Proceedings of the 2006 IEEE
International Conference on Mechatronics and Automation
June 25 - 28, 2006, Luoyang, China

Path Planning Based on Ant Colony Algorithm and Distributed Local Navigation for Multi-Robot Systems

Shirong Liu and Linbo Mao
School of Automation
Hangzhou Dianzi University
Hangzhou, Zhejiang 310018, China
liushirong@hziee.edu.cn, mlinbo80@126.com

Jinshou Yu
Research Institute of Automation
East China University of Sci & Tech
Shanghai 200237, China
jshyu@ecust.edu.cn

***Abstract* - This paper presents a decoupled path planning based on ant colony algorithm and distributed navigation with collision avoidance for multi-robot systems. An improved ant colony algorithm is proposed to plan a reasonable collision-free path for each mobile robot of multi-robot system in the decoupled path planning scheme in complicated static environment. The special functions are added into the regular ant colony algorithm to improve the selective strategy. When an ant explores a dead-corner in path searching, a dead-corner table is established and a penalty function is used for the trail intensity updated in order to avoid the path deadlock of mobile robot. A behavior strategy on "first come and first service" is adopted to solve the conflict between moving robots in distributed local navigation. Simulation results show that the proposed method can effectively improve the performance of the planned path, and the individual robots with collision-free can achieve to reach their goal locations by the simple local navigation strategies.**

Index Terms* - *Multiple mobile robots, ant colony algorithm, path planning, distributed navigation.

I. Introduction

Multi-robots path planning with collision avoidance is one of the fundamental problems in multi-robot systems, and also is the basis of performing all kinds of missions, reflecting capability of the robot interacting with the surrounding environment and other robots during motion process. Multi-robot path planning with collision avoidance is devoted to find an optimal or reasonable path from an initial location to a goal location so that the mobile robot is able to move safely through the workspace with collision avoidance. The existing methods for solving the problem of motion planning for multiple robots can be divided into two categories: centralized approach and decoupled approach [1]. In the centralized approach, the configuration spaces of the individual robots are combined into one composite configuration space which is then searched for a path for the whole composite system [2]. In contrast, the decoupled approach first computes separate paths for the individual robots and then resolves possible conflicts of the generated path [3].

A novel decoupled path planning for multi-robot systems and distributed navigation with collision avoidance is presented in the paper. In the decoupled path planning phase, we adopted an improved ant colony algorithm (IACA) to plan the motion path for each robot in the workspace with grids. Aiming at avoiding the possible collision between robots during movement, a behavior strategy on "first come and first service" and a priority strategy are employed. Simulation results show that the planned path by the proposed method is reasonable and efficient, and the individual robots with collision-free can achieve to reach their goal locations by these simple local navigation strategies.

II. Problem Statement

The workspace of mobile robot in 2D environment can be represented by grids with the same size. There are a set of static obstacles with different size and shape in the workspace. The premises and assumptions of our study are stated as follows:

- The mobile robot is assumed to be point-size and occupies only one grid at a time.
- Each robot has an assigned goal, and knows its start and goal positions.
- Each robot is in equal level without any priority in path planning.
- Collision may be caused because of the cross position of planned paths, not considering the collision generated by too nearer distance between robots.
- Each robot moves in an even speed, and its status can be switched instantaneously between the moving with a fixed speed and halting.
- The mobile robot is equipped with range sensors, target detectors, and communication sets.
- The robot may has eight moveable directions (North, East, South, West, NE, NW, SE, SW), and the range detected by the sensors cover the area with eight grids, as shown in Fig.1.

According to above premises, the decoupled path planning and distributed navigation with collision avoidance are adopted. The decoupled approach first computes separate paths for the individual robots and then the strategy of local navigation resolves possible conflicts of paths. The decoupled planning scheme for the individual robots is shown in Fig. 2. An improved ant colony algorithm was developed to plan a path for each robot in the grid workspace with static obstacles. The robot use reactive strategy to avoid local collision in motion. The distributed local navigation scheme of the mobile robot with coordination mechanism is given in Fig. 3. The strategy of "first come and first service" and prioritized rules are employed in coordinating the motion of robots so that the robots are able to reach their goals safely.

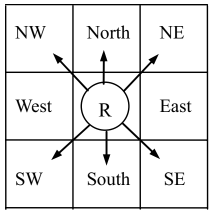


Fig.1 The moveable direction of a mobile robot

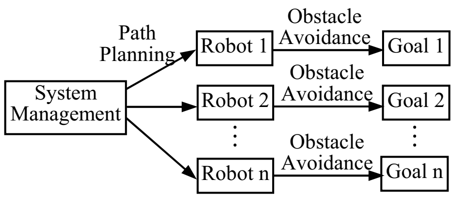


Fig.2 The decoupled planning scheme

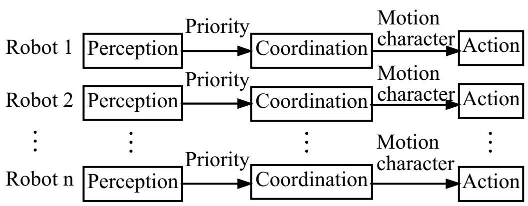


Fig. 3 The distributed local navigation scheme

III. PATH PLANNING BASED ON IMPROVED ANT COLONY ALGORITHM

To find collision-free path for each robot from its initial location to its goal location in multi-robot system, there have been various methods, such as A* algorithm [3], artificial potential field [4], genetic algorithm [5], neural network [6], and so forth. We have developed an improved ant colony algorithm to find optimal or reasonable paths for mobile robots in the workspace with grids. The proposed improved algorithm has been shown that is a powerful tool for solving the reasonable route of robot motion.

A. Ant Colony Algorithm

Ant colony algorithm is inspired on an analogy with real life behavior of a colony of ants when looking for food. The pioneering work has been done by Dorigo [7]. There have been more achievements in NP-hard problems such as Traveling salesman problem (TSP), assignment problem, flow-shop scheduling problem and so on. Ant colony algorithm is a powerful tool for solving hard combinatorial optimization problems, which has been demonstrated by many applications.

Ants are capable of finding the shortest path between a food source and the nest (adapting to changes in the environment) without the use of visual information. This intriguing ability of almost-blind ants has been extensively studied by ethologists. They discovered that, in order to exchange information about which path should be followed, ants communicate with one another by means of pheromone (a chemical substance) trails. As ants move, a certain amount of pheromone is dropped on the ground, marking the path with a trail of this substance. The more ants follow a given trail, the more attractive this trail becomes to be followed by other ants. This process can be described as a loop of positive feedback, in which the probability that an ant chooses a path is proportional to the number of ants that have already passed by that path.

The following describes the state transition rule, global updating rule, and local updating rule in (1)-(6). Equation (1) is a very greedy selection technique that favors the optimum combination of pheromone level and cost. This rule favors transitions toward nodes connected by short edges and large amounts of pheromone. Let j represent the next location of ant k, and it can be determined by

$$j = \begin{cases} \max_{j\in\Gamma(i)}[\tau(i,j)]^{\alpha}[\eta(i,j)]^{\beta} & if \quad q \le q_0 \\ J & otherwise \end{cases} \tag{1}$$

where:

- $\tau(i,j)$ is a positive real value associated to edge (i,j) and represents the accumulated strength of the pheromone left by the real ants.
- $\eta(i,j)$ is a local heuristic function of visibility, defining $\eta(i,j)=T/d_{i,j}$, T is a real value, $d_{i,j}$ represents the distance between locations i and j.
- α and β denote the weight of strength of pheromone and heuristic function of visibility, respectively.
- q is a random value uniformly distributed in [0, 1], and q_0 $(0 \le q_0 \le 1)$ is a parameter: the smaller q_0 the higher the probability to make a random choice. In short q_0 determines the relative importance of exploitation versus exploration in formula (1).
- J is a random variable selected according to the distribution given by formula (2).

$$P_k(i,j) = \begin{cases} \dfrac{[\tau(i,j)]^{\alpha}[\eta(i,j)]^{\beta}}{\sum_{q\in allowed(i)}[\tau(i,q)]^{\alpha}[\eta(i,q)]^{\beta}} & if \quad j \in allowed(i) \\ 0, & otherwise \end{cases} \tag{2}$$

where $P_k(i,j)$ is the probability which ant k transfers from location i to location j, and $allowed(i)$ denotes the feasible domain of ant k in location i.

The pheromone level of the selected location is updated by the local updating rule. In order to diversify solutions obtained by the ants, the amount of the pheromone of the visited edge (i,j) may be regulated by the local updating rule

$$\tau(i,j) = \varphi\cdot\tau(i,j) + (1-\varphi)\Delta\tau(i,j) \tag{3}$$

$$\Delta\tau(i,j) = \frac{\tau^0}{d_j} \tag{4}$$

where $(1-\varphi)$ represents the local pheromone decay parameter $(0<\varphi<1)$, τ^0 is a constant; d_j is the tour length obtained by ant k in this search, and if $\tau(i,j)<\tau_{\min}$, $\tau(i,j)=\tau_{\min}$.

While building a tour (that is, a solution), ants visit edges and change their pheromone level by applying the global updating rule. Global updating rule is done after each ant colony algorithm searching process is completed. Global updating rule is described by

$$\tau(i,j)=\alpha\cdot\tau(i,j)+(1-\alpha)\sum_{k=1}^{m}\Delta\tau_{i,j}^{k} \tag{5}$$

$$\Delta\tau_{i,j}^{k}=\begin{cases}Q/L_{k\min}, & if \quad (i,j)\in tour\ desiried\\ 0, & otherwise\end{cases} \tag{6}$$

where $(1-\alpha)$ is the global pheromone decay parameter $(0<\alpha<1)$, Q is a constant, $L_{k\min}$ is the length of the optimal global tour, $(i,j)\in tour\ desired$ is edge (i,j) belonging to the optimal tour, and if $\tau(i,j)<\tau_{\min}$, $\tau(i,j)=\tau_{\min}$.

B. Improved Ant Colony Algorithm for Path Planning

1) Improvement in Selective Strategy: In the ant colony algorithm, some edges have been went through by ants while some not during the initial stage. According to the basic selective strategy, ants usually choose the edge on which the pheromone is stronger, and thus the search of ants will tend to several local optimal paths so that lose the diversity of the solution. In order to overcome this difficulty in the searching process, we propose that create randomly n trial points between the start and goal, meanwhile n routes planned by ant colony algorithm go through these points. In this way, ants can choose more different paths during the initial stage, so as to obtain diversified solutions.

For simplifying the description of the problem, we assume that (Sx_i,Sy_i) and (Gx_i,Gy_i), $i=1,\cdots,m$, represent the initial location and the goal location respectively. As an example of Robot 1, n points are created randomly in the region W that is a quadrilateral area and four points of the area are denoted as (Sx_1,Sy_1), (Sx_1,Gy_1), (Gx_1,Sy_1), (Gx_1,Gy_1). Using the basic ant colony algorithm n paths through n random points in the region W can be obtained. Obviously, the shortest path from the start to the destination can be easily found among these paths, and the shortest path is chosen to update the global pheromone. In this way, the diversity of solution is increased during the initial stage, and the tendency of the solution falling in local optimum is decreased.

2) Solution for "Deadlock": "Deadlock" problem in robotics means that the robot is probable not to go forward and loses moveable possibility. Similarly it is probable to appear the status of "deadlock" in robot path planning, called as the route deadlock. The problem of route deadlock also occurs in the path planning with the basic ant colony algorithm. The definition of "deadlock" in this paper is: Ants enter into the location which is surrounded by obstacles during searching a path, thus losing the capability to go forward. We put forward establishing a dead-corner table and introducing a penalty function to solve this problem. The dead-corner is such a location in which ants come into the status of deadlock, as shown in Fig.4. If an ant comes into dead-corner in path searching process, the location of dead-corner is listed in dead-corner table and the ant returns to the former location, and then searches the next location newly.

As a consequence of local updating, the pheromone of edges around the dead-corner is increasing so that ants tend to choose these edges in next iterative search. It is likely to increase the time of finding optimal path, and even not find the optimal path. We take use of a penalty function to prevent this situation occur. If an ant encounters a dead-corner, we use a penalty function instead of local updating rule. The penalty function is defined below

$$\tau(i,j)=\lambda\cdot\tau(i,j) \quad 0<\lambda<1 \tag{7}$$

The penalty function assures that pheromone of the edges around dead-corner decreases, resulting in that the ant does not choose those edges in next iterative searching process. Thus, the situation of route deadlock is avoided, and the efficiency of searching for the optimal path is improved simultaneously.

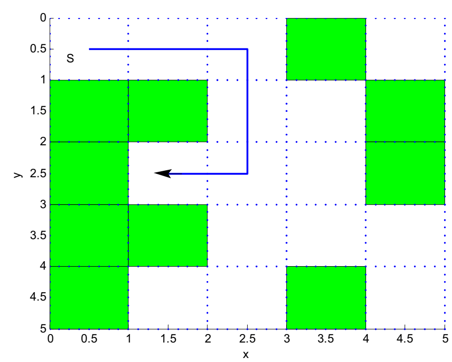


Fig.4 Illustration of dead-corner and route deadlock

C. Improved Ant Colony Algorithm (IACA)

The basic steps of the improved ant colony algorithm for path planning are presented below:

Step1 Initialize the times of searching and pheromone of each edge, i.e., N=0 and $\tau(i,j)=C$ (constant). The initial and goal locations are determined respectively, such as (Sx_i,Sy_i), (Gx_i,Gy_i), i=1,…,m. Create n random points in region W, a quadrilateral area with four points, (Sx_i,Sy_i), (Sx_i,Gy_i), (Gx_i,Sy_i) and (Gx_i,Gy_i).

Step2 If N=0, search the paths from the initial location (Sx_i,Sy_i) to the goal location (Gx_i,Gy_i), i=1,…,m; simultaneously find the routes that the n random points are used as the start points and the initial location (Sx_i,Sy_i) and the goal location (Gx_i,Gy_i) are viewed as the end points in the region W, i=1,…,m. Connect the path from the random point to (Sx_i,Sy_i) and the path from the random point to (Gx_i,Gy_i), thus, n shortest paths L_{ij}($j=1,\cdots,n$) which pass through the random points are obtained. Compare L_{ij} ($j=1,\cdots,n$) with the optimal path searched directly from (Sx_i,Sy_i) to (Gx_i,Gy_i), and find out the shortest path among these paths. If N>0, search the shortest path from the initial location to the goal location directly (the steps for searching are shown in step3~step5).

Step3 Put all the ants at the initial location, and choose the next location according to the state transition rule, i.e., formula (1). Perform the local updating rule (3) after completing the selection.

Step4 Examine whether there is an ant reaching the goal or not. Return to step2 to choose the next location if there is no ant getting to the goal, else turn to step5.

Step5 Compute the length of the path if an ant reaches the goal location. Compare the shortest path in this searching process with the shortest path in the past. When the path is superior to the path in the past, we save it instead of the path in the past.

Step6 Update the global pheromone by the updating rule (5).

Step7 Do N=N+1. If N<NC (predetermined searching times), put all the ants at the initial location newly, and then return to step2; else return the optimal route for each robot from the start to the terminal; finally, algorithm ends.

IV. Distributed Navigation with Collision Avoidance

Each robot has a planned path from its start to its destination. Then the robot will go to the goal location from the fixed initial location avoiding the obstacles and the other robots. There are various methods for dealing with conflicting between the moving robots, such as selecting a robot to stop randomly [8], traffic rules [9], choosing a steering direction [10], prioritized planning [11], artificial potential method [12], and so on. The reason of collision generation is that the paths are crossed each other. In order to avoid collision, we use the strategy of "*first come and first service*" and prioritized rules to coordinate the motion of robots.

Suppose there are m robots. R_i (i=1,2,…m) represents the robot i; $L_k(i,j)$ is the length from location i to location j of robot k; $t_k(i,j)$ is the time of robot k moving from location i to location j; C is the location of collision, v_i (i=1,2,…m) represents the velocity of robot i. If robot i is in the region of collision, we make sure that robot i occupies the region of collision C where is occupied by only one robot at any time. In the process of robot motion, the robot is able to know whether the next location may be occupied or not by local perception and detection of mobile robots.

For simplification, here only consider two mobile robots in the workspace. Suppose that R_1 possibly conflict with R_2 in the collision area C. In order to avoid collision, the proposed collision-fee strategies are below:

- If R_1 detects that C is occupied by R_2, R_1 pauses till R_2 leave C; similarly, if R_2 detects that C is occupied by R_1, R_2 pauses till R_1 leave C.
- If R_1 and R_2 come into the free region C at the same time, we adopt prioritized rules to solve it as follows: if V_1>V_2, R_1 enters C first, and R_2 pauses till R_1 leave C; on the contrary, R_2 enters C first, and R_1 pauses till R_2 leave C.

The rules of collision avoidance are presented below:

1) For R_1:

```
If  ((R2& C)!=0)  then  //part of R2 enter into C
    R1 pause; R2 move;
elseif
    ((R1 (next)& C)!=0)& ((R2 (next)& C)!=0)
then   if  (V1>V2)   then
           R1 move; R2 return and pause;
else
   R1 return and pause; R2 move;
end
end
```

2) For R_2:

```
If  ((R1& C)!=0)  then  //part of R1 enter into C
    R2 pause; R1 move;
elseif
    ((R1 (next)& C)!=0)& ((R2 (next)& C)!=0)
then   if  (V2>V1)   then
           R2 move; R1 return and pause;
else
   R2 return and pause; R1 move;
end
end
```

V. Simulation Studies

The proposed approach has been tested in extensive simulations. Let the workspace be divided by square grid with wide of one meter. Si and Gi represent the start location and the goal location of robot i, respectively. Consider the path planning and navigation with collision avoidance of two robots in the same workspace. To illustrate the validity of the proposed approach, we take two examples discussed by reference [5]. The planned paths for individual robots in a simple workspace and a complex workspace are given in Fig.5 and Fig.6, respectively. Genetic algorithm (GA) based path planning was employed in reference [5]. By comparing the trajectories planned by the two approaches, we can find that the path generated by the proposed method is more reasonable than the path planned by GA in reference [5]. The trajectory length value of the above two approaches in two workspaces are listed in Table I. It shows that the length of the path planned by IACA is much shorter than that of the path planned by GA.

For collision avoidance, the distributed navigation rules proposed in Section IV is used to deal with the conflict between two robots. Considering an example of Fig.5, suppose that the velocities of two robots R_1 and R_2

are $v_1 = 2.1\mathrm{m}/s$, $v_1 = 3.0\mathrm{m}/s$, respectively, and sampling period is $T = 0.1s$ ($0 < T < 1/\max(v_1, v_2)$). Under ideal condition, when t=6.7*s*, R_1 will conflict with R_2 at the location (19,11), i.e., collision region *C*. Obviously R_2 enters *C* first, and R_1 pauses till R_2 leave *C* according to the rules for collision avoidance. The all process of collision avoidance between robots is shown in Fig.7. At t=6.6*s*, before two robots encountering, the moving trajectories of individual robots are shown in Fig.7 (a). Then, R_2 move first, and R_1 pauses to wait for R_2 going through the collision region *C*, as shown in Fig. 7(b). When R_2 leaves the region of collision, R_1 start to move and go to its goal location. This process is shown in Fig. 7(c). If there is no conflict between R_1 and R_2, R_1 and R_2 take 16.6*s* and 12.2*s* from each start location to goal location, respectively. But there exists a conflict region *C* such as in Fig. 7, R_1 and R_2 take 17.7*s* and 12.2*s* from each start location to goal location respectively. Under this status the moving time of R_1 obviously increase.

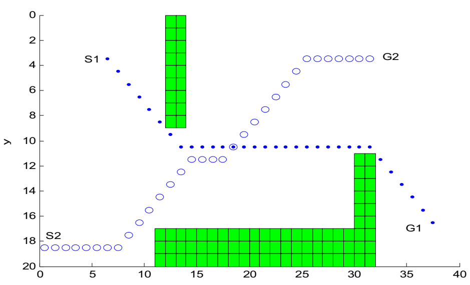


Fig.5 The planned paths in environment I

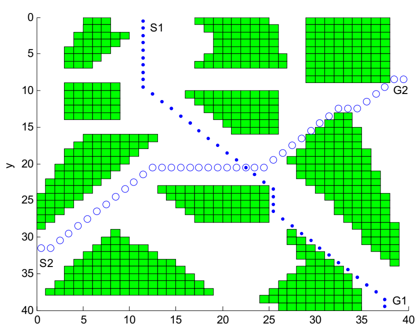
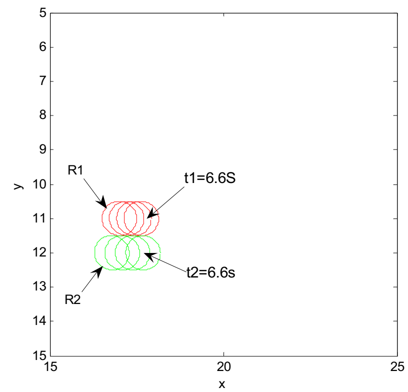


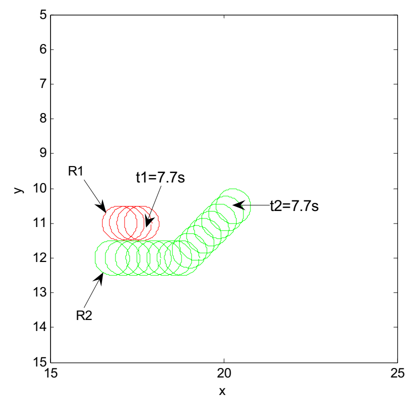
Fig.6 The planned paths in environment II

TABLE I
COMPARISON OF TRAJECTORY LENGTH WITH TWO METHODS

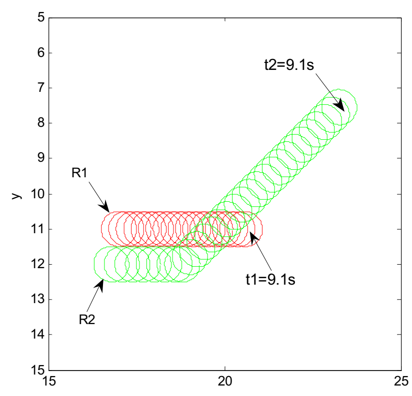
Approach	Environment I		Environment II	
	$L_1(S_1,G_1)$	$L_2(S_2,G_2)$	$L_1(S_1,G_1)$	$L_2(S_2,G_2)$
Improved ant colony algorithm	36m	37m	50m	49m
Genetic algorithm in reference [6]	61m	41m	53m	57m



(a) Trajectories of R_1 and R_2 (t=6.6s)



(b) Trajectories of R_1 and R_2 (t=7.7s)



(c) Trajectories of R_1 and R_2 (t=9.1s)

Fig.7 Illustration of process for collision avoidance

VI. CONCLUSIONS

The decoupled approach can be effectively applied to a class of motion planning problem that each robot has its independent goal in multi-robot systems. The improved ant colony algorithm is able to plan an optimal or reasonable path in static environment with different obstacles. The collision avoidance strategy with “first come and first service” and the priorities make the robots navigate safely. Extensive simulations have shown that the proposed approach is very simple and efficient.

ACKNOWLEDGMENT

The work is supported by Natural Science Foundation of Zhejiang Province, China (Grant No.Y104560) and Research Start-up Foundation of Hangzhou Dianzi University, China (Grant No.KYS091505043).

REFERENCES

[1] J Latombe, *Robot Motion Planning,* Kluwer Academic Publishers, Boston, 1991.

[2] I H Suh, H J Yeo, and J H Kim et al, "Design of a Supervisory Control System for Multiple Robotics Systems," *IEEE IROS'96*, pp. 332-339, 1996.

[3] M Bennewitz, W Burgard, and S Thrun, "Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots," *Robotics and Autonomous Systems*, vol.41, pp. 89-99, 2002.

[4] J Barraquand, B Langois, and J C Latombe, "Numerical potential field techniques for robot path planning," *IEEE Transactions on Robotics and Automation, Man and Cybernetics,* vol.22, no.2, pp.224-241, 1992.

[5] S Liu, Y Tian, and J Liu, "Multi mobile robot path planning based on Genetic Algorithm," *Proceedings of the 5th World Congress on Intelligent Control and Automation*, pp. 4706-4709, 2004.

[6] C-H Fan, W-D Chen, and Y Xi, "Hopfield neural networks for path planning in dynamic and unknown environments," *Control Theory and Applications*, vol.21, no.3, pp. 345-350, 2004.

[7] M Dorigo, V Maniezzo, and A Colorni. "Positive feedback as a search strategy," Technical Report 91-016, Politecnico di Milano, Italy, 1991.

[8] S Oliver, M Saptharishi, J Dolan, A Trebi-Ollennu, and P Khosla, "Multi-robot Path Planning by Predicting Structure in a Dynamic Environment," *Proceedings of the First IFAC Conference on Mechatronic Systems*, Vol. II, pp. 593-598, September, 2000.

[9] S Kato, S Nishiyama, and J Takeno, "Coordinating mobile robots by applying traffic rules," *IEEE international Conference on intelligent robots and systems,*vol.3, IROS'92, Raleigh,(USA), pp. 1535-1541, July 1992.

[10] B L Brumitt and A Stentz, "Dynamic mission planning for multiple mobile robots," *In Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2396-2401, 1996.

[11] M Erdmann and T Lozano-Perez, "On multiple moving objects," *In Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, pp. 1419-1424, 1986.

[12] Prahlad Vadakkepat, Kay Chen, and Wang Mingliang, "Evolutionary artificial potential fields and their application in real time robot path planning," *Congress of Evolutionary Computation*, San Diego, California, pp. 256-263, 2000.