2009 Fifth International Conference on Information Assurance and Security

Swarm Intelligence: Ant-based Robot Path Planning

Jing Zhou, Guan-Zhong Dai, De-Quan He, Jun Ma, Xiao-Yan Cai
School of Automation
Northwestern Polytechnical University
Xi'an, P.R.China
zhoujingnwpu@gmail.com

Abstract — **In this paper we proposed a novel algorithm Ant-based Robot Path Planning (ARPP) based on Ant Colony System (ACS) to mimic a swarm of ants to find the globally optimal path for autonomous mobile robots. Visibility graph was used as both the roadmap and construction graph in ARPP. Although the near-optimums were readily available by ARPP, it is hard to find the optimality; therefore we proposed the special ARPP (S-ARPP) algorithm as its supplement. Experimental results show S-ARPP outperforms ARPP for higher qualities of the global-best path it found, and a flexible trade-off between satisfactory solutions and the number of iterations was also easily available to meet the certain needs.**

Keywords-ant colony system; visibility graph; ant-based robot path planning; S-ARPP

1. INTRODUCTION

Even though there are several research efforts on mobile robot Global Path Planning (GPP) like maklink graph methods[1], voronoi diagrams[2], artificial potential fields[3], fast marching methods[4][5], grid-based method[6] *etc*., these methods are relatively traditional; Ant Colony Optimization (ACO)[10] as a new computational paradigm for solving distributed problems on self-organization is one of the prevailing algorithms in Swarm Intelligence, and therefore ACO-based approaches for GPP had been put forward by scholars from all over the world. But the previous works, such as [7] [8] [9] *etc*, were mainly based on Ant System (AS) [10], an original prototype for a bunch of ACO algorithms. Moreover [11] used Ant-Q framework which extends ACO with reinforcement learning with grid; [12] [13] applied Ant Colony System (ACS) [14] with maklink graph. However, the research using ACS with visibility graph (VG) for GPP is rare. Hence we propose ACS-based robot path planning algorithm (ARPP) with VG for GPP in this paper; moreover a further modified algorithm based on ARPP, named Special-ARPP (S-ARPP), is also proposed.

2. PROBLEM REPRESENTATION

In this section, the assumptions for GPP problem were given. When ACO is applied to an optimization problem, usually the construction graph will be involved.

The following assumptions need to be made: 1) we assume the moving environment of the robot is a two-dimensional space; 2) both the environment and obstacles have a polygonal shape; 3) the boundaries of every obstacle will be expanded in order to avoid moving path too close to the obstacles, so that we can ignore the size of the robot[12].

Fig. 1 illustrates the moving environment of robot $\Re$, which is a 100 by 100 meters square including six obstacles. Obstacle set is denoted by S, $S \overset{def}{=} \{S_1, S_2, \ldots, S_6\}$. Points p_{start} and p_{end} denote the starting point and goal respectively.

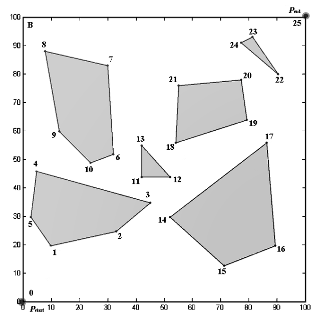

Figure 1. Moving Environment of $\Re$

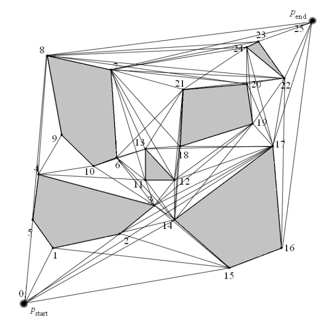

Figure 2. Visibility Graph $\zeta_{\text{vis}}(S^*)$

Fig.1. Symbols 0 to 25 denote respectively the vertices of expanded obstacles. The (x, y) coordinates of vertices 0 to 25 are (0, 0), (10, 20), (33, 25), (45, 35), (5,46), (3, 30), (32, 52), (30, 83), (9, 88), (13, 60),(24, 49), (42, 44), (52, 44), (42, 55), (52, 30), (71,13), (89,20), (86,56), (54,56), (79,64), (77,78), (55,76), (90,80), (81,93), (77,91), and (100,100) respectively. The (x, y) coordinates of p_{start} and p_{end} are (0, 0) and (100, 100) respectively.

In this paper, we use the visibility graph S^* as both the roadmap and construction graph. To simplify the GPP problem, points p_{start} and p_{end} are added to S, denoted S^*, $S^* \overset{def}{=} S \cup \{p_{\text{start}}, p_{\text{end}}\}$; we construct visibility graph of S^*, denoted $\zeta_{\text{vis}}(S^*)$, $\zeta_{\text{vis}}(S^*) \overset{def}{=} \{V, E\}$.

Fig.2 illustrates visibility graph $\zeta_{\text{vis}}(S^*)$, construction method of visibility graph can refer to [1].

3. ARPP ALGORITHM FOR GPP

3.1. Modified ACS for Global Path Planning

ACS is often one of the best-performing algorithms of ACO [15] and has given good results for Traveling Salesman Problem (TSP) and some other problems. For GPP problem, there are three aspects are modified in ACS: *Initialization*, *Active State Transition Rule*, and *Prior Candidate List*.

Initialization: In ARPP, all ants were placed in p_{start} to

978-0-7695-3744-3/09 $25.00 © 2009 IEEE
DOI 10.1109/IAS.2009.120

construct solutions until reaching p_{end}. The initial value of pheromone is defined as $\tau_0 \stackrel{def}{=} 1/nC_{nn}$, where C_{nn} is constructed by a heuristic method — Nearest-Neighbor (NN)[14]. n denotes the number of vertexes in $\zeta_{\text{vis}}(S^*)$; the value of heuristic information is defined as $\eta_{il} \stackrel{def}{=} 1/d_{il}$, where d_{il} denotes the distance between node i and j, two neighbors in graph $\zeta_{\text{vis}}(S^*)$.

Active State Transition Rule: we suggest all ants should construct paths with an active attitude; otherwise, ants will be lost in obstructions, then p_{end} can seldom be arrived by which. This means that if ant k is in node i, p_{end} is i 's next step neighbor, no matter how far from i to p_{end}, k should go directly to p_{end} to finish its construction. We use *Attitude Equation* as the first branch of simultaneous equation of *Active State Transition Rule*. Supposing Ant k is in node i choosing node j as its next step by applying *Active State Transition Rule*, it is defined by equation (1):

$$j = \begin{cases} p_{\text{end}} & \text{if} \quad p_{\text{end}} \in N_i^k; \\ \underset{l \in N_i^k}{\operatorname{argmax}} \{ \tau_{il}{}^{\alpha} \eta_{il}{}^{\beta} \} & \text{else if} \quad q \le q_0 \text{ (exploitation);} \\ S & \text{else} \quad \text{(exploration);} \end{cases} \tag{1}$$

Where N_i^k is the feasible neighborhood of ant k, it is a set of nodes ant k hasn't visited yet; q is a random number, q~U(0,1); q_0 is a parameter, $0 \le q_0 \le 1$; and S is a node chosen according to[14] (biased exploration).

Where α, β are two parameters determining the relative importance of the pheromone and heuristic information. The parameter q_0 determines the relative importance of exploitation and exploration[14].By changing q_0 the degree of exploration of ants can be adjusted.

Prior Candidate List: The *Prior Candidate List* in GPP is different then candidate list of TSP. By always placing p_{end} at the first position of *Prior Candidate List*, the highest priority of it in candidate list can be guaranteed.

3.2. Updating Rules of Pheromone Trail

Global Updating Rule: ARPP has a global pheromone trail updating approach which exploits the best solutions found. The deposit of pheromone trail, as a kind of positive feedback, aims to reinforce an ant's ability of finding the excellent solutions. *Global Updating Rule* is defined by (2):

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho \Delta \tau_{ij}{}^{\text{bs}}, \qquad \forall l_{ij} \in \mathrm{T}^{\text{bs}} \tag{2}$$

Where $\Delta \tau_{ij}{}^{\text{bs}} \stackrel{def}{=} 1/C^{\text{bs}}$, will be added to the arcs l_{ij} which belongs to the best-so-far solution[14]. ρ ($0 < \rho < 1$) is the global evaporation rate, for avoiding the excessive increase of pheromone.

Local Updating Rule: ARPP includes a local pheromone update to reduce emphasis on exploitation of existing solutions, which is applied on only the arcs ants have been visited. *Local Updating Rule* is defined by (3):

$$\tau_{ij} \leftarrow (1-\xi)\tau_{ij} + \xi \tau_0 \tag{3}$$

Where ξ is the local evaporation rate and τ_0 is the initial amount of pheromone. *Local Updating Rule*, as a kind of negative feedback, aims to encourage the exploration of other regions out of the area around the best-so-far path

By evaporating pheromone trails from visited arcs, the exploration of alternative paths is increased. It is the common efforts of the positive and negative feedback that make the whole process of path optimization evolve.

3.3. Performance Analysis of ARPP

To test the performance of ARPP, an artificial ant colony system Ω including m ants are made. Simulations executed up to 100 trials, initial parameters are shown as Table 1:

TABLE 1. INITIAL PARAMETERS IN ARPP

Parameter	α	β	ξ	ρ	q_0	m
Parameter Value	0.15	2.0	0.15	0.2	0.8	6

The global-best path found by ARPP, with the highest probability up to 74%, on average acquired approximately after 48.77 iterations, is the path with length of 148; The other global-best path, with the lowest probability only 1%, acquired after 825 iterations on average, is the path with length of 141. Statistic results are shown in Table 2.

Variation Coefficient, *Best-so-far Solution (Best-so-far, for short)* and *Iteration-best Solution (Iteration-best, for short)* are used as factors to evaluate the performance of ARPP.

Variation Coefficient: Variation coefficient δ is used to describe the exploration of ARPP algorithm. δ is defined as the quotient between the standard deviation σ_{L} of the path lengths and the average path length $\overline{L}$ [15].

If δ is zero, we sure all ants in Ω construct the same solutions, that is an indication to check whether there is a stagnation or not. Fig.3 illustrates the average tour length constructed by Ω after 10000 iterations; and Fig.4 illustrates the variation coefficient in 5000 iterations.

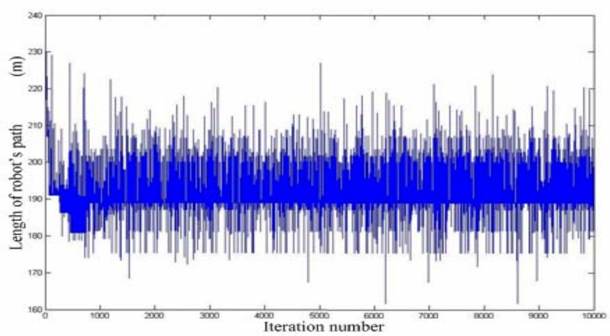

Figure 3. Average path length of all ants

Fig.3 illustrates that at the beginning of iterations there is a ladder-like decline, and after 1000 iterations, the average path length tends to stabilization.

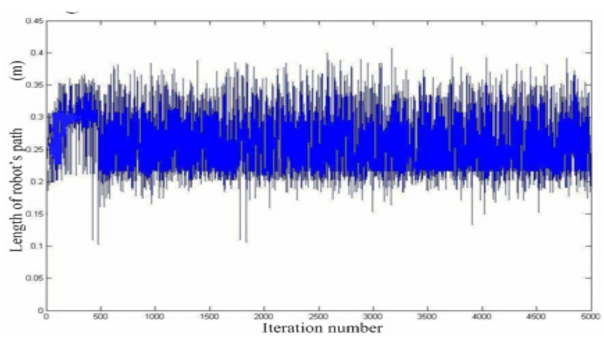

Figure 4. Variation Coefficient

Fig.4 illustrates there is a small δ throughout the 5000 run of algorithm, the value of which doesn't tend to 0 show no stagnation happened.

***Best-so-far* Solution**: For more comprehensively, three kinds of convergence situation of *Best-so-far*, the best, and the worst situation as well as five ordinary situations in 100 trials are used to illustrate the behavior of ARPP algorithm.

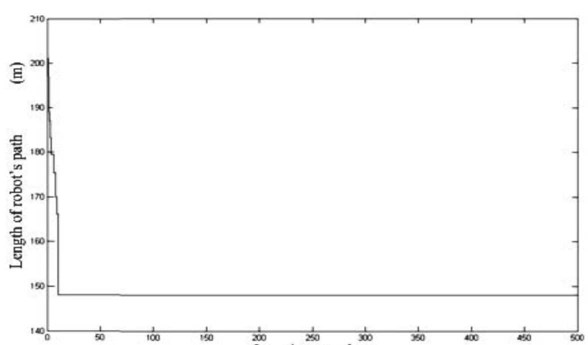

Figure 5. The best convergence of best-so-far

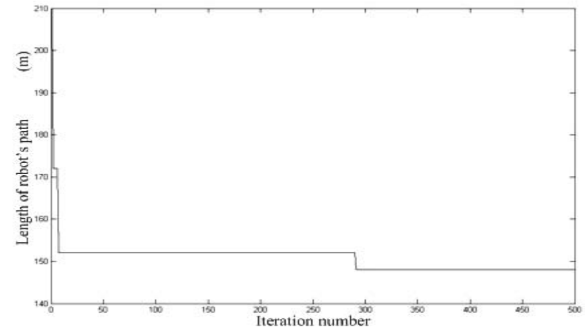

Figure 6. The worst convergence of best-so-far

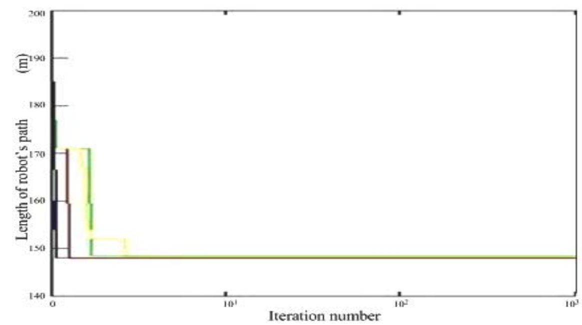

Figure 7. The five ordinary convergences of best-so-far solutions respectively

Fig.5 and Fig.6 show the best and worst convergence situation of *Best-so-far* respectively, and Fig.7 illustrates five ordinary situations. From Fig.5 to Fig.7 we could know nearly optimal solutions are readily available by ARPP. That is to say as best-so-far can easily and quickly converge after a few iterations, ARPP demonstrates a good performance in real-time optimization.

***Iteration-best* Solution**: Fig. 8, 9, and 10 illustrate the best, the worst, and ordinary convergence situation of *Iteration-best*, which are correspondences to Fig.5, 6, and 7. From these Figures we can conclude that new iteration-best solutions seldom can be found after ARPP converges to the nearly optimal solutions, which means that the latter iterations might be basically wasted. Because ARPP uses a direct and aggressive search way, searches of ants concentrate on the area around the nearly optimal paths at the very beginning by employing a large value for q_0.

In conclusion, ARPP has the following characters:1) ARPP has a real-time optimization property—can converge to near-optimum very fast; 2) after ARPP obtained the nearly optimal solution, it is hard to find better solutions afterwards. Although convergence is readily available by ARPP, unsatisfactorily the near-optimum it found is always with high deviation from the optimal.

TABLE 2. STATISTIC RESULTS OF PERFORMANCE OF ARPP

Path	0→17→25	0→2→17→25	0→14→17→25	0→2→14→17→25	0→2→3→12→18→19→22 →25	0→2→14→19→22 →25	0→2→3→19→22→25
Path Value	148	148.1	148.2	148.23	146	144	141
Average Iterations	48.77	12.25	24.37	34.31	15	81	825
Percent	74	9	7	6	2	1	1

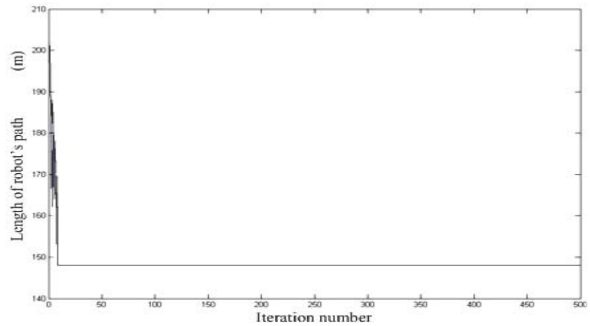

Figure 8. The best convergence of iteration-best corresponds to the Fig.5.

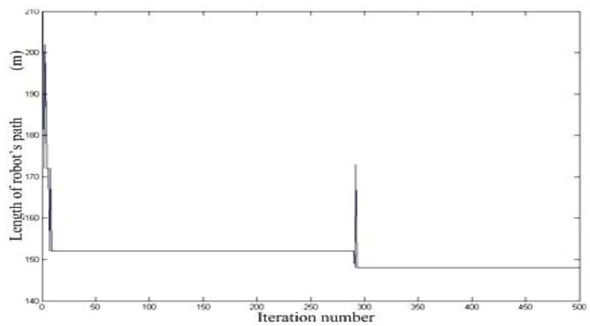

Figure 9. The worst convergence of iteration-best corresponds to the Fig.6.

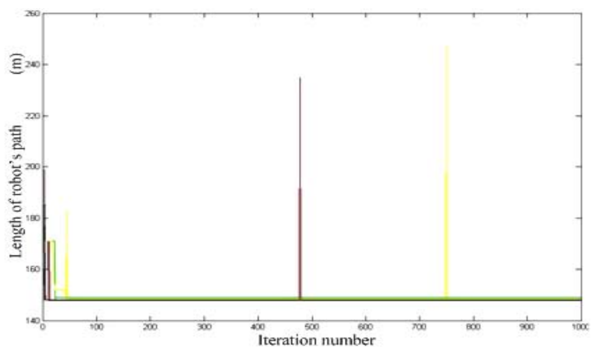

Figure 10. The five ordinary convergences of iteration-best correspond to the Fig.7.

4. S-ARPP ALGORITHM FOR GPP

In this section, we propose S-ARPP algorithm, it has two differences from ARPP:1) parameter q_0 is set to a small value; 2) S-ARPP will reinitialize pheromone trails on the best-so-far paths when no better best-so-far is found after a given number of iterations.

4.1. S-ARPP

S-ARPP uses the same rules of *Active State Transition Rule*, *Global Updating Rule*, and *Local Updating Rule* as ARPP does. However, S-ARPP employs a small q_0 in *Active State Transition Rule*, this is diametrically opposite to ARPP.

Through experiments we know pheromone trails on some arcs must be limited, otherwise the evolution of all paths would be delayed. For instance, the path with length of 152 is shown as Fig.6. Due to accumulating too many pheromones, it attracts ants to its surrounding, which narrows the search space of ants dramatically. As a result it takes too long time to converge to the near-optimum 148. Therefore, we added pheromone reinitialization rule to S-ARPP to eliminate the excessive pheromone trails on some certain arcs.

4.2. Pheromone Reinitialization Rule

S-ARPP implements *Pheromone Reinitialization Rule* by resetting pheromone to a fixed value $n\tau_0$ on the arcs which belong to best-so-far path T^{bs} when no better T^{bs} is found after a given number of iterations. The *Pheromone Reinitialization Rule* is defined by (4):

$$\tau_{ij} \leftarrow n\tau_0, \quad \forall l_{ij} \in \mathrm{T}^{bs} \tag{4}$$

Where n denotes the number of vertexes in $\zeta_{vis}(\mathrm{S}^*)$.

5. PERFORMANCE EVALUATIONS OF S-ARPP

Simulations executed up to 100 trials, the parameters $\{\alpha, \beta, \xi, \rho, m\}$ are the same as ARPP shown in Fig.1. We set q_0 =0.1 in this experiments. The statistic results of simulation are given by Table 3.

Experimental results illustrate the global-best path found by S-ARPP, with the highest probability up to 67%, gained approximate after 172.31 iterations, is the path with length of 141; The other global-best with the lowest probability only 1%, gained after 21 iterations, is the path of 148.23.

Fig.11 and 12 show the satisfactory convergence of best-so-far while Fig.13 demonstrates the dissatisfied convergence of it; Fig.14 shows the convergence of iteration-best. We can see obvious difference between Fig.14 and Fig.8. After S-ARPP obtained the global-best, new iteration-best solutions emerge consistently, since there are more active searches in S-ARPP so that the satisfactory solutions can be guaranteed. From the simulation it can be seen a trade-off between satisfactory solutions and the number of iterations was also easily available.

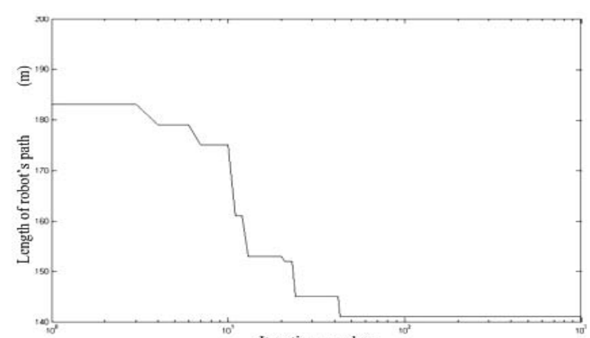

Figure 11. The convergence of best-so-far solutions for the optimal 141.

TABLE 3. STATISTIC RESULTS OF PERFORMANCE OF S-ARPP

Path	0→17 →25	0→2→17→25	0→14→17→25	0→2→14 →17→25	0→2→14→19→22 →25	0→2→3→19→22 →25	0→1→2→3→19→22→25	0→2→3→12→18 →19→22→25
Path Value	148	148.1	148.2	148.23	144	141	145	146
Average Iterations	76.43	33.5	17.66	21	91.64	172.31	85.3	123
Percent	8	4	3	1	12	67	3	2

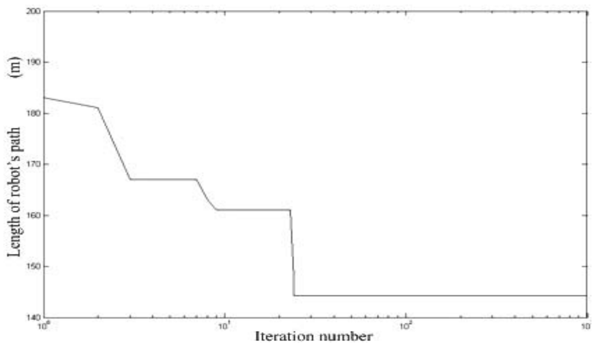

Figure 12. The convergence of best-so-far solutions for the optimal 144

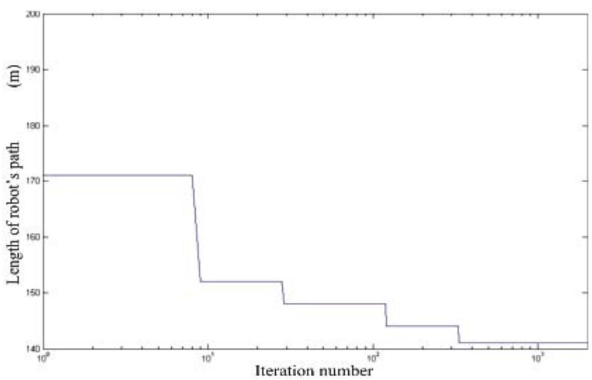

Figure 13. The bad convergence of best-so-far to the optimum 141.

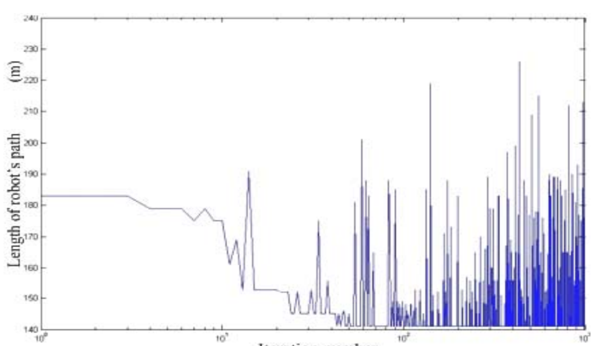

Figure 14. The convergence of iteration-best

6. CONCLUSIONS

In this paper, we proposed ARPP and S-ARPP algorithm for GPP respectively, they are both based on ACS framework, and apply visibility graph as the roadmap and construction graph. ARPP has a merit of real-time optimization, but it often either can't find the optimal or can but with the unacceptable number of iterations. For this in S-ARPP, we suggested giving q_0 a small value to expand the exploration of ants as well as limiting the pheromone on the best-so-far paths which reoccur uninterruptedly. Experimental results illustrate that S-ARPP is effective in small scale, and a flexible balance between the optimum and the number of iterations is available for certain needs.

ACKNOWLEDGMENT

The work is financially supported by the National "863" Project of China (Grant no. 2005AA147030).

REFERENCE

[1] M De Berg, O Cheong,"Computational Geometry Algorithms and Applications (Second Edition)", Springer

[2] P. Bhattacharya, ML. Gavrilova "Voronoi diagram in optimal path planning", 4th International Symposium on Voronoi Diagrams in Science and Engineering, Glamorgan, 9-11 July 2007: 38-47.

[3] Warren, C.W., Mech. Eng. Dept., Alabama Univ.,"Global path planning using artificial potential fields", IEEE International Conference on Robotics and Automation, Scottsdale, AZ, USA, 1989, 316-321 vol.1

[4] CH Chiang, JS Liu,"Boundary following in unknown polygonal environment based on fast marching method", IEEE Workshop on Advanced robotics and Its Social Impacts, Taipei, 23-25 Aug. 2008: 1-6.

[5] S. Garrido, L. Moreno,"Voronoi diagram and fast marching applied to path planning", IEEE International Conference on Robotics and Automation,Orlando, Florida,May 2006, pp.3049-3054.F

[6]S. Thrun, A.Bücken,"Integrating grid-based and topological maps for mobile robot navigation", Proceedings of the Thirteenth National Conference on Artificial Intelligence, Portland,1996, pp. 1-7.

[7] Joon-Woo Lee, "Improved Ant Colony Optimization Algorithm by Potential Field Concept for Optimal Path Planning",8th IEEE-RAS International Conference on Humanoid Robots, Korea, 2008, 662-667.

[8]Zhang Chibin, Wang Xingsong,"Complete Coverage Path Planning Based on Ant Colony Algorithm",15th International conference on Mechatronics and Machine Vision in Practice, 2008, Auckland, 357-361.

[9] Y.Cen, C.Song,"Path Planning Method for Mobile Robot Based on Ant Colony Optimization Algorithm", 3rd IEEE Conference on Industrial Electronics and Applications, 2008, pp. 298-301.

[10] M. Dorigo, V. Maniezzo, "The ant system:optimization by a colony of cooperating agents", IEEE Trans.Systems Man Cybernet. 1996, 29–42.

[11] Ngo Anh Vien, Nguyen Hoang Viet,"Obstacle Avoidance Path Planning for Mobile Robot Based on Ant-Q Reinforcement Learning Algorithm", Lecture Notes in Computer Science, Springer Berlin/Heidelberg,South Korea, 2007,Volume 4491,pp.704-713.

[12] TAN Guan-Zheng,HE Huan ,"Ant Colony System Algorithm for Real-Time Globally Optimal Path Planning of Mobile Robots", Acta Automatica Sinica,Volume 33, Issue 3, March 2007, Pages 279-285.

[13] G Tan, H He,"Global optimal path planning for mobile robot based on improved Dijkstra algorithm and ant system", Journal of Central South University of Technology, 2006

[14] M. Dorigo, and L.M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," Proc.IEEE Trans. Evolut. Comput.,1(1997) 53–66.

[15] M. Dorigo and T. Stützle, "Ant Colony Optimization". Cambridge, Massachusetts: The MIT Press, 2004.