional APF is improved by employing the global environment information of the grid map as follows.

3.2.1. Improving the Attractive Function.
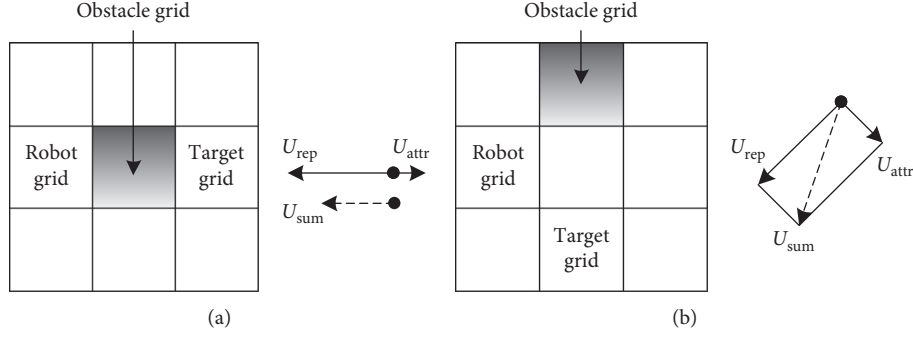For the grid rap shown in Figure 2, the attractive field distribution opposite

FIGURE 2: Two kinds of grid traps. (a) The obstacle grid is between the robot grid and target grid. (b) The obstacle grid is not between the robot grid and target grid.

to that of the conventional potential field is used, which is described as

$$
U_{\text{attr}}(\mathbf{X}_r) = \begin{cases} K \times \left(\dfrac{S}{|\mathbf{X}_r - \mathbf{X}_g|} \right)^{\alpha}, & |\mathbf{X}_r - \mathbf{X}_g| < S, \\ \\ K, & |\mathbf{X}_r - \mathbf{X}_g| \geq S, \end{cases} \tag{6}
$$

where S is the threshold of the attractive distance.

S is generally taken to 1/10 of the grid map length and width. If the value of S is 3 and the coordinates of the target grid $\mathbf{X}_g$ is (c, d), then attractive area R_{attr} is

$$
R_{\text{attr}} = \{(i, j) \mid i \geq c - 3, i \leq c + 3, j \geq d - 3, j \leq d + 3\}, \tag{7}
$$

where (i, j) are the coordinates of those grids within the attractive distance.

3.2.2. Improving the Resultant Force of APF. There are only eight cases of the robot's transfer directions in the grid map; that is, the resultant force has eight standard directions. The direction of vertically upward is set to 0° and clockwise is the positive direction. The angular distribution of eight standard directions in the grid map is shown in Figure 3.

The angle difference between the resultant force direction and the eight standard directions is calculated respectively. The standard direction with the smallest absolute value of the angle difference is the direction of the resultant force. If the minimum absolute value of the angle difference has two or more standard directions, then the smaller standard direction is adopted as the resultant force direction.

3.2.3. Additional Force. When the direction of the robot's resultant force is opposite to the direction of the robot's attraction, the robot will fall into an endless loop. At this moment, the robot will be forced out of the endless loop by adding an additional force perpendicular to the resultant force. The force analysis of adding an additional force is shown as Figure 4. In Figure 4, F_{sum} is the original resultant force, F_{add} is the added additional force, F'_{sum} is the new resultant force, and γ is the angle between the new resultant force and the original resultant force. To ensure F'_{sum} forces
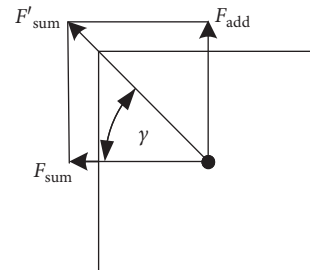

FIGURE 3: The angle of eight grid neighbors of robot.

FIGURE 4: Force analysis of adding additional force.

the robot to move out of the loop, there must be $\gamma > 22.5°$. So it should be $F_{\text{add}} > \tan 22.5° \times F_{\text{sum}}$.

4. Transition Algorithm of Ant Colony

The improved ACO optimization is divided into two phases. In the prophase, the direction of the resultant force calculated by the improved APF is used to be as inspired factor of ACO. In the anaphase, inspired factor is canceled, and ant colony transition is completely based on pheromone updating.

The paper uses the cellular ant algorithm to plan robot's path, which imitates ants' movement through the moving and iterating of cellular. The transition algorithm of ant colony is designed by pseudorandom proportion rule. The rule uses prior knowledge about the problem and carries out a search with tendentiousness, which forces some ants to choose these grids of nonmaximally probability for searching the other path except the optimal path.

4.1. Transition Probability Function.

The transition probability function of ant colony cellular automata is given as

$$P_{ij} = \frac{\left[\tau_{ij}\right]^{p_\alpha} \cdot \left[\eta_{ij}\right]^{p_\beta}}{\sum_{j \in \mathbf{C}_i}\left[\tau_{ij}\right]^{p_\alpha} \cdot \left[\eta_{ij}\right]^{p_\beta}}, \tag{8}$$

where P_{ij} is comprehensive probability, η_{ij} is the visibility of arc (i, j) decided by inspiring factor, τ_{ij} is the track strength of edge arc (i, j) which means the number of pheromone on the trajectory, p_α is the relative importance of the trajectory, p_α is the relative importance of the visibility, $\mathbf{C}_i\{c_0, c_1, \ldots, c_7\}$ is the neighbor cells set of the cellular automata c_i, and $j = 0, 1, \ldots, 7$.

η_{ij} represents the visibility between the target location of next step and the current cellular automata. The higher the visibility is, the greater the preference probability is, and the greater η_{ij} is. In the initial of operation, the value of visibility can be formulated as

$$\eta_{ij} = \begin{cases} \dfrac{1}{\left|\langle \mathbf{M}_j, \mathbf{F}_i \rangle\right|}, & \left|\langle \mathbf{M}_j, \mathbf{F}_i \rangle\right| > \dfrac{1}{D}, \\ \\ D, & \left|\langle \mathbf{M}_j, \mathbf{F}_i \rangle\right| \le \dfrac{1}{D}, \end{cases} \tag{9}$$

where the vector $\mathbf{M}_j$ $(j = 0, 1, \ldots, 7)$ is determined by the cellular automata c_i and its neighbor cells. $\mathbf{F}_i$ is the resultant force of the artificial potential field acting upon c_i. D is the maximum visibility defined in advance. The visibility of the neighbor cells to c_i is determined by the angle between $\mathbf{M}_j$ and $\mathbf{F}_i$. The smaller the angle is, the higher the visibility is.

To reflect the visibility of neighbor cells in different directions and simplify the algorithm calculation, the visibility is discretized to. According to equation (9), eight-neighbor cells are sorted according to the size of their visibilities. The vector of visibility is defined as $\mathbf{V}\{v_0, v_2, \ldots, v_7\}$ and meets $\sum_{l=0}^{7} v_l = 1$, and $1 \ge v_0 \ge v_1 \ge v_2 \ge v_3 \ge v_4 \ge v_5 \ge v_6 \ge v_7 \ge 0$.

4.2. Updating Mechanism of Pheromone.

The hybrid strategy of global updating and local updating is presented. The global updating is used to update the number of pheromone for grids in the map when the ant colony finishes one movement cycle. The local updating is used to update the number of pheromone for grids corresponding cellular automata when the ant colony implements the move rules of cellular automata.

The global updating rule of pheromone is defined as

$$\tau_{ij}^{n} = \rho \cdot \tau_{ij}^{o} + \sum_{k} \Delta\tau_{ij}^{k}, \tag{10}$$

where ρ $(0 \le \rho < 1)$ is the durability of the trajectory, τ_{ij}^{n} is the new τ_{ij}, τ_{ij}^{o} is the original τ_{ij}, and $\Delta\tau_{ij}^{k}$ is the number of pheromone per unit length on the edge arc (i, j) of ant k.

According to different types of ant colony, $\Delta\tau_{ij}^{k}$ can be set different values. It can be described by threshold mechanism as

$$\Delta\tau_{ij}^{k} = Q = \begin{cases} Q_{\max}, & N_n \le M_n, \\ Q_{\min}, & M_n < N_n \le S_n, \\ 0, & N_n > S_n, \end{cases} \tag{11}$$

where $Q_{\max}$ and $Q_{\min}$ are constants to reflect the track number kept by ants, respectively; M_n is the threshold of node determined by the map information, the starting point, and the end point; N_n is the node number of planned path; and S_n is the maximum number of skipping steps.

The principle of equation (11) is that when $N_n < M_n$, the updating of nodes pheromone will be an optimum global updating and when $M_n < N_n \le S_n$, the updating of nodes pheromone will be an suboptimum global updating.

When ants move from the former cellular automata to the next, they will record the coordinate of the cellular automata and update the amount of current cellular automata pheromone. The locally updating mechanism can prevent ants to be lazy, which avoids the algorithm converging to a known current path and giving up finding new paths. The locally updating mechanism can be represented as

$$\tau_{ij}^{n} = \tau_{ij}^{o} + \Delta\tau_{ij}^{k}. \tag{12}$$

4.3. Ant Colony Punishment Mechanism.

The comprehensive probability P_{ij} can be calculated by equation (8). The moving rules of cellular automata can be represented as

$$\mathbf{p}(i_{t+1}) = \begin{cases} c_{0,t}, & 0 \le R < P_{i0}, \\ c_{1,t}, & P_{i1} \le R < P_{i2}, \\ \vdots & \\ c_{7,t}, & P_{i6} \le R \le P_{i7}, \end{cases} \tag{13}$$

where $\mathbf{p}(i_{t+1})$ is the position of cellular automata c_i at time $t + 1$, R is a random integer from 0 to 100, $c_{n,t} \in \mathbf{C}_i$ $(n = 0, 1, \ldots, 7)$, and $P_{im} = 100 \times \sum_{l=0}^{m} P_{il}$ $(m = 0, 1, \ldots, 7)$.

4.4. Moving Rules of Cellular Automata.

It is possible that there are some trap grids in the grid map. Ants will lose their way when they move into these traps. As shown in Figure 5, it is assumed that the robot starting point is $(0, 0)$ and the target point is $(3, 3)$. The probabilities of ants moving from $(0, 0)$ to $(1, 1)$ and from $(1, 1)$ to $(1, 2)$ are 60% and 100%, respectively. Therefore, a lot of ants will fall into a trap grid.

A punishment mechanism is developed to avoid this trap phenomenon. Firstly, the last five nodes of the current ant are traversed to judge whether these five nodes are the same. Secondly, if these five nodes are the same node, the ant queue is deleted from the active queues and its header is set to 2, and the grid corresponding to the node is treated as an obstacle grid and to be marked to 0. Lastly, the jumping probability table of ants is updated.

5. Experiment and Discussion

5.1. Simulation and Contrast Experiments.

In this paper, experiment platform is VC++ 6.0, the diameter of robot is

FIGURE 5: The typical form of a trap grid.

250 mm, the size of environment is $12\,\text{m} \times 9\,\text{m}$, and the map's pixel is 640×480. The robot's path is planned by the improved APF, the APF-ACO mixed method, the traditional ACO, and the traditional APF, respectively. The feasibility, advantage, and disadvantage of our algorithm are verified by comparing with other three algorithms from the planning path's time-consuming, number of node, turning angle, etc.

(1) Set up the map lattice model. We set the safety distance is 125 mm, $\phi = \lambda = 1$, and $\varphi = 0$. The grid size is $375\,\text{mm} \times 375\,\text{mm}$ calculated by equation (1), and the size of the grid is 32×24.

(2) Set up the potential field in the grid map according to the improved APF. According to the target and obstacle information in the grid map. First, calculate the force of each grid and store them in the corresponding memories. Then, traverse all grids and compare the direction of the resultant force with the direction of attraction. Finally, if the direction of resultant force is opposite to the direction of attraction, the additional force is added to change the direction of the resultant force. We set $\alpha = \beta = 1$, $K = L = 1$, $S = 2$, $D_d = 1$, and $F_{\text{add}} = 0.5 \times F_{\text{sum}}$.

(3) Initialize the pheromone table and calculate the visibility of eight-neighbor cell according to the orientations of grid determined by step 2. We set the initial values of the pheromone of all nonobstacle grids and all obstacle grids to 12.5 and 0, respectively. At the same time, we set $\mathbf{V} = \{0.4, 0.2, 0.2, 0.08, 0.08, 0.02, 0.02, 0\}$, $p_\alpha = 2$, $p_\beta = 1$, $\rho = 0.8$, $Q_{\text{max}} = 160$, $Q_{\text{min}} = 25$, $M_n = 50$, and $S_n = 100$. So, the pheromone table of jump probability of ants can be got.

(4) Place 1000 ants to plan the path by the improved ACO. Cellular automatas move according to the table of jump probability. The pheromone table and the path nodes of ants can be updated or recorded synchronously. The process is repeated until the shortest path is outputted. We get the ants nodes the shortest planned path and computational time.

(5) Run the improved APF to plan the path based on the direction of resultant force by step 2. The process is repeated 100 times, and then output the nodes of the planned path and computational time.

(6) Run the traditional ACO to plan the path. We set the visibilities $v_i = 0.125$ $(i = 0, 1, \ldots, 7)$, the amount of

ants to 1000, and other parameters same as step 3. Finally, we get the planning results, including the ant nodes of the shortest path and computational time.

The paths planned by four algorithms are shown in Figure 6.

The parameters of calculating and path are given in Table 1.

The traditional APF falls into an endless loop at the beginning of the path and fails to reach the target point after 100 repeated experiments. The improved APF can avoid obstacles and takes 1500 ms to plan the path from the starting point to the target point. This shows that the improved APF can effectively solve the problem of local minimum. But, the planned path is not satisfactory because it has the path length error of 96.9% and the turning angle error of 1600%.

Using the traditional ACO, only a few ants reach the desired position within the set 100 steps. So, the traditional ACO cannot get a convergence result to form a path within the given time of 30000 ms. There are various reasons for the failure, among which the main reason is that the initial values of the visibilities v_i $(i = 0, 1, \ldots, 7)$ are no heuristic for ant colony searching. In fact, we can get satisfactory results in some experiments based on a suitable set of $\mathbf{V}$. But, the problem is that we do not know how to set these parameters properly.

Our mixed algorithm can obtain a more satisfactory result. Compared to the improved APF, although the mixed algorithm takes more than 390 ms, it plans out a better path which reduces unnecessary turning and redundant paths and has the path length error of 10.3% and the angle error of 100%. Compared to the traditional ACO, the mixed algorithm improves the convergence speed and gets a convergence path with the same p_α, p_β, and initial pheromone. As a result, this mixed algorithm combines the advantages of APF and ACO and has better performance of path planning.

5.2. Mobile Robot Motion Experiment.

The purpose of mobile robot experiments is to verify the APF-ACO mixed algorithm adaptability to different tasks. In this paper, experiments are designed to mobile robot's path planning and path tracking from different start points to different end points in a same obstacle environment.

The mobile robot used in experiments is Traveler II as shown in Figure 7, which can be considered as a cylinder with a diameter of 250 mm.

The mobile robot's experimental environment with the size of $12\,\text{m} \times 9\,\text{m}$ and two sets mobile robot's start and end points are designed as shown in Figure 8.

A comprehensive index for the planned path is presented to value the merits and effectiveness of the proposed APF-ACO mixed algorithm. The index is defined as

$$ST = \sigma \times \text{Len} + \varsigma \times \text{Tangle}, \qquad (14)$$

where Len is the length of planned path, Tangle is the turning angle of planned path, and σ and ς are weighted
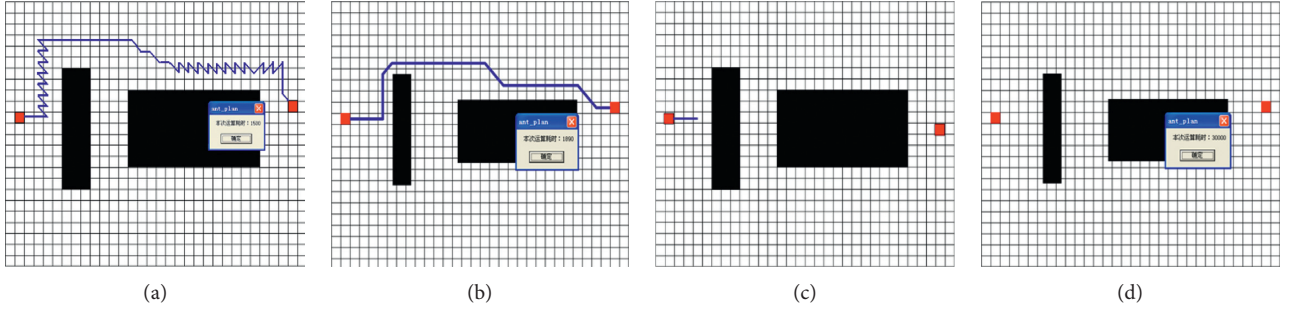
(a)   (b)   (c)   (d)

Figure 6: The experimental results. (a) The path planned by improved APF; (b) APF-ACO mixed method; (c) the traditional APF; (d) the traditional ACO.

Table 1: Test results of robot path planning by three algorithms.

Algorithm	Time-consuming (ms)	Grid number of path	Path length (the number of grids)	Turning angle (rad)
APF-ACO	1890	32	34	2π
Improved APF	1500	57	65	27π
Traditional ACO	30000	0	0	0
Traditional APF	∞	3	3	0

Figure 7: The image of Traveler II robot.

coefficients, and $\sigma + \varsigma = 1$. σ is taken a larger value than ς if the path length is paid more attention is concerned. Otherwise, ς is taken a larger value than σ.

The experimental steps are as follows:

(1) *Grid modeling.* This paper sets the size of grid map to 640×480 (pixels), $\phi = \lambda = 0.5$, and $\varphi = 0$. Then the size of a single grid is 187.5 mm $\times$ 187.5 mm, and the ratio of the grid size and real environment size is 4 : 75. So, the mobile robot's environment shown by Figure 8 can be rasterized to grid map as shown in Figure 9. According the grip size of 187.5 mm, the coordinates of two sets of start and end points in Figure 9 are transferred to (1, 22), (62, 22) and (1, 46), (62, 1), respectively.

(2) *Establishing the potential field.* This paper sets $\alpha = \beta = 2$ $K = L = 1$ $S = 3$ $D_d = 2$, and $F_{add} = 0.5 \times F_{sum}$. According to the potential field and the information of grid map, we can calculate the resultant force of each grid and its direction, and the information is stored.

(3) *Path planning.* We set $\mathbf{V} = \{0.5, 0.18, 0.18, 0.05, 0.05, 0.02, 0.02, 0, 0\}$, $p_\alpha = 2$, $p_\beta = 1$ $\rho = 0.8$, $Q_{max} = 160$, and $Q_{min} = 25$ and placed 1000 ants in the grid map. Lastly, the optimum path obtained by APF-ACO algorithm is shown in Figure 10.

(4) *Mobile robot motion control.* Firstly, we obtain the sequences of nodes in grid map of the path planned by the APF-ACO algorithm. Secondly, the coordinate transformation of the next node is processed from the grid map to the environment map. Then, the mobile robot's displacement length and the turning angle between the current node and the next node can be calculated. So, according to these motion parameters and control laws, the function of the left and right wheels' velocity and time can be determined. Finally, the mobile robot is controlled to move until it reaches the last node by the function.

Figure 11 shows the moving process of the mobile robot. The motion results show the mobile can move from the start point to the end point along a smooth and nonobstacles path.

Table 2 gives some results of the algorithm execution process and planning path. The data show the APF-ACO algorithm can plan a higher quality path.

Further, the error rate of the path length e_l is defined as

$$e_l = \frac{\left| l_p - l_a \right|}{l_p},\qquad(15)$$

where l_a is the mobile robot's actual path length and l_p is theoretical movement distance.

The error rate of consumption time e_t is defined as

$$e_t = \frac{\left| t_p - t_a \right|}{t_p},\qquad(16)$$

(a)   (b)
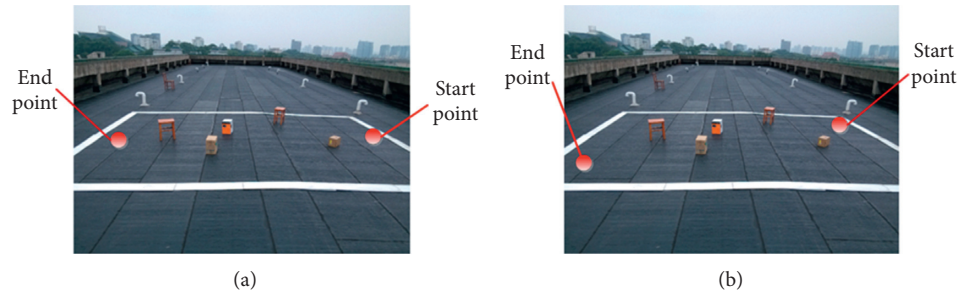
Figure 8: The experimental environment and two kinds of start point and end point. (a) The setting points (I), its start point coordinates and end point coordinates are (187.5, 4125) and (187.5, 8625), respectively; (b) the setting points II, its start point coordinates and end point coordinates are (11625, 4125) and (11625, 187.5), respectively.
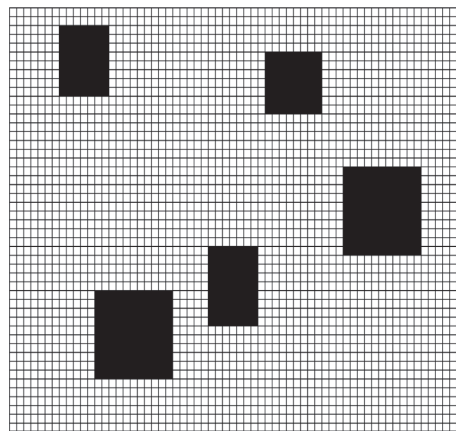

Figure 9: Raster map for barrier environment.

(a)   (b)
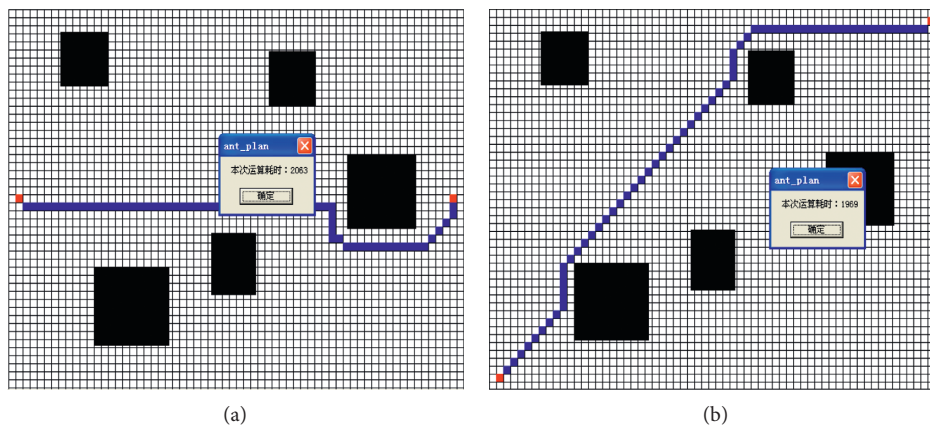
Figure 10: Path planning results. (a) The path planned by APF-ACO algorithm for the setting points I and (b) APF-ACO algorithm for the setting points II.

where t_a is the actual consumption time and t_p is the theoretical consumption time.

The error rate of path length and the error rate of consumption time are shown in Table 3, which shows that the error rate of path length and the error rate of consumption time will increase correspondingly with the increase of the path nodes. The maximum error rate of path length is 6.58%, and the maximum value of the error rate of time is 13.97%.

Considering the accumulated error of the inertial sensor of the mobile robot, the error is within the expected range.

6. Conclusion

This paper presents a hybrid path planning mixed of an improved APF and an improved ACO for mobile robot based on grid map. The main improvements are as follows:
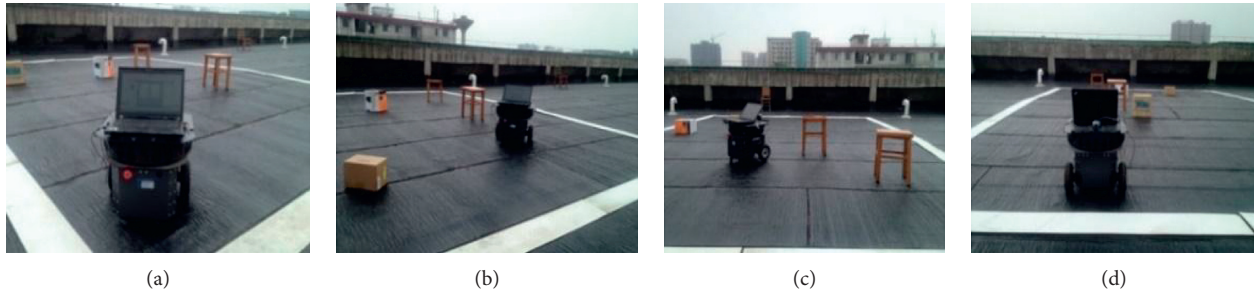
(a)   (b)   (c)   (d)

FIGURE 11: The images of moving process of mobile robot.

TABLE 2: Process and result data.

Items	Setting points I	Setting points II
Time-consuming (ms)	2063	1969
Numbers of path nodes	66	68
Path length (unit grid)	69	84
Turning angle (radian)	2π	1.5π
Comprehensive indicator (ST)	9.61	10.2
Robotic exercise time (s)	147	155
Robot moving distance (mm)	13788	16661

TABLE 3: The results of path tracking.

Items	Experiment of setting points I	Experiment of setting points II
e_l (%)	6.58	5.78
e_t (%)	11.36	13.97

(1) The improvements about APF: Firstly, to avoid grid traps, the attraction is calculated by the reciprocal of the distance between the current position and the target position. Secondly, to simplify the calculation, the direction of resultant force is separated into 8 directions based on the current grid. Thirdly, an additional force perpendicular to the direction of the resultant force is added to form a new resultant force when the robot falls into an endless loop.

(2) The improvements about ACO: The process of optimization is divided into two phases. In the pro-phase, the direction of the resultant force obtained by the improved APF is used as the inspired factors, which leads ant colony to move in a directional manner. In the anaphase, the inspired factors are canceled, and ant colony transition is completely based on pheromone updating, which can overcome the inertia of the ant colony and forces them to explore a new and better path.

In the grid environment, the planning result by the improved APF is employed as the optimizing foundation of ACO. The method overcomes some drawbacks of APF such as goal being nonreachable and easily falling into the extreme point, etc., and improves the optimization effect and the rate of convergence of ACO. The simulation experiments and mobile robot environment experiments verify that the method has stronger stability and environmental adaptability.

Data Availability

The research library related to the dissertation in this study is established in https://github.com/glchenwhut/research.git, where you can access the folders and find experimental data and lists.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Guoliang Chen conceived the idea, designed the experiments, and wrote the paper. Yang Xu helped with the algorithm and analyzing the experimental data.

Acknowledgments

References

[1] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[2] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1–4, pp. 65–100, 2010.

[3] D. J. Bennet and C. R. McInnes, "Distributed control of multi-robot systems using bifurcating potential fields," *Robotics and Autonomous Systems*, vol. 58, no. 3, pp. 256–264, 2010.

[4] G. Tanzmeister, M. Friedl, D. Wollherr, and M. Buss, "Efficient evaluation of collisions and costs on grid maps for autonomous vehicle motion planning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2249–2260, 2014.

[5] O. Montiel-Ross, R. Sepúlveda, O. Castillo, and P. Melin, "Ant colony test center for planning autonomous mobile robot navigation," *Computer Applications in Engineering Education*, vol. 21, no. 2, pp. 214–229, 2013.

[6] C.-J. Kim and D. Chwa, "Obstacle avoidance method for wheeled mobile robots using interval type-2 fuzzy neural

network," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 3, pp. 677–687, 2015.

[7] M. Duguleana, F. G. Barbuceanu, A. Teirelbar, and G. Mogan, "Obstacle avoidance of redundant manipulators using neural networks based reinforcement learning," *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 2, pp. 132–146, 2012.

[8] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," *Procedia Computer Science*, vol. 127, pp. 180–189, 2018.

[9] M. A. Contreras-Cruz, V. Ayala-Ramirez, and U. H. Hernandez-Belmonte, "Mobile robot path planning using artificial bee colony and evolutionary programming," *Applied Soft Computing*, vol. 30, pp. 319–328, 2015.

[10] P. K. Das, H. S. Behera, S. Das, H. K. Tripathy, B. K. Panigrahi, and S. K. Pradhan, "A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment," *Neurocomputing*, vol. 207, pp. 735–753, 2016.

[11] B. Li, H. Du, and W. Li, "A potential field approach-based trajectory control for autonomous electric vehicles with in-wheel motors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2044–2055, 2017.

[12] Y. Rasekhipour, A. Khajepour, S. K. Chen, and B. Litkouhi, "A potential field-based model predictive path-planning controller for autonomous road vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1255–1267, 2017.

[13] N. Malone, H.-T. Chiang, K. Lesser, M. Oishi, and L. Tapia, "Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1124–1138, 2017.

[14] J. Sun, J. Tang, and S. Lao, "Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm," *IEEE Access*, vol. 5, pp. 18382–18390, 2017.

[15] Y. J. Liu, T. Dai, and J. H. Song, "Research of path planning algorithm based on improved artificial potential field," *Journal of Shenyang Ligong University*, vol. 36, pp. 61–66, 2017, in Chinese.

[16] Z. Pan, J. Q. Li, K. M. Hu, and H. Zhu, "Intelligent vehicle path planning based on improved artificial potential field method," *Applied Mechanics and Materials*, vol. 742, pp. 349–354, 2015.

[17] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[18] A. Akbarimajd and A. Hassanzadeh, "Autonomously implemented versatile path planning for mobile robots based on cellular automata and ant colony," *International Journal of Computational Intelligence Systems*, vol. 5, no. 1, pp. 39–52, 2012.

[19] O. Montiel-Ross, R. Sepúlveda, O. Castillo, and P. Melin, "Ant colony test center for planning autonomous mobile navigation," *Computer Applications in Engineering Education*, vol. 21, no. 2, pp. 214–229, 2013.

[20] X. M. You, S. Liu, and J. Q. Lv, "Ant colony algorithm based on dynamic search strategy and its application on path planning of robot," *Control and Decision*, vol. 32, pp. 552–556, 2017, in Chinese.

[21] Z. Liu and Y. N. Hu, "Multi-granularity pattern ant colony optimization algorithm and its application in path planning," *Journal of Central South University (Science and Technology)*, vol. 44, pp. 3713–3722, 2013.

[22] Z. Q. Wang, X. G. Zhu, and Q. Y. Han, "Mobile robot path planning based on parameter optimization ant colony algorithm," *Procedia Engineering*, vol. 15, pp. 2738–2741, 2011.

[23] C.-H. Hsu and C.-F. Juang, "Multi-objective continuous-ant-colony-optimized FC for robot wall-following control," *IEEE Computational Intelligence Magazine*, vol. 8, no. 3, pp. 28–40, 2013.

[24] S. C. Chu and P. W. Tsai, "Computational intelligence based on the behavior of cats," *International Journal of Innovative Computing, Information and Control*, vol. 1, pp. 163–173, 2007.

[25] P. W. Tsai, J. S. Pan, B. Y. Liao, and S. C. Chu, "Enhanced artificial bee colony optimization," *International Journal of Innovative Computing, Information and Control*, vol. 12, no. B, pp. 5081–5092, 2009.

[26] Y. K. Ever, "Using simplified swarm optimization on path planning for intelligent mobile robot," *Procedia Computer Science*, vol. 120, pp. 83–90, 2017.