An Improved Ant Colony Optimization Applied in Robot Path Planning Problem

Xiangyang Deng
Department of Electric and Information Engineering,
Naval Aeronautical and Astronautical University, Yantai, China
Email: d_sophie@sina.com

Limin Zhang
Department of Electric and Information Engineering,
Naval Aeronautical and Astronautical University, Yantai, China
Email: iamzlm@163.com

Lan Luo
Basic Teaching Department, Yantai Vocational College, Yantai, China

***Abstract*—an improved Ant colony optimization algorithm (PM-ACO for short) is proposed to solve the robot path planning problem. In PM-ACO, ants deposit pheromone on the nodes but not on the arcs, resulting in that the trails of pheromone become the form of marks, which consist of a series of pheromone points. After ant colony's tours, the iteration-best strategy is combined with an r-best nodes rule to update the nodes' pheromone. The stability of PM-ACO is analyzed and some advancement to the algorithm is proposed to improve the performance. Because the pheromone on several arcs is integrated into the pheromone on one node, a rapid pheromone accumulation occurs easily. It is the major causes to the instability. An r-best nodes rule is presented for regulating the pheromone distribution and an adaptive mechanism is designed to further balance the pheromone arrangement. In addition, to shorten the time wasted in constructing the first complete solution and get a better solution, an azimuth guiding rule and a one step optimization rule are used in local optimization. By establishing a grid model of the robot's navigation area, PM-ACO is applied in solving the robot path planning. Experimental results show that an optimal solution of the path planning problem can be achieved effectively, and the algorithm is practical.**

***Index Terms*—ant colony optimization, robot path planning, pheromone mark, r-best nodes rule**

I. INTRODUCTION

Ant colony optimization (ACO) is a meta-heuristic algorithm as well as a biomimetic evolutionary algorithm, which was first proposed in the first European Conference on Artificial Life (ECAL) in 1991[1]. It is suitable to solve the discrete combinatorial optimization problem, and has the characteristics of parallel computing, self-adaptive, positive feedback, and good robustness. The first ACO algorithm is Ant system (AS), which has three different versions called ant-density[2], ant-quantity[1] and ant-cycle[3]. As we know, the ant-cycle is the most popular version, and it has formed the basic framework of an ACO algorithm. And then, a mutation of ACO called Ant-Q[4] system was proposed subsequently, which replaced the pheromones with Q values, and it is similar to the Q-Learning algorithm. In 1997, the ant colony system (ACS)[5], which improved the ACO performance substantially, was presented and analyzed the ACO algorithms in an all-round way, and some new mechanisms were introduced. Afterwards, it has become a standard version of ACO algorithms which has been applied extremely in a great many engineering fields. After that, the ACO attracted highly attention of the scholars, and a lot of improved ACO algorithms were put forward to solve all kinds of NP-C problems, such as quadratic assignment problem (QAP)[6], job scheduling problem[7], multi-target tracking[8], dynamic manufacturing scheduling[9], vehicle routing problem[10] system identification[11] and so on.

For most of the ACO algorithms, ant individuals usually use pheromones deposited on the arcs to share information during the tour construction. By means of the strategy, ant colony can achieve the goal of co-evolution. The pheromones are often deposited on the arcs according to the people's intuitive cognition, and can be easily used to construct a mathematical model. However, Dorigo M once adopted nodes but not arcs to associate pheromone[12], when introducing a hyper-cube framework (HCF) for ACO. Although this HCF was well discussed as a general framework for all ACO algorithms, it was not involved in the comparison of the effects of depositing pheromones on the nodes and on the arcs. In literature [13], an improved ACO was proposed to solve the continuous space optimization problem based on a grid model, and it has put the pheromones on the discrete grid nodes. Chen[14] discussed a modified ACO algorithm depositing pheromones on the nodes. The algorithm introduced an adaptive strategy of pheromone evaporation factor to enhance the algorithm's

Manuscript received April 8, 2012;

doi:10.4304/jcp.8.3.585-593

performance, and was applied in robot path planning problem. All the above three literatures has adopted a similar method of depositing pheromones on nodes, but the generality of the adopted method for various requirements has not been particularly studied. At the same time, It has not been mentioned what influences will be resulted from the new method.

In this paper, the new method adopted in the above literatures is called pheromone marks strategy for convenience of the following study. The proposed new improved ACO algorithm called PM-ACO is based on the strategy and it has deeply studied the generality which has not been elaborated in the previous literatures. Some tests are implemented and have demonstrated that PM-ACO has a lower robustness compared with ACS. Further more, it analyzes how the ant population's evolution is influenced and the lower robustness is achieved under the pheromone marks strategy. In addition, for a better performance when applying PM-ACO in solving robot path planning problem, a regulation and controlling measure is performed to balance the pheromone distribution so that the algorithm can get rapid convergence and good robustness.

II. Robot Path Planning Based on ACO

Robot path planning problem is a basic issue of robot's navigation and control, the key of which is to find a collision free path from the starting point to the destination point to make robot bypass the obstacles safely. There are many conventional methods which are proposed to solve the robot path planning problem, such as Petri network algorithm, fuzzy programming based on data fusion, artificial potential field method, visible vertices of graph method and voronoi graph method and so on[15]. The computational efficiency of these methods is usually rather low. Therefore, several heuristic methods for the problem were designed to develop some solutions which have good performance, such as artificial potential field, fuzzy logic, neural networks, genetic algorithm, ant colony optimization and etc.[16].

A. Grid Model of Navigation Area

The navigation area of a robot is usually limited to the two dimensional space. A feasible solution of robot path planning problem is a curve in the navigation area, which connects the begin point and the destination point. As a result, constructing robot's navigation area model based on the method of surface grid conforms to the requirements of most practical problems. The navigation area discussed in this paper mainly includes the robot's working region and the known obstacles.

Suppose $ABCD$ is the limited oblong working region of the robot on the two dimensional plane, where there are some static obstacles $b_1, b_2, ..., b_n$ which are convex polygons. Constructing Cartesian coordinate system inside $ABCD$, taking the lower left corner of $ABCD$ as the origin of coordinates, the horizontal as the X axis and the vertical as the Y axis, supposing the maximum value of $ABCD$ on direction of X and Y is I and J. Suppose the step length of the robot's movement is one unit in each time, divide the region into I on direction of X and J on direction of Y equivalent portions respectively and the grid model of navigation area is described as the follows (see figure 1).

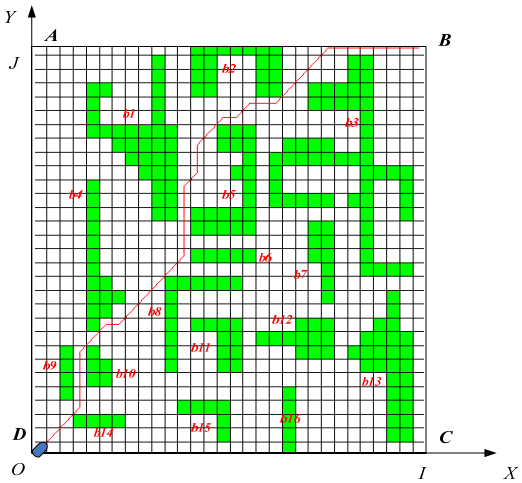


Figure 1. Grid model of robot's navigation area

Navigation area of the above diagram is divided into grid cells of 30×30, and there are 16 obstacles identified between b_1 and b_{16}. The lower left corner is the starting point of the task and the top right corner is the destination point. The grid cells colored green represent obstacles, while the grid cells colored white represent the free area of robot.

Now, searching the path for a robot in the navigation area is to combine a series of the grid cells, and this combination has the following characteristics:

1) The gird cells should be adjacent from each other.

2) The cells sequence should start with the original point O and end with the point B.

B. Robot's Moving Rules

The robot moves one grid cell per step and walks all around randomly till encounter the end point of the task. When a robot is in a grid cell, considering the robot can change direction freely, it moves forward complying with the following rules:

1) According to the grid model of navigation area, robot can move to 8 adjacent grid cells at the most, which is shown as figure 2.

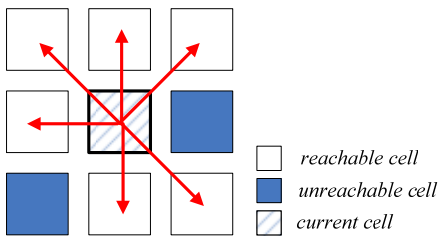
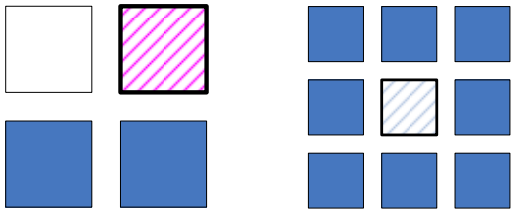


Figure 2. Rules of the robot's movement. Robot in the diagram can move to the 6 light adjacent grid cells but not to the dark ones which are obstacles.

2) When the current cell of robot is the end point of the task or the successive reachable cells are empty, the

movement of the robot is ended. If the current cell is the end point, grid cells visited by the robot should be stored, otherwise be given up, shown as figure 3.



(a) search succeed (b) search not succeed

Figure 3. Two situations when robot finishes the moving behaviors. The slash cell of figure (a) is the end point of the task, while the slash cell of figure (b) is not.

C. ACO Applied in Robot Path Planning Problem

To make use of ACO algorithms for robot path planning problem, grid cells are modeled as nodes, and arcs are created between adjacent cells. A robot is mapped to an ant, and its moving rules can be directly used to be applied in ants' transition rules. When an ant completes its tour, it constructs a safe path between the beginning point and end point. Through the cooperation of ant population, better paths can be found gradually till it achieves our requirements.

Moreover, for getting a solution more efficiently to the robot path planning problem, ACO algorithms should have the following features:

1) It has a taboo table to save the visited nodes, and ants can't visit the nodes in the taboo table even when it isn't an obstacle node.

2) The search of ants will be defeated under the following two situations, and the partial solution is discarded.

a) All the successive nodes of the current node have been visited.

b) The reachable successive nodes of current node are obstacle nodes.

Based on the changes according to the above indications, the ACO algorithm can be used to find the optimal moving path for a robot.

During the ant colony's tour construction, ants of different generations affect each other in constructing solutions, resulting in that later ant generations are affected by the pheromone accumulation deposited by the ancestor ants. The pheromone updating strategy determines the pheromone distribution, and it will exert influence on the evolution of the whole ant population. The nodes that have more pheromones will have more priority in the selection behaviors of the next ant generation, which can make the ant colony get a unified recognition about the problem.

III. ACO WITH PHEROMONE MARKS

When the pheromones are deposited on the arcs called arc pheromone strategy, the ant's searching behaviors will leave pheromones on the arcs called pheromone trails. Differently, if the pheromones are deposited on the nodes, a series of pheromone points are generated by ants called pheromone marks. In the pheromone marks strategy, an ant on a node selects the next node by comparing the pheromones on the reachable nodes but not on the successive arcs. Therefore, the improved ACO algorithm based on the pheromone marks strategy is called pheromone marks ACO.

A. Pheromone Marks ACO Algorithm

PM-ACO is on the basis of ACS[5], which has a good performance in solving many problems. PM-ACO modifies the pheromone depositing object from arcs to nodes, and keeps the pseudorandom proportional rule and global pheromone updating strategy.

AS is the first version of ACO algorithm, but it is inferior to state-of-the-art algorithms[17]. Compared with AS, ACS has the following improvements:

1) State transition rule changes from the random proportion rule to the pseudo-random proportional rule.

2) ACS updates local pheromone when constructing paths.

3) After each iteration, it does a global pheromone update on the iteration-best tour.

PM-ACO explores the ant colony's evolutionary computation pattern based on ACS, which has a global pheromone updating strategy depositing pheromone on the best-so-far tour's arcs and a local pheromone updating strategy depositing pheromone immediately on the arcs that ants traverse. The modifications of PM-ACO comparing to ACS are as follows:

1) The state transition rule is decided by the following formula:

$$j = \begin{cases} \arg\max\limits_{l \in J_k(i)} \left[(\tau_l)^{\alpha} \times (\eta_{il})^{\beta} \right], & if\ q \le q_0 \\ J & ,\ otherwise \end{cases} \tag{1}$$

Where j is the next node, τ_l is the pheromone strength of node l, α is the pheromone trail heuristic factor, β is the heuristic factor, η_{il} is the distance heuristic function, q is a random variable uniformly distributed in [0,1], q_0 is a parameter, and J is the transition probability defined in formula (2):

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_j^{\alpha} \times \eta_{ij}^{\beta}}{\sum_{u \in allowned} \left[\tau_u^{\alpha} \times \eta_{iu}^{\beta} \right]}, & if \quad j \in allowned \\ 0, & otherwise \end{cases} \tag{2}$$

Where p_{ij}^{k} is the transition probability of ant k from node i to node j. The formula has some changes compared to the state transition rule in HCF-ACO[12]. The distance heuristic function η_{il} isn't associated to nodes, and it can preserve the heuristic information derived from the distances.

2) The global pheromone updating rule is defined as follows:

$$\tau_i = (1-\rho) \times \tau_i + \Delta\tau_i \tag{3}$$

Where ρ is the evaporation factor, $\Delta\tau_i$ is the pheromone increment, and is calculated by the following formula:

$$\Delta\tau_i = \begin{cases} \dfrac{1}{L_{iterbest}} & , \ if \ i \in best\ tour \\ 0 & , \ else \end{cases} \tag{4}$$

Where $L_{iterbest}$ is the length of the path constructed by the iteration-best ant.

3) The local pheromone updating rule is as follows:

$$\begin{cases} \tau_i = (1-\rho)\times\tau_i + \Delta\tau_i \\ \Delta\tau_i = \tau_0 \end{cases} \tag{5}$$

Where τ_0 is the initial pheromone strength on the nodes. The following codes are the pseucodes of PM-ACO (see figure 4).

Algorithm PM-ACO
$iter \leftarrow 0$;
initialize α, β, τ_0, Q, ρ;
for each node **do**
  $\tau_{ij}^{iter} \leftarrow \tau_0$;
end
repeat *iter*
  initialize *path, tabu*;
  $f(path) \leftarrow 0$;
  for ant k **do**
    repeat constructing tour
      calculate P_{ij}^k by formula (2);
      select j by formula (1);
      $path_k \leftarrow path_k \cup (i, j)$;
      $tabu_k \leftarrow tabu_k \cup j$;
      pheromone updating by formula (5);
    end constructing tour
  end
  calculate $f(path_k)$;
  calculate the pheromone increments according to the formula (4);
  pheromone updating by formula (3);
end

Figure 4. The pseucodes of PM-ACO

Ants aren't essential to cross all nodes in a tour as the covered problem. If crossing all nodes in a tour, the deposited pheromones on the nodes will equal with each other, resulting in a similar pheromone density all the time. The robot path planning problem doesn't need the ants to cross all grid cells in a tour, so that PM-ACO can be used to solve it in theory.

B. Stability Analysis of PM-ACO

To solve the dissymmetrical path planning problem which has n nodes applying the arc pheromone strategy, it is essential for a $n \times n$ matrix to store the pheromone values, which makes the space complexity is $O(n^2)$. Applying the pheromone marks strategy, just a matrix of n dimensions is needed to store the pheromone values, the space complexity will reduce to $O(n)$. However, the above advantages can not be presented in a simple problem which has a bit of grid cells.

The symmetrical double bridge experiment[18] and dissymmetrical double bridge experiment[19], which are implemented with an arc pheromone strategy as the above, demonstrate that most of ants prefer to choose the arcs which have more pheromones preserved(see figure 5). The different possibility of selection of branches is obtained through the difference of deposited pheromone strength. Differential pheromone density associated with arcs results in the tendentiousness of random walk of the ants. And the tendentiousness makes the ants' choice of each stage more likely concentrate to the arcs with higher pheromone density. With the accumulation of pheromone, ant colony's searching behaviors will come along to follow one path steadily.

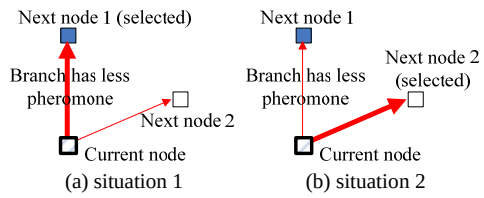


Figure 5. Selection based on path pheromone trail

Under the pheromone marks strategy, pheromone marks associated to a node contain the knowledge gained from all the prior nodes' selections rather than from the experience of the one node's historical decisions. Compared with the arc pheromone strategy, the pheromone marks of a node is a comprehensive embodiment of several arcs' pheromones (see figure 6). Therefore, when an ant on a node is making decision, it is influenced not only by the ancestor ants that have passed this node but also by the ancestor ants that have passed the next node (see figure 7).

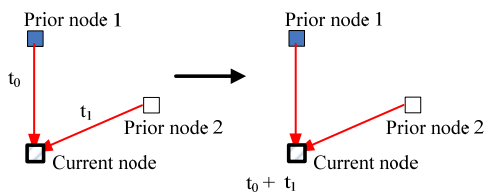


Figure 6. Pheromone mark of the current node means integration of two branches' pheromones

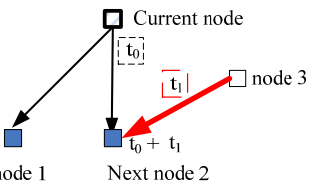


Figure 7. Ants' selection at current point is influenced by not only the ants which passed the Current node and arrived in the Next node 2 (t_0),

but also by the ants which passed the node 3 and arrived in the Next node 2 (t_1)

Because of the integration of pheromones, the nodes updated in each iteration can get a considerable increment of pheromones, which makes it easy to produce a differential pheromone density compared with the other nodes. If the nodes updated in each iteration are basically unalterable, the rapid pheromone accumulation can result in a more quick convergence. In fact, although the ant often selects the successive node based on a rand proportion rule, it can't be excluded the possibility that a lot of ants will therefore select some nodes not in the previous updated nodes. As a result, under the pheromone marks strategy, the variation of pheromone density is violent especially with the local pheromone updating method.

IV. PM-ACO WITH PHEROMONES DEPOSITED ON THE RANK-BEST NODES

Ants in each iteration always cross a certain amount of same nodes no matter how different their tours are. The more quantity of ants passes a node, the more important is the node in generating the best-so-far tour. Therefore the number of traversed ants represents a partial rough understanding of the solution of the problem. A node where an increasingly amount of ants cross denotes a more likely component of the best-so-far solution. In other words, the more ants cross a node, the more pheromones will be left on the node, and it is more possible to influence the behavior of the successive ant colony by means of pheromone marks.

By counting the ants that pass the nodes, a profile of the importance of the nodes where ants traversed is defined. Based on the information expressed in the profiles, a global pheromone updating mechanism called r-best nodes rule is designed to adjust and control the distribution of pheromones. It is mainly achieved by paying an emphasis to the partial nodes of the tours, which denote the more important components belonging potentially to the optimal path.

A. Pheromones Deposited on r-best Nodes

In each iteration, all the nodes are sorted by the statistical measures of the amount of ants traversing a node, and we define the rank-best nodes as those nodes which are near the front. Suppose the number of extracted rank-best nodes is r, define those nodes as r-best nodes and name the corresponding rule as a r-best nodes rule.

During the search iteration by a certain ant generation, after all ants have constructed a tour, the nodes are sorted by the number of traversed ants, and the pheromone marks will be updated. The main procedures of r-best nodes rule are as follows:

1) Set the counter matrix of the traversed ants to zero matrix.

2) Ants select nodes to construct the solution, when one node is selected, the counter of the node pluses one.

3) Sort the counter matrix's elements after all the ants finish constructing the solution.

4) Update the pheromone marks according to the extended pheromone update strategy and clear the counter.

The extended pheromone updating strategy affects the best ranked nodes as follows:

$$\tau_i = \tau_i + \tau_0 \tag{6}$$

Where τ_i is the pheromone mark's strength of node i, and τ_0 is the initial pheromone strength on the nodes.

The pheromone mark strategy somewhat resembles the local pheromone updating strategy. It takes all the ant-traversed subpaths into account, and when an ant passed a node, there is a quantity of pheromones deposited on the node. Unlike what local pheromone updating strategy does, the r-best nodes rule only counts the number when an ant crosses a node, and a pheromone update is carried out after all ants complete construction. The pheromone deposited on a subpath is proportional to the number of ants that traverses. According to the formula (6), the pheromone amount deposited on a node has nothing to do with the number of traversed ants. That is to say, the counted number of traversed ants is just for sorting the nodes.

If the ant is k, the number of nodes is $I \times J$, nodes are sorted and the first r nodes are selected to update pheromone. The following codes are the pseucodes of the r-best nodes rule (see figure 8).

```
Procedure r-best nodes rule in each iteration
for each node do
    counter_j ← 0, 0 < j ≤ I × J ;
end
for ant k do
    repeat constructing tour
        calculate P_ij^k by formula (2);
        select j by formula (1);
        counter_j = counter_j + 1 ;
    end constructing tour
    sort counter_j ;
    select the first r nodes for pheromone updating
    according to formula (6);
end
```

Figure 8. The pseucodes of the r-best nodes rule

B. Adaptive Transition of r-best Nodes

After adding an r-best nodes rule, the pheromone increasing procedure of nodes becomes smooth. The algorithm can escape from the local minimum more effectively, and the global exploration ability of PM-ACO at the initial stage is improved. However, it is paradoxical for the two capabilities of an algorithm to be updated at the same time. That is to say a good global exploiting capability will lead to a worse convergence, conversely as.

When pheromone marks gradually concentrates to the best-so-far tour, the pheromone can accelerate this procedure so as to quicken the convergence. To explore globally better at the initial stage, it is necessary for the pheromone marks to influence even more nodes. But, if the number is too big, it can advance the convergence on the later stage, which will influence the steadiness of PM-ACO. Therefore, the initial value r in r-best nodes and the procedure of change are key factors to impact the control of the global pheromone distribution by the updating strategy.

At the first stage, that a lot of nodes are involved in the r-best nodes rule makes it easy to distribute pheromones in a large scale scope, which denotes a good global exploiting capability. At the same time, the fewer nodes are involved in, the lower possibilities will the ants select the nodes not belonging to the best-so-far path. Therefore, gradually decreased number of nodes involved in pheromone updating can effectively balance the pheromone distribution at different stages and the value of r is updated by follows:

$$r = r_0 - \lambda \times iter \tag{7}$$

Where r is the number of nodes that participates in the pheromone marks update in each iteration. And r_0 is its initial value, λ is the cyclic reduction, and $iter$ is the serial number of each iteration. The setting of r_0 and λ has influence on the PM-ACO and should be treated seriously in practice. Besides, the setting of the minimum value of r, denoted by $r_{\min}$, can also have certain impact on the PM-ACO.

By the adaptive strategy, PM-ACO can adjust the pheromone arrangement steadily between the initial and latter stage, and the r-best nodes rule can better benefit the global exploration on the initial stage and achieve rapid convergence on the latter stage.

V. Local Optimization Strategies

The local optimization strategies considerate about shortening the time wasted in blindly searching and getting a more optimal path. An azimuth guiding rule acts in an early stage, and a one step optimization rule acts in the end of each iteration.

A. Azimuth Guiding Rule

On the initial stage of the algorithm, when ants fail in searching for a complete path, the behavior of ants searching for the end point of the task is totally blind. Thus, at the beginning most of the time is wasted on constructing the first complete path, which is not conducive to solving the problem quickly. Therefore, according to the fact that the end point of the task is generally known, an azimuth guiding rule is introduced to accelerate the convergence of the algorithm. This rule is accomplished by setting a global azimuth angle, which is as follows:

1) Set 8 discrete azimuth angles $0°$, $45°$, $90°$, $135°$, $180°$, $225°$, $270°$, $315°$ around a node corresponding respectively to the 8 adjacent nodes where the robot can move to.

2) When the algorithm begins to initialize, calculate the relative azimuth angle between the starting point and the end point and select an azimuth among the eight discrete angles as the global azimuth angle according to the nearest neighbor rule.

3) During the calculation of transition probability, do the $+\xi$ operation to the pheromone of the node whose azimuth to the current node is the global azimuth angle.

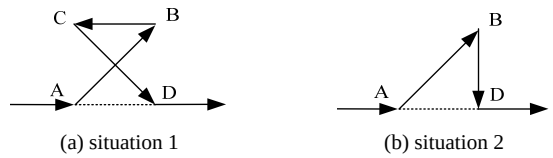
Where ξ represents a fixed pheromone increment. Through the static azimuth guiding rule, the time of the algorithm constructing the first feasible solution is shortened and the efficiency is improved. The rule acts before getting a first feasible solution, and disables after that. The pseucodes of the rule are as figure 9.

```
Procedure azimuth guiding rule
 iter ← 0;
 bValidpath ← 0 ;%a state variable indicating whether
 a valid complete path is constructed
 Set global azimuth angle θ_g;
 repeat iter
    for ant k do
        repeat constructing tour
            if θ_j = θ_g and bValidpath = 0
                τ_j = τ_j + ξ;
            end
            calculate P_ij^k by formula (2);
        end constructing tour
        if getting a complete path
            bValidpath ← 1;
        end
    end
 end
```

Figure 9. The pseucodes of the azimuth guiding rule

B. One Step Optimization Rule

In the tour construction, the pheromone integration effect can result in a rapid pheromone accumulation on a node, and it usually causes the absolutely differential pheromone strength. The situation will interrupt the local optimization searching, and make the solution easily have the following characteristics: crossing situation and triangle situation (see figure 10).



(a) situation 1 (b) situation 2

Figure 10. Situations need local optimization. (a) The constructed solution is a crossing situation. (b) The constructed solution is a triangle situation. If the partial solution can be replaced by the dotted line AD, the solution can be improved.

In the above situations, a one step reachable relation of two nodes takes three or four steps, and it wastes some steps of the robot's movement. It can be detected and eliminated by the following steps:

1) Visit all the nodes of a solution from the first node to the end.

2) Check whether the node has direct link to another. switch to 3) if it has, and switch to 2) if not.

3) Extract the subsequence between the two nodes and calculate the length. If the length is longer than the length of the direct link, replace the subsequence with the two nodes and switch to 11).

4) End the algorithm.

Procedure one step optimization rule
$i \leftarrow 1, 0 < i < length_{iteration-best} - 1$;
while node $i < length_{iteration-best} - 1$ **do**
  $j \leftarrow i+1, 0 < j \le length_{iteration-best}$;
  $pos \leftarrow j$;
  set s_p the partial solution to be replaced;
  for node j **do**
    $s_p \leftarrow s_{ij}^p$;
    if i is an adjacent node to j
      $pos \leftarrow j$;
    end
    $j \leftarrow j+1$;
  end
  replace s_{ipos}^p with (i, pos);
  update $length_{iteration-best}$;
  $i \leftarrow i+1$;
end

Figure 11. The pseucodes of the one step optimization rule

The processing of the one step optimization rule has the complexity of $o(n!)$. It will take a few minutes to run it in the ants' constructing tour, and make a neglect impact on the performance of PM-ACO. As a result, it is designed to works only at the end of each iteration.

VI. Experiment Tests

To test the performance of the PM-ACO, three groups of tests are designed to be carried out in the following environment: a notebook HP540 running Windows XP with CPU T5470 of 1.6GHz and memory of 2G, and the programs are implemented with MatLab7.8.0 (R2009a). The parameters of PM-ACO is set as follows: $m = 30$, $I \times J = 30 \times 30$, $\tau_0 = 1000$, $Q = 500$, $\rho = 0.8$, $\alpha = 3$, $\zeta = 0.2$, $\xi = 5$.

1) Compare the stability between PM-ACO and the algorithm in the literature [14] (ACO2 for short).

ACO2 also associated the pheromone to the nodes, and is applied in the robot path planning problem. Therefore, ACO2 has a similar structure and object to PM-ACO, and it has compared with several well-known ACO algorithms. We implement ACO2 again in the above environment and don't change any parameters of it.

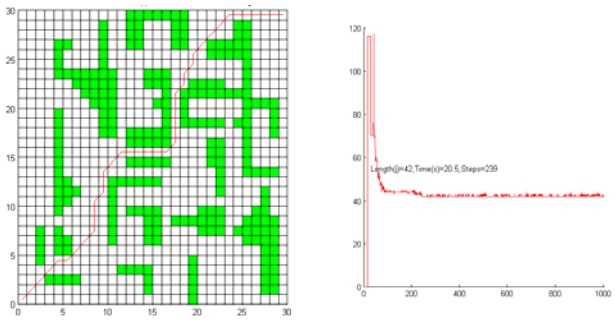
2) Test the exploring ability and convergence of PM-ACO with azimuth guiding rule and PM-ACO without azimuth guiding rule in a navigation area of 30×30.

3) Test the exploring ability and convergence of PM-ACO with azimuth guiding rule and PM-ACO without azimuth guiding rule in a navigation area of 50×50.

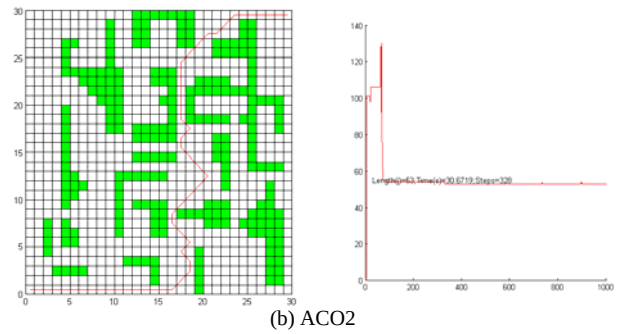
Note that the length of constructed path is set to 0 when the ants have not gotten a complete valid path in an iteration.

A. Tests Group1: Experiment of Stability of PM-ACO When Compared to ACO2

To examine the stability of PM-ACO compared with ACO2, tests are performed in the environment which has 30×30 grid cells. The results are showed in figure 12.



(a) PM-ACO



(b) ACO2

Figure 12. The results of PM-ACO compared with ACO2. When PM-ACO operates to the 239 iterations, it reach the global optimal solution at the first time. The value is 42, and it takes 20.5 seconds. When ACO2 operates to the 328 iterations, it reach the global optimal solution at the first time. The value is 53, and it takes 30.6719 seconds.

The experiment is executed 10 times with the same parameters in a same environment. The results are listed in table 1.

TABLE 1

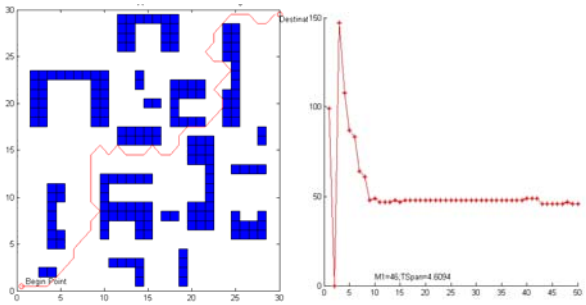
TEN RESULTS OF PM-ACO AND ACO2

ID	The length of the global optimal path		The iterations when getting the optimal solution at the first time		Time cost when getting the optimal solution at the first time (s)	
	PM-ACO	**ACO2**	**PM-ACO**	**ACO2**	**PM-ACO**	**ACO2**
1	46	52	42	120	3.4	11
2	42	55	134	83	12.7	7.7

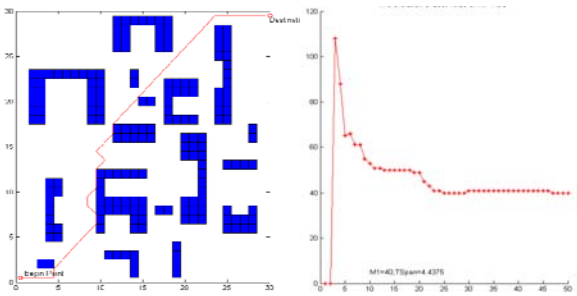
3	41	61	77	179	6.7	17
4	46	49	113	129	10.4	11
5	43	48	50	167	4.5	14
6	46	53	95	173	7.7	17
7	46	51	95	135	8	12
8	48	61	74	44	6.8	3.5
9	48	47	52	81	4.8	6.7
10	58	47	135	48	12.6	3.8
AVG	46.4	52.4	81.7	115.9	7.4	10.4
MIN	41	47	42	44	3.4	3.5
MAX	58	61	135	179	12.7	17

It can be concluded from the figure 12 and table 1 that PM-ACO, compared with ACO2, has a better global exploiting capability, better robustness and better convergence. The adaptive strategy makes PM-ACO have a balanced pheromone distribution at the first stage, and rapidly concentrate the nodes around the best-so-far path. Furthermore, PM-ACO has a good performance because of benefiting the azimuth guiding rule and an easily implemented local optimization strategy, which is called one step optimization rule.

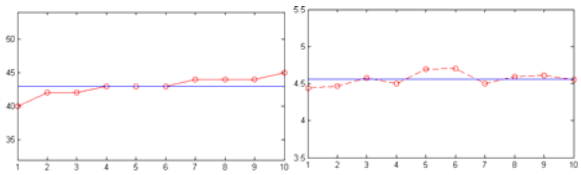
B. Tests Group2: Experiment of PM-ACO with Azimuth Guiding Rule and PM-ACO without Azimuth Guiding Rule in Navigation Area of 30×30 *Grid Cells*



(a) The length of the shortest path by PM-ACO without the azimuth guiding rule is 46 jumps, and it takes 4.6094 seconds. Because of the azimuth guiding rule is disabled, ants search all the azimuth angles with a same probability. They move ahead blindly in the initial iterations, and it makes the ant colony is difficult to find a complete path between beginning node with end node. At the same time, ant colony can't concentrate to an important azimuth angle to searching particularly. As a result, PM-ACO without the azimuth guiding rule is difficult to find a global best path close to the global optimal path.



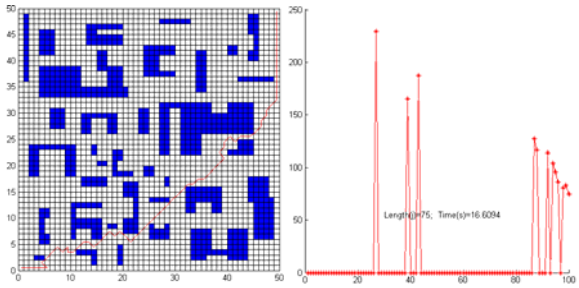
(b) The length of the shortest path by PM-ACO with the azimuth guiding rule is 40 jumps, and it takes 4.4375 seconds. Because of adapting the azimuth guiding rule, the length of the shortest path is reduced from 46 to 40 jumps, and the time costs is reduced from 4.6094 to 4.4375 seconds.



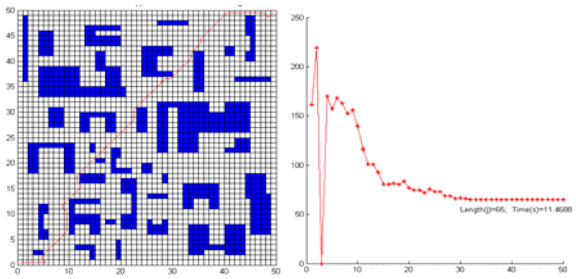
(c) Results curve of 10 tests of PM-ACO with the azimuth guiding rule. The left is the curve of the global best values of 10 tests and the right is the curve of the time cost of 10 tests. The values are all floating up and down the average value respectively. It is concluded that PM-ACO is robust.

Figure 13. Performances of PM-ACO with the azimuth guiding rule and PM-ACO without the azimuth guiding rule in Navigation area of 30×30 Grid Cells.

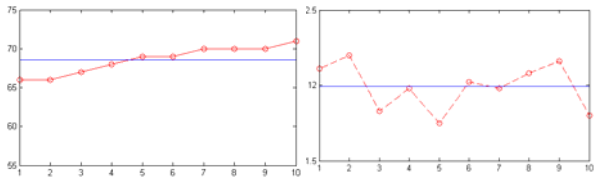
C. Tests Group3: Experiment of PM-ACO with Azimuth Guiding Rule and PM-ACO without Azimuth Guiding Rule in Navigation Area of 50×50 *Grid Cells*



(a) The length of the shortest path by PM-ACO without the azimuth guiding rule is 75 jumps, and it takes 16.6094 seconds. In most of the iterations, PM-ACO fails to find a complete path because the problem has a large scale compared with the problem which has 30×30 grid cells. Without a global azimuth to leading the ants concentrate to a more important searching adjacent node, PM-ACO becomes unstable and can't get a correct path in some times.



(b) The length of the shortest path by PM-ACO with the azimuth guiding rule is 65 jumps, and it takes 11.4688 seconds. Because of adapting the azimuth guiding rule, the length of the shortest path is reduced from 75 to 65 jumps, and the time costs is reduced from 16.6094 to 11.4688 seconds. In addition, PM-ACO with the azimuth guiding rule can get a complete path except the 3th iteration, and it can achieve convergence smoothly.



(c) Results curve of 10 tests of PM-ACO with the azimuth guiding rule. The left is the curve of the global best values of 10 tests and the right is the curve of the time cost of 10 tests. The values are all floating up and down the average value respectively. However, the amplitude of

floating in this circumstance is bigger than the circumstance with 30×30 grid cells.

Figure 14. Performances of PM-ACO with the azimuth guiding rule and PM-ACO without the azimuth guiding rule in Navigation area of 50×50 Grid Cells.

VII. Conclusion

In this paper we proposed a new improved ant colony optimization algorithm which is abbreviated to PM-ACO and its application in solving the robot path planning problem. In PM-ACO, ants are allowed to deposit pheromone on the nodes but not on the arcs. The nodes include the iteration-best tour's nodes and the statistically rank-best nodes. The pheromone update strategy derived from ACS leads to an integration effect of mutual contributions, resulting in a rapid concentration of pheromone marks. It makes the algorithm have a bad robustness. To lower the negative influence of the pheromone mark strategy, an r-best nodes rule is introduced by counting the numbers of ants traversing the nodes in each iteration. For a better performance, some local optimization strategies including an azimuth guiding rule and a one step optimization rule are proposed. Experimental tests show that the PM-ACO algorithm is a promising heuristic method for achieving a shortest path for a robot.

In conclusion, the PM-ACO brings to us a new light to the ant colony's knowledge expression and evolutionary pattern. We will focus the future work on the algorithm's stability research and numerical analysis under the pheromone mark strategy.

References

[1] Colorni A , Dorigo M, Maniezzo V, etal. "Distributed optimization by ant colonies". Proceedings of the 1st European Conference on Artificial Life. Paris, 1991, 134-142.

[2] Dorigo M, etal. "Positive feedback as a search strategy". *Technical report 91-016*, Department of Electronics, Politecnico diMilano, Italy, 1991.

[3] Dorigo M. "Optimization, Learning and Natural Algorithms". Ph.D. Thesis, Department of Electronics, Politecnico diMilano, Italy, 1992.

[4] Gambardella L M, Dorigo M. "Ant Q: a reinforcement learning approach to the traveling salesman problem." Proceedings of the 12th International Conference on Machine Learning. 1995, 252-260.

[5] Dorigo M, Gambardella L M. "Ant colony system: a cooperative learning approach to the traveling salesman problem". IEEE Transactions on Evolutionary Computation. 1997 1(1):53-66.

[6] Maniezzo V, Colorni A, Dorigo M. "The ant system applied to the quadratic assignment problem". Technical report IRIDIA/94-28, IRIDIA, Universit é Libre de Bruxelles, Belgium, 1994.

[7] Gao Q L, Luo X, Yang S Z. "Stigmergic cooperation mechanism for shop floor control system". International Journal of Advanced Manufacturing Technology. 2005, 25(7-8):743-753.

[8] Ali Onder Bozdogan, Murat Efe. "Improved assignment with ant colony optimization for multi-target tracking". *Expert Systems with Applications.* 38, 2011, 9172-9178.

[9] W. Xiang, H.P. Lee. "Ant colony intelligence in multi-agent dynamic manufacturing scheduling". Engineering *Applications of Artificial Intelligence.* 21, 2008, 73-85.

[10] John E. Bella, Patrick R. McMullen. "Ant colony optimization techniques for the vehicle routing problem". *Advanced Engineering Informatics.* 18, 2004, 41-48.

[11] Wang L, Wu Q D. "Linear system parameters identification based on Ant system algorithm". Proceedings of the IEEE Conference on Control Application. 2001, 401-406.

[12] Blum C, Dorigo M. The hyper-cube framework for ant colony optimization. IEEE Transactions on Systems, Man and Cybernetics—Part B; to appear. Also available as Technical Report TR/IRIDIA/2003-03, IRIDIA, Université Libre de Bruxelles, Belgium; 2003.

[13] Gao S, Zhong J, MO S J. Research on ant colony algorithm for continuous optimization problem. Microcomputer Development, 2003, 13(11):12-13.

[14] Chen X, Yuan Y. Novel ant colony optimization algorithm for robot path planning. Systems Engineering and Electronics, 2008, 30(5):952-955.

[15] Abe Y, Shikann M, Fokuda T, etal. "Vision based navigation system by variable template matching for autonomous mobile robot". Proceedings of the IEEE International Conference on Robotics & Automation. Leaven, 1998, 952-957.

[16] Weerayuth N, Chaiyaratana N. "Closed-loop time-optimal path planning using a multi-objective diversity control oriented genetic algorithm". IEEE International Conference on Man and Cybernetics. 2002, 7 pp. vol6.

[17] Dorigo M, Stützle T.: Ant Colony Optimization. The MIT Press, Cambridge, Massachusetts London, England. 2004, 25-221.

[18] J.-L. Deneubourg, S. A., S. Goss, etal. "The Self-Organizing Exploratory Pattern of the Argentine Ant". *Journal of Insect Behavior.* 1990, 3(2): 159-168.

[19] S. Goss, S. A., J.L. Deneubourg etal. "Self-organized shortcuts in the argentine ant". *Naturwissenschaften.* 1989, 76: 579-581.

Xiangyang Deng was born in 1981.He is working for PhD degree in information and communication engineering at Naval Aeronautical and Astronautical University. His research interest is swarm intelligence and complex system simulation.

Limin Zhang was born in 1966. He is a professor of Naval Aeronautical and Astronautical University. His research interest is artificial intelligence and complex system modeling.

Lan Luo was born in 1983. She got her Master Degree from York University, and is a lecturer of Yantai Vocational College. She is interested in complex system.