

# BASED ON ANT COLONY ALGORITHM TO SOLVE THE MOBILE ROBOTS INTELLIGENT PATH PLANNING FOR AVOID OBSTACLES

GUO Yue<sup>1</sup>, SHEN Xuelian<sup>1</sup>, ZHU Zhanfeng<sup>1</sup>

Management Engineering Institute, Ningbo University of Technology, No.201 Fenghua  
Road, Ningbo. 315211 Zhejiang, China

## **ABSTRACT**

*With the development of robotics and artificial intelligence field unceasingly thorough, path planning for avoid obstacles as an important field of robot calculation has been widespread concern. This paper analyzes the current development of robot and path planning algorithm for path planning to avoid obstacles in practice. We tried to find a good way in mobile robot path planning by using ant colony algorithm, and it also provides some solving methods.*

## **KEYWORDS**

*Mobile robots; Path planning; Avoid obstacles; Ant colony algorithm*

## **1. INTRODUCTION**

The research of mobile robot started from the late 1960s. The Stanford Institute successfully developed the autonomous mobile robot—Shakey robot in 1966. The robot has independent reasoning, planning, control and other functions in complex conditions with the application of artificial intelligence. At the end of the 1970's, the application of computer and sensor technology researches on mobile robot reach to a new high tide as a result of the development. The mid 1980's, a large number of world famous company started to develop mobile robot platform. The mobile robot is mainly used as the mobile robot experiment platform in university laboratories and research institutions, and promoting the multi-directional learning of the mobile robot. Since the 1990s, the symbol of environment information sensor and information processing technology development of high level, high adaptability of mobile robot control technology, programming technology under the real environment has emerged, and the higher level research of mobile robotics able to be conducted. In recent years, mobile robots are widely used in space exploration, ocean development, atomic energy, factory automation, construction, mining, agriculture, military, and service, etc. Research on mobile robot has become a hot research issue and the concern of the international robot.

Intelligent mobile robot is a set of integrated system of multiple functions which consists of environment perception, dynamic decision-making and planning, behavior controlling and executing. In recent years, mobile robot has wide application prospect in space exploration, ocean development, atomic energy, factory automation, construction, mining, agriculture, military, and service, etc. China started the research on the intelligent robots later than some developed countries, and there still existed a big gap within China and developed countries. In recent years,

the research theory and method for robot have reached the international advanced level by the China Robotics Lab, and achieved a number of important scientific research achievements in robotics frontier exploration and demonstration application etc. Because the state and society paid much attention on the robot field, which including: all-weather 120Kg suspended wing UAV system, polar research snow mobile robot, Ling Lizard-anti-terrorism and anti-riot robot, robot nano operating system, etc.

Mobile robot's path programming technology is one of the core technology in the field of robot research, which study of the algorithms is advantageous to the improvement of robot planning to meet the needs of practical applications. The path programming is that, in the obstacle environment, according to a certain evaluation standard, finding collision free path from the initial state to the target state. The main issues include finding the optimal or approximate optimal one from the initial state to the target state collision through the free path and an algorithm to built reasonable model by using of the mobile robot environmental information. In the model which being able to cope with uncertain factors and path tracking errors in the environment, making the influence of external objects to the robot reduced to a minimum: how to use all the information known to guide the robot motion, resulting in a better decision [1] .

## 2. LITERATURE REVIEW

The traditional method of path programming is carried out simulation test based on graph. The general approach is based on the global path planning. At present domestic and foreign common methods include grid method, topology, visibility graph, Voronoi graph, method, the artificial potential field method, A\* algorithm etc.

Grid method proposed by Howden in 1968 [2] , decomposing the robot planning space into a number of information network unit working space is divided into unit after the use of heuristic algorithm to search the safe path in the unit. [3] The search process always uses work space with four quadtree or octree. Consistency and standards grid makes simple adjacency relation in raster space. After giving each grid traffic factor, path planning problem turns into a problem of searching optimal path with two grid nodes in the grid network. Topological method is mainly divided space with topological feature subspace, and then look for topological path is the starting point to the target point based on the topology of the network, and finally find the path geometry by the topological path.[4] The basic idea of the method is to find paths in high dimensional space transformed into the problem of determining the connectivity of the problem of Low Dimensional Topology space.[5] The visibility graph method is a kind of configuration space method,[6] it mainly regards the robot as a particle processing, expanding the boundary of the corresponding outward obstacles in the work environment, and the boundary is formed with vertices of polygons, determining its vertices, including the robot starting point and the target point. These points connect, but each vertex cannot connect across each other, forming a visibility graph.

The Voronoi plot method is first discovered by the Russian mathematician Voronoi which can be applied to static random environment, which is to say in the process of robot running, the environment is static.[7] All the obstacles are motionless, but environment is uncertain before the robot starts its path planning, the size and the location parameter of the obstacles in the environment is changeable. This method using the path which may be far away from obstacles to show walls arc, that results the path will increase from the initial node to the target node. Artificial potential field method is a method of local path planning, proposed by Khatib etc.[8] The basic idea of the method is regarding the motion of the robot in the environment as a virtual artificial force field motion.[9] Obstacles generate repulsive force on the robot, and attraction on

the target. The joint force of attraction and repulsive force controls robot motion direction, which will determine the position of the robot[10].

In recent years, with the rapid development of in-depth research and the modern computing technology in mobile robot path programming, the traditional path programming is hard to meet its requirements and failure in meeting the need of actual environment changes. Therefore intelligent path algorithms have been studied and used in robot path programming widely. The artificial intelligent path programming algorithm improves the accuracy of robot obstacle avoidance path programming greatly and accelerates the programming speed, all these are to be met the needs of practical application. Intelligent path programming algorithm includes genetic algorithm[11][12], particle swarm algorithm[13], fuzzy logic[14][15], neural network[16][17], artificial immune algorithm[18] and hybrid algorithm[19][20][21]. The above algorithms have been made certain achievements for the robot protecting the obstacle in known or unknown circumstances.

Italian scholar Dorigo and Coloni proposed a heuristic optimization algorithm in 1991, which is biologically inspired.[22] It simulated and reference the behavior of ants in the real world to solve combinatorial optimization problems under distributed environment.[23] It also solves the problems of large cost when robot in complex environment contains a large number of irregular obstacles in the path programming[24].

Ant colony algorithm is produced to simulate the process of ants foraging. Ants release specials in the search of path when confronted with a no through road, they will randomly select one while releasing hormone information of path length. When the ants again encountered this intersection, optimal path on the pheromone concentration increase, while the other pheromone concentration is cutting with the passage of time.[25] At the same time, the ants can adapt to changes in the environment when obstacles emerge, they will find an optimal path to go. Ant colony algorithm has the features of group cooperation, positive feedback and distributed computing. Group cooperation is a cooperation for better optimization task. Although each artificial ant can build a solution, but the solution with high quality is always produced by ant colony cooperation. The feedback mode of the algorithm is used in the optimum solution which leaves more pheromone on a path, and more pheromone in turn attracts more ants. The positive feedback process guides the system towards the optimal solution of the evolution direction. Distribute Computing of Ant colony algorithm can calculate each artificial ant at the multiple points in the problem space, at the same time, it began to separate the structural problems of solutions. The result will not be affected only because of one artificial ant cannot successfully obtained the solution Distributed Computing makes the algorithm easy to be implemented.[26] These characteristics make the ant colony algorithm suitable for solving complex combinatorial optimization problems.

The path programming is a kind of combinatorial optimization problem, so the ant colony algorithm is suitable to solve the path planning problems.

### **3 QUESTIONS**

#### **3.1 Problems in design**

In a  $800 \times 800$  planar scene graph, there is a robot at the origin of  $O(0, 0)$ , which can only activities in the planar scene range. The 12 different shapes of the regions are the obstacles that the robot cannot collide The description of the mathematical are as shown in the following table:

Table 1. Planar scene graph description

Region No.	Obstacles' name	vertex coordinates at left corner	Other description of the characteristics
1	square	(300, 400)	length of 200
2	circular		center coordinates (550, 450) a radius of 70
3	parallelogram	(360, 240)	Base length140, the coordinates of the vertices on the left top(400, 330)
4	triangles	(280, 100)	top vertex coordinates (345, 210), the lower right vertex coordinates (410, 100)
5	square	(80, 60)	length 150
6	triangles	(60, 300)	top vertex coordinates (150, 435), the lower right vertex coordinates(235, 300)
7	rectangular	(0, 470)	length 220, width 60
8	parallelogram	(150, 600)	Base length 90, the coordinates of the vertices on the left top(180, 680)
9	rectangular	(370, 680)	length60, width120
10	square	(540, 600)	length130
11	square	(640, 520)	length80
12	rectangular	(500, 140)	length300, width60

### 3.2 Maintaining the Integrity of the Specifications

Specify a point outside the obstacles as the target for the robot to reach (target point and the distance to the obstacle are at least more than 10 units). Set the rule that walking path of the robot should be by lines and arcs. The arc parts are robot's turning path. A robot cannot turn by line. The turning path consists with straight path tangent to a circle, and can also be composed of two or more circular arc path, but the minimum radius of each circular arc path is 10 units in order not to collide with obstacles. It also requires the distance between the robot walking route and obstacle is no more than 10 units, or a collision will occur and the robot can not complete the straight walking. 4 point O in the scene graph(0, 0), A(300, 300), B(100, 700), C(700, 640).

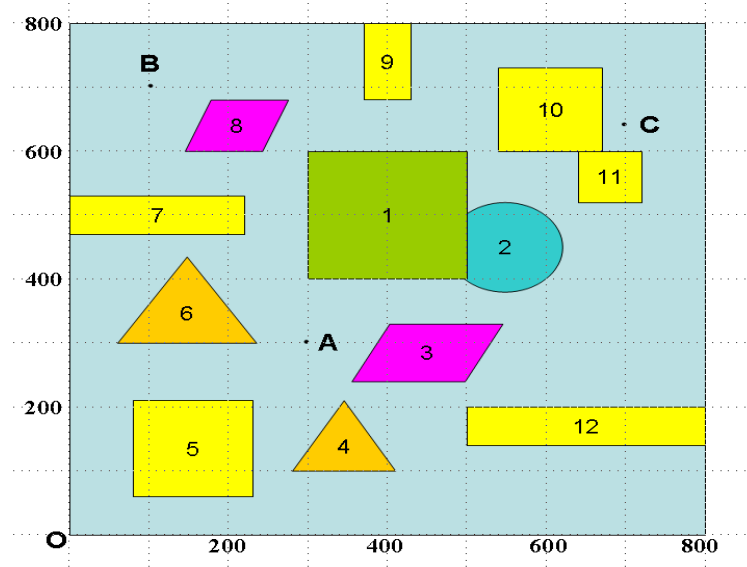


Fig1. 800 × 800 planar scene graph

The maximum speed of the robot walking straight into 5 unit / sec. and maximum turning speed

is  $v = v(\rho) = \frac{v_0}{1 + e^{10 - 0.1\rho^2}}$  where P is radius of turn . If the speed is higher than that, the robot will rollover, and is unable to walk.

**Question: Start from O (0, 0) ,which is the shortest path of O → A, O → B, O → C?**

#### 4 METHODOLOGY

In order to find the shortest path to the target from (0,0) with certain rules walking around obstacles ,we can draw the envelope of robot walking hazardous area, just around the corner with a radius of 10 units a quarter arc, by the method of the rope then to find the shortest possible path (for example, seek the shortest path between O and A, can be connected to the section of rope between O and A, to the arc of the corner support taut, then the length of this wire is O to a shortest possible path (A),and then list shortest path possible paths to each target point with Brute-force method.

Designated O (0,0) after the middle of a number of points around obstacles to reach the target point according to certain rules in the back of O, which allows us to consider not just obstacles inflection point, should be considered after the target point in the path at the turn of the problem.Simple line circle structure can not solve this problem,so we have adopted the form of a minimum turning radius at the inflection point and the target point on the way.We can also be appropriate to transform the inflection point of the turning radius, so that the robot can along straight line through the target point of the way, and then create optimization model to optimize these two programs, and ultimately obtained the shortest path.

The model assumes and symbol description are analyzed by the following assumptions:

- (1) Assuming the width of the robot itself is negligible. Thus, the movement of the robot can be regarded as a point moves.
- (2) Assuming the robot walk straight and turn at maximum speed.
- (3) Assuming that the obstacle is always subject to 12 different shapes of the area and the nature of the location, size, etc. has been the same.

Table2 Symbol & Description

Symbol	Description
$v_0$	Maximum speed straight line when walking
$\rho$	Turning radius
$S_i$	The i-th sub-length of the shortest path from $O \rightarrow A$
$L_i$	The i-th sub-length of the shortest path from $O \rightarrow B$
$L_i'$	The i-th sub-length of the shortest path from $O \rightarrow C$
$t_{\min}$	The shortest time from $O \rightarrow A$
$\overline{B_i B_j}$	Arc length from $B_i \rightarrow B_j$
$ B_i B_j $	Length of the line from $B_i \rightarrow B_j$

## 5 FINDINGS AND INTERPRETATIONS

### 5.1 The shortest path from $O \rightarrow A$ optimization model

Two points within the plane of the shortest path based on the length of the line segment as the endpoint, but the connection of these two segments with obstacles intersect, so try to attempt to bypass the obstacle and its hazardous areas other path. Obstacle is a square, the center of this square is located in the lower part of the connection, so the robot to bypass the obstacle from the top of the obstacle path is the shortest path.

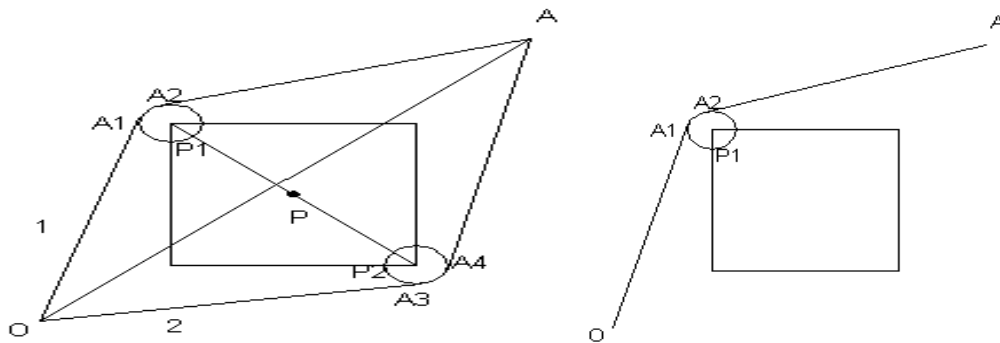


Figure 2.  $O \rightarrow A$  path

Shown in Figure 2, the shortest path from  $O \rightarrow A$  is constituted by straight line  $OA_1$  and  $A_2A$  and a tangent arc  $\overline{A_1A_2}$  wherein the cut point. Which  $A_1A_2$  as the cutoff point. Arc  $\overline{A_1A_2}$  thought the center of

the circle  $P_1(80,210)$ , the radius is 10. Set cut-point coordinates  $A_1(x_1, y_1)$   $A_2(x_2, y_2)$ , these three sections of the path length can be calculated:

$$A_2A = s_3 = \sqrt{(x_2 - 300)^2 + (y_2 - 300)^2} \quad OA_1 = s_1 \quad (1)$$

$$OA_1 = s_1 = \sqrt{(x_1 - 0)^2 + (y_1 - 0)^2} \quad (2)$$

$$\overset{\frown}{A_1}A_2 = s_2 = 2 \arcsin \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{20} \cdot r \quad (3)$$

Here we have  $x_1, y_1, x_2, y_2$  for the decision variables, the total length of the shortest path as the

$$s = \sum_{s_1}^3 s_i$$

objective function: Min

Constraints: radius  $r = 10$ ,  $OA_1^2 + A_1P^2 = OP^2$ ,  $A_2A_1^2 + PA_1^2 = A_2P$ , points  $A_1, A_2$  on the arc. In summary, the structure optimization model as follows:

$$\text{Min } s = \sum_{s_1}^3 s_i = \sqrt{x_1^2 + y_1^2} + \sqrt{(x_1 - 300)^2 + (y_2 - 300)^2} + 20t \quad (4)$$

$$s.t. \begin{cases} x_1 \leq 80 \\ x_1 \geq 70 \\ y_1 \leq 220 \\ y_1 - \sqrt{100 - (x_1 - 80)^2} = 210 \\ x_1^2 + y_1^2 + 100 = 50500 \\ (x_1 - 300)^2 + (y_2 - 300)^2 + 100 = 56500 \\ \sin t = \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{20} \end{cases} \quad (5)$$

Solving the above model (see Appendix 1 lingo procedures), the results are as follows:

1) The  $OA$  length of the shortest path:  $s_{\min} = 471.0372$

$$A_1 = (70.50596, 213.1406)$$

2) Two arc tangent point coordinates:  $A_2 = (76.60640, 219.4066)$

Robot shortest path from point O to point A is reached can be expressed in the table3:

Table3. the shortest path from  $O \rightarrow A$

No	Start	End	Types of segments	Length
1	(0, 0)	(70.50596,213.1406)	Straight line	224.4994
2	(70.50596,213.1406)	(70.6064,219.4066)	(80,210) as the center of the arc	9.1105
3	(70.6064,219.4066)	(300, 300)	Straight line	237.4273
Total length				471.0372

## 5.2 The shortest path from $O \rightarrow B$ model

In this section, we will simplify the roadmap as an empowered network diagram, and use the ant colony algorithm to find the approximate route of the shortest path.

The ant colony algorithm is a bionic algorithm derived from the nature of ants routing mode simulation. Ants in the process of movement will leave a substance called Pheromone on its path through the information transferred. Ants can perceive this substance in the course of the campaign as their movement direction. Therefore, a large number of ants showed an information feedback phenomenon: more ants walking on a path, choose the after the greater the probability.

We put on a plane at some point number, and the relationship of the distance between them simplified network chart. If the node can directly reach in a straight line rather than an obstacle, the weights of the edges between them weight of the straight-line distance, otherwise there is no edge between them. As follows:

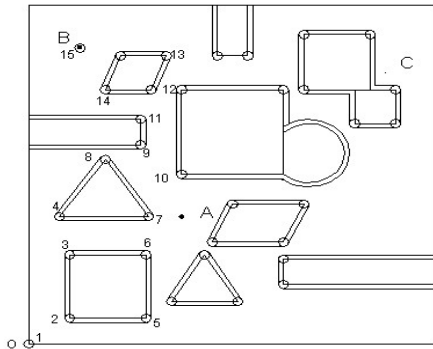


Figure 4. Nodes Numbers Figure

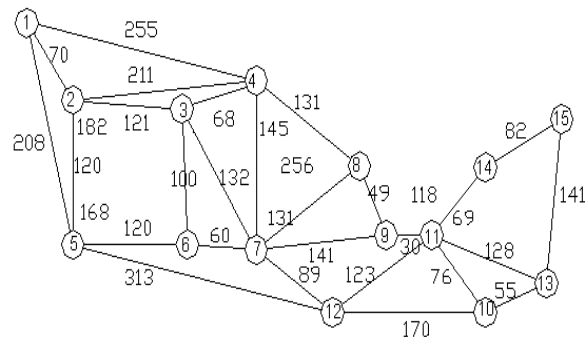


Figure 5. Empowering network chart

Using ant colony algorithm to select the shortest route from  $O \rightarrow B$  from the simplified network chart.

### 1) Ant colony algorithm model

Value the point  $1 \rightarrow 15$ , 0-1 whether if it on the path, form 15 bit sequence 0,1, thereby calculating the distance of this path. The distance as a mapping of the pheromone variable, due to the requirements of the most short-circuit, so you can use the countdown or relative distance as the pheromone concentration. Then you get each ant transition probability. If transition probability is greater than the global transfer factor, then the global transfer; otherwise transfer must have step. So that you can step to the global optimal solution close.

### 2) Perform steps

The first step to initialize N ants. In fact N road, and calculate the current position of ants.

The second step initialization of operating parameters, start the iteration.

The third step in the iterative complement the range of calculated transition probabilities, less than the global transition probability for small-scale search, or a wide range of search.

The fourth step is to update the pheromone, records state, ready for the next iteration.



The fifth step is to enter the third step  
 The Sixth step output and programming.(See as Appendix 1)  
 3) Show the results as following:

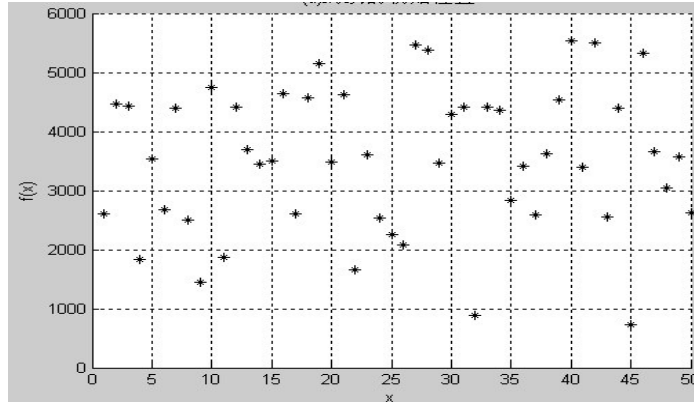


Figure 6. The initial position of the ants

Figure 6 shows where the initial position of the 50 ants in disorderly distribution.

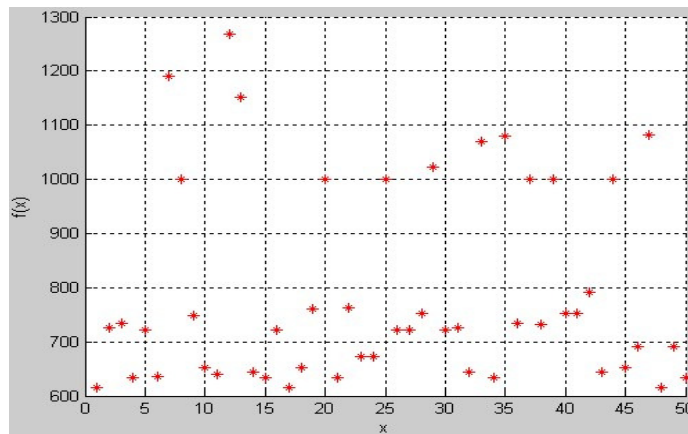


Figure 7Ants' final position

Figure 7 shows that the ants' position after moved, After optimization, ant the two level differentiation, so we get the optimal solution.

### 3) Analysis of results

The initial state of 50 ants are disorderly distribution, optimized the final position to the polarization, so that we get the optimal solution.

Figure6 are average and optimal curve, from which you can know that the algorithm converges very fast, the effect is better. The shortest path is  $1 \rightarrow 4 \rightarrow 8 \rightarrow 9 \rightarrow 11 \rightarrow 14 \rightarrow 15$  . Chromosome: 100100011010011, running time is 0.3910.

Therefore, the shortest path from  $O \rightarrow B$  as shown below:

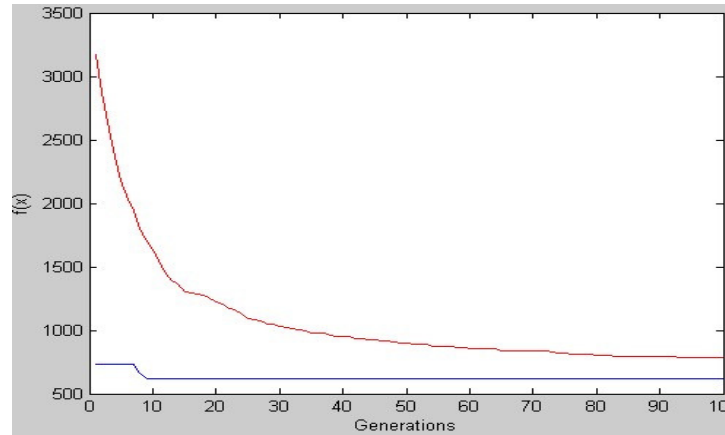


Figure 8. Pheromone concentration average value and the optimum value

### 5.3 The segmented path length from $O \rightarrow B$ optimization model for solving

In the previous section, we have determined to run the route of the shortest node  $1 \rightarrow 4 \rightarrow 8 \rightarrow 9 \rightarrow 11 \rightarrow 14 \rightarrow 15$  in Figure 3, but this simplified diagram from only consider a straight line, without regard to the actual deployment of the arc length. Therefore, we put this route segment, making each piece only route to bypass an obstacle.

Based on the above analysis, from this route  $O \rightarrow B$  is divided into five sections  $L_1(O \rightarrow B_3)$ ,  $L_2(B_3 \rightarrow B_6)$ ,  $L_3(B_6 \rightarrow B_9)$ ,  $L_4(B_9 \rightarrow B_{12})$ ,  $L_5(B_{12} \rightarrow B)$  calculate their length, then the sum thus obtained the shortest path from  $O \rightarrow B$ .

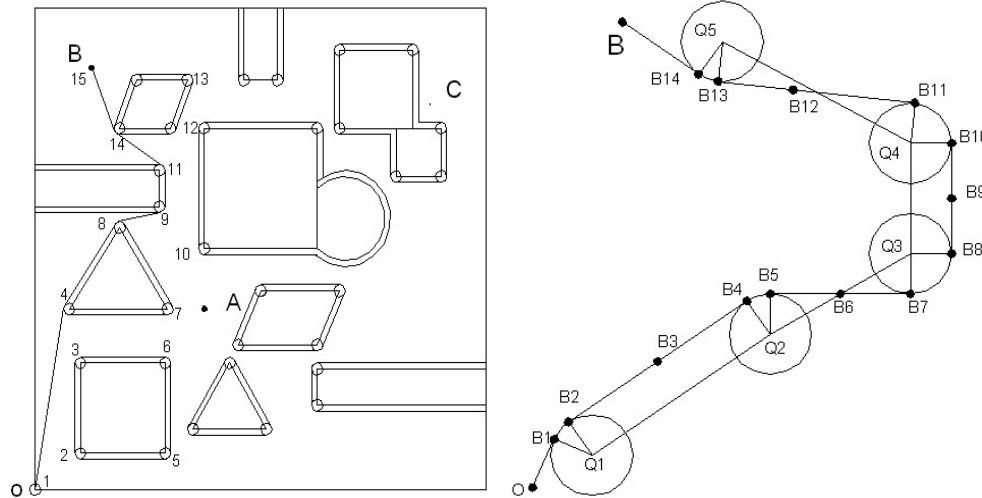


Figure9. Seeking from  $O \rightarrow B_3$  shortest path

**1) Seeking from  $O \rightarrow B_3$  shortest path:**

seek the shortest path  $O \rightarrow A$  structure optimization model is similar to when seeking the shortest path  $O \rightarrow B_3$ , we will coordinate  $O, B_3, Q_1$  as the route start point  $(a,b)$ , end point  $(c,d)$  and the arc center  $(m,n)$  coordinates variable values.

Namely:  $a = 0, b = 0; c = 100, d = 378; m = 60, n = 300$ .

The coordinates of the cut-off point  $B_1(x_1, y_1), B_2(x_2, y_2)$  as a decision variable,  $O \rightarrow B_3$  as the objective function of the length of the shortest structure optimization model as follows:

$$\text{Min } s = \sqrt{(x_1 - m)^2 + (y_1 - n)^2} + \sqrt{(x_1 - c)^2 + (y_2 - d)^2} + 20t \quad (6)$$

$$s.t. \left\{ \begin{array}{l} x_1 \leq m + 10 \\ x_1 \geq m \\ y_1 \leq n + 10 \\ y_1 - \sqrt{100 - (x_1 - m)^2} = n \\ (x_1 - a)^2 + (y_1 - b)^2 + 100 = m^2 + n^2 \\ (x_1 - c)^2 + (y_2 - d)^2 + 100 = (m - c)^2 + (n - d)^2 \\ \sin t = \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{20} \end{array} \right. \quad (7)$$

Solving the above model, the results are as follows:

The length  $OB_3$  of the shortest path:  $s_{\min} = 397.0986$

Two arc tangent point coordinates:

$$B_1 = (50.1353, 301.6396)$$

$$B_2 = (51.6795, 305.547)$$

**2) Seeking  $B_3 \rightarrow B_6$  shortest path model**

The optimization model with the same route as the previous paragraph, the coordinates of  $B_3, B_6, Q_2$  as the beginning  $(a,b)$  of the route, the end  $(c,d)$  and the arc center  $(m,n)$  coordinates of the variable value. Namely:

$$a = 100, b = 378; c = 185, d = 452.5; m = 150, n = 435.$$

Using the lingo program solving the optimization model, the length of the shortest path. Similarly, we can calculate the shortest path of the other sub-routes. In summary, we have come to the shortest path.

Table4. The result of the length of the shortest path

Segm- ented	Start	End	Line Types	Length
1	(0,0)	(50.1353,301.6396)	Straight line	305.7777
2	(50.1353,301.6396)	(51.6795,305.547)	(60, 300) as the center of the arc	5.88
3	(51.6795,305.547)	(141.6795,440.547)	Straight line	162.2498
4	(141.6795,440.547)	(147.9621,444.79.0)	(150, 435) as the center of the arc	7.7756
5	(147.9621,444.79.0)	(222.0379,460.2099)	Straight line	75.6637
6	(222.0379,460.2099)	(230,470)	(220, 470) as the center of the arc	13.6557
7	(230,470)	(230,530)	Straight line	60
8	(230,530)	(225.5026,538.3538)	(220, 530) as the center of the arc	9.8883
9	(225.5026,538.3538)	(144.5033,591.6462)	Straight line	96.9536
10	(144.5033,591.6462)	(140.6892,596.3523)	(150, 600) as the center of the arc	6.1545
11	(140.6892,596.3523)	(100,700)	Straight line	110.377
Total length				854.3759

#### 5.4 The shortest path modle from $O \rightarrow C$

In this section, we also use the same algorithm as solve the path from  $O \rightarrow B$ , first use the ant colony algorithm selects the shortest route, then segmenting the route and calculate each segment of the length, and then added them, at last get the shortest path from  $O \rightarrow C$ .

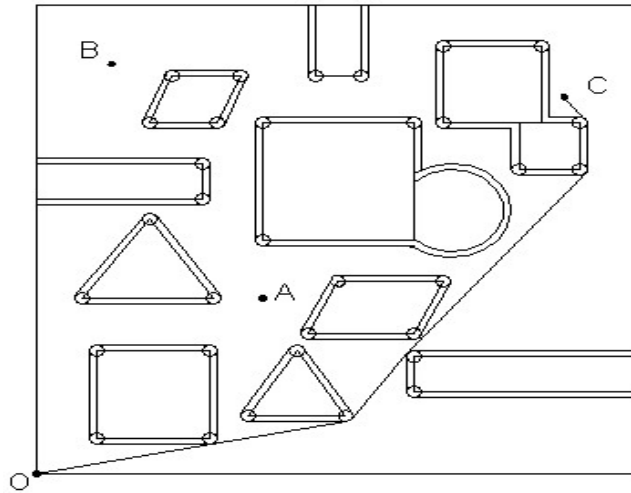


Figure 10 The route map of  $O \rightarrow C$

We found the ideal route in the range of  $O \rightarrow C$  (Figure 3), the robot can walk the route to B, which this route is composed of six segments and five segments of circular arcs. we can take the path abstraction for the following geometric figure (see figure 4) if calculate it directly and the route length can not get a solution, we can divided the route into five routes ( $L_1(OC_3)$ ,  $L_2(C_3C_6)$ ,  $L_3(C_6C_9)$ ,  $L_4(C_9C_{12})$ ,  $L_5(C_{12}C)$ ), and then added them, the shortest path can be obtained.

Table 5 The shortest route from  $O \rightarrow C$

Segment	Start	Final	Line Type	Length
1	(0,0)	(232.1149,50.2262)	Straight line	237.4868
2	(232.1149,50.2262)	(232.1721,50.2381)	(230, 60) as the center of the arc	0.0557
3	(232.1721,50.2381)	(412.1693,90.2381)	Straight line	184.3909
4	(412.1693,90.2381)	(418.3448,94.4897)	(410, 100) as the center of the arc	7.6852
5	(418.3448,94.4897)	(491.6552,205.5103)	Straight line	133.0413
6	(491.6552,205.5103)	(492.0623,206.0822)	(500, 200) as the center of the arc	0.7021
7	(492.0623,206.0822)	(727.9377,513.9178)	Straight line	387.8144
8	(727.9377,513.9178)	(730,520)	(720, 520) as the center of the arc	6.5381
9	(730,520)	(730,600)	Straight line	80
10	(730,600)	(728.0503,605.9324)	(720, 600) as the center of the arc	6.8916
11	(728.0503,605.9324)	(700,640)	Straight line	43.589
Totle length				1088.1951

**5.5 The shortest path heuristic model to solve from  $O \rightarrow A \rightarrow B \rightarrow C \rightarrow O$**

This part try to solve the shortest path from  $O \rightarrow A \rightarrow B \rightarrow C \rightarrow O$ , which is same as solve the shortest path model of  $OB$  and  $OC$ . We use the route map and a route table list all the results.

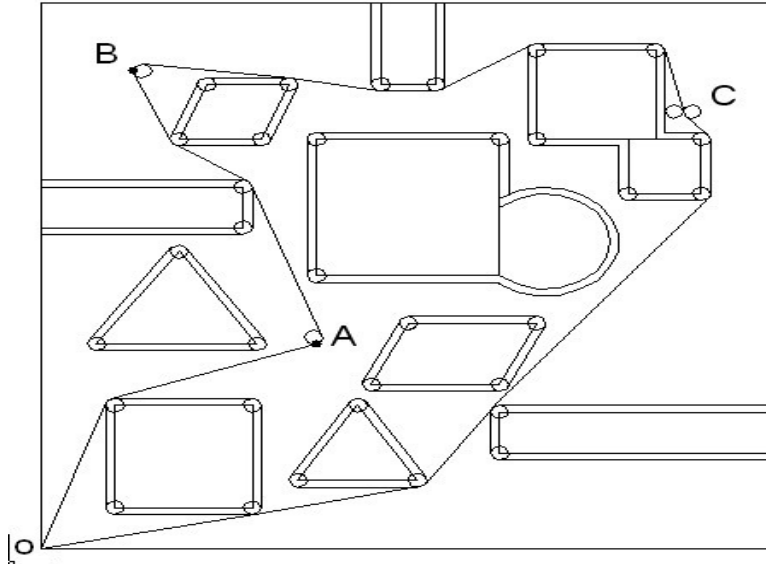


Figure 11. The shortest path map from  $O \rightarrow A \rightarrow B \rightarrow C \rightarrow O$

Table6 The shortest path list for  $O \rightarrow A \rightarrow B \rightarrow C \rightarrow O$

Segment	Start	Final	Line Type	Length
1	(0,0)	(70.50596,213.1406)	Straight line	224.4994
2	(70.50596,213.1406)	(76.6064,219.4066)	(80,210) as the center of the arc	9.1105
3	(76.6064,219.4066)	(300,300)	Straight line	237.4273
4	(300,300)	(306.0528,312.6871)	(296.6062, 309.4065) as the center of the arc	15.5905
5	(306.0528,312.6871)	(229.4525,533.2814)	Straight line	233.5166
6	(229.4525,533.2814)	(225.496,538.3535)	(220,530) as the center of the arc	6.5459
7	(225.496,538.3535)	(144.5027,591.6462)	Straight line	96.9536
8	(144.5027,591.6462)	(140.7174,596.2861)	(150,600) as the center of the arc	6.0832
9	(140.7174,596.2861)	(100.0022,696.2865)	Straight line	107.7033
10	(100.0022,696.2865)	(111.2403,709.9228)	(110.0077,700) as the center of the arc	20.7566

11	(111.2403,709.9228)	(271.2403,689.9228)	Straight line	161.2452
12	(271.2403,689.9228)	(272.0022,689.798)	(270,680) as the center of the arc	0.77
13	(272.0022,689.798)	(368 ,670.202)	Straight line	97.7996
14	(368 ,670.202)	(370, 670)	(370,680) as the center of the arc	2.0136
15	(370, 670)	(430, 670)	Straight line	60
16	(430, 670)	(431.2708,671.7068)	(430,680) as the center of the arc	5.9291
17	(431.2708,671.7068)	(530.0951,738.2932)	Straight line	119.1638
18	(530.0951,738.2932)	(540, 740)	(540,730) as the center of the arc	5.9291
19	(540, 740)	(670, 740)	Straight line	130
20	(670, 740)	(679.7675,732.1438)	(670,730) as the center of the arc	13.5474
21	(679.7675,732.1438)	(699.7689,641.6437)	Straight line	92.1954
22	(699.7689,641.6437)	(700, 640)	(690,640) as the center of the arc	2.1867
23	(700, 640)	(702.6928,633.1732)	(710,640) as the center of the arc	7.5142
24	(702.6928,633.1732)	(727.3094,606.8268)	Straight line	36.0555
25	(727.3094,606.8268)	(730, 600)	(80, 210) as the center of the arc	7.5142
26	(730,600)	(730,520)	Straight line	80
27	(730,520)	(727.9377,513.9178)	(720, 600) as the center of the arc	6.5381
28	(727.9377,513.9178)	(492.0623,206.0822)	Straight line	387.8144
29	(492.0623,206.0822)	(491.6552,205.5103)	(500, 200) as the center of the arc	0.7021
30	(491.6552,205.5103)	(418.3448,94.4897)	Straight line	133.0413
31	(418.3448,94.4897)	(412.1693,90.2381)	(410, 100) as the center of the arc	7.6852
32	(412.1693,90.2381)	(232.1721,50.2381)	Straight line	184.3909
33	(232.1721,50.2381)	(232.1149,50.2262)	(230, 60) as the center of the arc	0.0557
34	(232.1149,50.2262)	(0,0)	Straight line	237.4868
Total length				2737.765

## 6. CONCLUSIONS

With the continual development of robotic research in the field of artificial intelligence, the use of ant colony algorithm effectively solves the problem of robot path planning in the practical work of calculation. Our studies show that in a certain range, the optimized model for ant colony algorithm can be used to calculate and design the shortest path when a robot moves from a starting point beyond some obstacles and reaches the specified target points opposite the obstacles without any collision.

There existed some advantages to use above model to solve the robot avoid obstacles, we can get the best solution in the relative optimization by using of path optimization of multiple solutions. Also we knew there would be got the solution in higher accuracy with the optimized model was solved by analytic geometry. What's More, the model is simple and easy to understand, easy to practice and application. Nevertheless, further study is necessary in that some limitations still exist in mobile robot path planning via ant colony algorithm, e.g. the model for the shortest path planning remains to be optimized, and whether there exist other algorithm solutions for mobile robot path planning etc.

## ACKNOWLEDGMENT

We would like to thank the referees very much for their valuable comments and suggestions, there also got some help from Miss Lu Yan and Ms. Xu Si.

## REFERENCES

- [1] TAN Min, WANG Shuo.(2013) Research Progress on Robotics. ACTA AUTOMATICA SINICA, 39(7):pp.963-972.
- [2] Yahja A., Singh S., Stentz A.(2000) An Efficient Online Path Planner For Outdoor Mobile Robots. Robotics And Autonomous Systems, 32(2):pp.129-143.
- [3] LU Qing.(2007) Research of Path Planning for Car-Like Robot Based on Grid Method. Computer and Information Technology, 15(6):pp.24-27.
- [4] Oommen B., Iyengar S., Rao N.,Kashyap R.(1987) Robot Navigation In Unknown Terrains Using Learned Visibility Graphs. IEEE Journal of Robotics and Automation, 3(6):pp.672-681.
- [5] LI Shan-shou, FANG Qian-sheng, XIAO Ben-xian,QI Dong-liu.(2008) Environment Modeling in Global Path Planning Based on Modified Visibility Graph. Journal of East China Jiaotong University, 25(6):pp.73-77.
- [6] CAI Xiao-hui.(2007)The Path Planning of Mobile Robots Based on Intelligent Algorithms. Zhejiang University Master Degree Thesis, 5:pp.6-12.
- [7] Takahashi O. and R. Scilling.(1989) Motion Planning in a Plane using Generalised Voronoi Diagrams. IEEE Transactions on Robotics and Automation, 5,(2):pp.169-174.
- [8] JIN Lei-ze, DU Zhen-jun, JIA Kai.(2007) Simulation study on mobile robot path planning based on potential field. Computer Engineering and Applications, 43(24):pp.226- 229.
- [9] Khatib O.(1985) Real Time Obstacle Avoidance For Manipulators And Mobile Robots[J].The International Journal of Robotics Research, 5(2):pp.500-505.
- [10] Koren Y, And Borenstein J.(1991) Potential field methods and their inherent limitations for mobile robot navigation[C]. Proceedings of the IEEE Conference on Robotics and Automation, Sacramento, California, , April 7-12: pp.1398- 1404.
- [11] Al-Taharwa I.,Sheta A.,Al-Weshan M.(2008) A Mobile Robot Path Planning Using Genetic Algorithm In Static Environment. Journal of Computer Sciences, 4(4):pp.341-344.
- [12] YANG Lin-quan, LUO Zhong-wen, TANG Zhong-hua, LV Wei-xian. (2008) Path Planning Algorithm For Mobile Robot Obstacle Avoidance Adopting Bezier Curve Base on Genetic Algorithm. 2008 Chinese Control and Decision Conference:pp.3286-3289.
- [13] Gony Lally, Ann M. Lally and Hsin Chun Chen.(2003) The use of dynamic contexts to improve casual Internet searching. ACM Transactions on Information System, 21(3):pp.229-253.
- [14] St.Preitl, R.E.Precup, J.Fodor, B.Bede.(2006) Feedback Tuning In Fuzzy Control System.Theory and Applications, 3(3):pp.81-96.
- [15] P.Vadakkepat, O.C.Miin, X.Peng, T.H.Lee.(2004) Fuzzy Behavior-based Control of Mobile Robots[J]. IEEE Transactions on Fuzzy Systems, 12(4):pp.559-564.
- [16] CHEN Huahua, DU Xin, GU Weikang.(2004) Neural Network and Avoidance Genetic Algorithm Based Dynamic Obstacle and Path Planning for A Robot. Journal of Transluction Technology, (4):pp.551-555.



- [17] NI Bin, CHEN Xiong, LU Gongyu.(2006) A Neural Networks Algorithm for Robot Path Planning in Unknown Environment. Computer Engineering and Applications, (11):pp.73-76+109.
- [18] XU Xin-ying, XIE Jun, XIE Ke-ming.(2008) Path Planning of Mobile Robot Based on Artificial Immune Potential Field Algorithm.Journal of Beijing University of Technology, 34(10):pp.1116-1120.
- [19] CHEN Xi, TAN Guan-zheng, JIANG Bin.(2008) Real-time optimal path planning for mobile robots based on immune genetic algorithm. Journal of Central South University (Science and Technology), 39(3):pp.577583
- [20] DING Wei.(2007) Path Planning Based On Immune Evolution And Chaotic Mutation For Mobile Robot. Master Degree in Engineering Dissertation, Harbin University of Science and Technology, March:pp.19-24.
- [21] CUI Shi-gang, GONG Jin-feng, PENG Shang-xian, WANG Jun-song.(2004) Hybrid intelligent algorithm based 3-D of robot path planning. Manufacturing Automation, (2):pp.49-51.
- [22] GUO Yu, LI Shi-yong.(2009) Path Planning for Robot Based on Improved Ant Colony Algorithm. Computer Measurement and Control, 17(1):pp.187-190.
- [23] ZHU Qing-bao.(2005) Ant Colony Optimization Parallel Algorithm And Based On Coarse-grain Model.Computer Engineering, 31(1):pp.157-159.
- [24] FAN Lu-qiao YAO Xi-fan BIAN Qing-qing JIANG Liang-zhong.(2008) Ant Colony Algorithm and the Application on Path Planning For Mobile Robot. Robotics Technology, 23:pp.257-259+261.
- [25] XIE Min,GAO Li-xin.(2008) Ant algorithm applied in optimal path planning.Computer Engineering and Applications. Computer Engineering and Applications, 44(8):pp.245-248.
- [26] ZHU Qing-bao.(2005) Ants Predictive Algorithm For Path Planning Of Robot In A Complex Dynamic Environment. Chinese Journal of Computers, 28(11):pp.1898-1906.

## Appendix1: Matlab programming of Ant Colony Algorithm

### function shortroad\_ant\_main

```

% Ant main program
clear all;close all;clc;%clear all
tic;%time start
Ant=50;Ger=100;% Running parameter initialization

power=[0      70      1000      276      208      1000      1000      1000      1000      1000      1000
        1000      1000      1000      1000
70      0 141      211      120      182      1000      1000      1000      1000      1000      1000
        1000      1000
1000 141      0      68      168      100      132      1000      500      1000      1000      1000
        1000      1000      1000
276 211      68      0      1000      1000      145      131      1000      1000      1000      1000
        1000      1000      1000
208 120      168      1000      0      120      1000      1000      1000      1000      1000      313
        1000      1000
1000 182      100      1000      120      0      60      1000      1000      1000      1000      1000
        1000      1000      1000
1000 1000      132      145      1000      60      0      131      141      1000      1000      89
        1000      1000
1000 1000      500      131      1000      1000      131      0      49      1000      1000      1000
        1000      555      1000
1000 1000      1000      1000      1000      1000      141      49      0      1000      30      1000
        1000      118      1000
1000 1000      1000      1000      1000      1000      1000      1000      1000      0      76      170
        55      1000      1000
1000 1000      1000      1000      1000      1000      1000      1000      30      76      0      123
        128      69      1000
1000 1000      1000      1000      313      1000      89      1000      1000      170      123      0
        1000      1000
1000 1000      1000      1000      1000      1000      1000      1000      1000      55      128      1000
        0      1000      141

```

1000	1000	1000	1000	1000	1000	1000	555	118	1000	69	1000
	1000	0	82								
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
	141	82	0];								

```

[PM PN]=size(power);
% Initialization Ant place
v=init_population(Ant,PN);
v(:,1)=1;v(:,PN)=1;% The beginning and end points in the path
% The distance when the information factors concentration
fit=short_road_fun(v,power);
% Distance as small as possible, so and information factors concentration corresponding
T0 = max(fit)-fit;
% Draw the picture
figure(1);grid on;hold on;
plot(fit,'k*');
title('(a) The initial position of ants ');
xlabel('x');ylabel('f(x)');
% Initialization
vmfit=[];vx=[];
P0=0.2; % P0---- Global transfer factor
P=0.8; % P ---- Pheromone evaporation coefficient
%C=[];
% Start search the shortest path
for i_ger=1:Ger
lamda=1/i_ger; % Transfer of step size parameters
[T_Best(i_ger),BestIndex]=max(T0);% The most information pheromone concentration
for j_g=1:Ant % To obtain the global transition probability
r=T0(BestIndex)-T0(j_g);% The best distance with ant
Prob(i_ger,j_g)=r/T0(BestIndex);% How much speed rate should be closer to it
end
for j_g_tr=1:Ant
if Prob(i_ger,j_g_tr)<P0
% Local transfer ---- Small step transfer
M=rand(1,PN)<lamda;
temp=v(j_g_tr,:)-2.*(v(j_g_tr,:).*M)+M;
else
%Globe transfer---- Big step transfer
M=rand(1,PN)<P0;
temp=v(j_g_tr,:)-2.*(v(j_g_tr,:).*M)+M;
end
% Reput the beginning and end points,which is not in the moving process of change
temp(:,1)=1;temp(:,end)=1;
if short_road_fun(temp,power)<short_road_fun(v(j_g_tr,:),power)
%Recorder
v(j_g_tr,:)=temp;
end
end
% Update the information factors, To prepare for the next iteration
fit=short_road_fun(v,power);
T0 = (1-P)*T0+(max(fit)-fit);% information factors evaporation
[sol,indb]=min(fit);

```



```

x(1)^2+y(1)^2+100=80^2+210^2;
(x(2)-300)^2+(y(2)-300)^2+100=(80-300)^2+(210-300)^2;

@sin(t)=(((x(1)-x(2))^2+(y(1)-y(2))^2)^(1/2))/20;
@for(qiedian(I):x(I)<=80);
@for(qiedian(I):x(I)>=70);
@for(qiedian(I):y(I)<=220);
@for(qiedian(I):y(I)-(100-(x(I)-80)^2)^(1/2)=210);

```

**END**

### Appendix3: Using Lingo Programming To Solve The Shortest Route For OB

```

model:
sets:
qiedian/1..2/:x,y;
endsets

DATA:
a=0;
b=0;
c=100;
d=378;
m=60;
n=300;
ENDDATA

min=((x(1)-a)^2+(y(1)-b)^2)^(1/2)+((x(2)-c)^2+(y(2)-d)^2)^(1/2)+20*t;
(x(1)-a)^2+(y(1)-b)^2+100=m^2+n^2;
(x(2)-c)^2+(y(2)-d)^2+100=(m-c)^2+(n-d)^2;
@sin(t)=(((x(1)-x(2))^2+(y(1)-y(2))^2)^(1/2))/20;
@for(qiedian(I):y(I)<=n+10);
@for(qiedian(I):(x(I)-m)^2+(y(I)-n)^2=100);

```

**END**

### Appendix4: Using Lingo Programming To Solve The Shortest Route And Time For OA

```

model:
sets:
qiedian/1..2/:x,y;
endsets

DATA:
a=0;
b=0;
c=300;
d=300;
m=80;
n=210;
ENDDATA

min=(((x(1)-a)^2+(y(1)-b)^2)^(1/2))/5+(((x(2)-c)^2+(y(2)-d)^2)^(1/2))/5+2*r*t/(v);

@exp(10-0.1*r^2)=5/v-1;
@sin(t)=(((x(1)-x(2))^2+(y(1)-y(2))^2)^(1/2))/(2*r);

r>=10;
(x(1)-a)^2+(y(1)-b)^2+r^2=m^2+n^2;

```

$$(x(2)-c)^2+(y(2)-d)^2+r^2=(m-c)^2+(n-d)^2;$$

@for(qiedian(I):x(I)<=m+r);  
@for(qiedian(I):x(I)>=m-r);  
@for(qiedian(I):y(I)<=n+r);  
@for(qiedian(I):(x(I)-m)^2+(y(I)-n)^2=r^2);

**END**

## **Authors**

### **(1) GUO Yue**

PhD in Management Science and Engineering, MAIB, BBA  
Profersor, Dean of Department of Enterprising Management  
Ningbo University of Technology



In 2010 Dr. Guo award Chinese Provincial New Century Hundred-Thousand-Ten thousand Engineering Talent, in 2011 he also got Chinese Provincial Backbone Middle-Young Aged Teachers. Dr. Guo has been published over 50 papers and 8 books in his academic field. In management science and engineering field he also do some consulting services to solve the actual problems for local companies and enterprises.

### **(2) SHEN Xuelian**

PhD graduated from Southwest University of Finance and Economics, as a lecturer teaching in economics field courses at Ningbo University of Technology. He got his master degree from Swiss University.



### **(3) ZHU Zhanfeng**

Postdoctoral fellow, PhD in Management Science and Engineering, MBA, BA Profersor,  
Headof Department of Economics & Management Ningbo University of Technology



He was visiting scholar of University of magdeburg in Germany; and got advanced logistics management division, registered senior consultant and Doctoral tutor. Also presently for ningbo college of engineering management, and director of the institute of the ningbo college of engineering development of small and medium-sized enterprises. At the same time, he was also the ministry of education of institutions of higher learning MBA class teaching steering committee of logistics and electronic commerce, deputy director of the national demonstration on repository construction project logistics management specialty teaching work committee.