OPTIMAL PRIORITY SELECTION FOR MULTI-AGENT TASK EXECUTION

Sang Hoon Ji* **Jeong Sik Choi*** **No San Kwak*** **Beom Hee Lee****

E-mail: robot91@snu.ac.kr jsforce@netian.net robot97@snu.ac.kr bhlee@asri.snu.ac.kr
School of Electrical Engineering
Seoul National University, Seoul, Korea

* *student*
** *professor*

Abstract: In this paper, effect of priority order on navigation performance was analyzed. And priority order was selected to give optimal navigation performance considering robots' trajectories. To include chain interference in the trajectories of robots with lower priorities, the problem was formulated as the multiple linear equations using collision map analysis. The priority order problem was then converted to an optimal priority selection for multi-agent task execution problem. The solution to the optimal problem was obtained using the dynamic programming approach. Numerical examples were finally presented to demonstrate the significance of the proposed method for optimal priority selection for multi-agent task execution.

Keywords: optimal priority assignment, multimachine, robot navigation, deadlines, scheduling

1. INTRODUCTION

Collision free motion planning for multiple robots that have kinodynamic constraints and environmental constraints is an important task. Kinodynamic planning problem for even one robot is known to be too complex to be solved by mathematical methods(Canny, et al, 1988). Moreover, multiple robot scheduling turns out to be NP-hard problem and not to be solved mathematically(Arinivas, 2002).

Researches attempted for solving the combined problem has been based on numerical methods. They are classified into decentralized heuristic methods or centralized deterministic methods. The former are very simple but not guaranteed to give a solution for this problem. The latter give an optimal solution for multi-agent but have a few of drawbacks in applying the problem of robots with limited activities. Moreover, the more are the number of robots, the higher are complexities of these algorithms.

The typical ways of planning multi-agent in centralized algorithms are priority based speed tuning methods and PRM based methods. The latter gives multiple solutions to the multiple robots scheduling problem and its activities expands to dynamic environmental problems(Sanchez and Latombe, 2002; Guang, et al, 2003). But it doesn't an optimal solution for completion time. The former coordinates the multi-agent, but performance is affected by priority order and need global environment information.

Strinvas Akella showed that the trajectory coordination problem for multiple robots is closely related to job scheduling (Arinivas, 2002; Jufeng, 2003). The constraint-based approach formulates a scheduling problem as one or a set of Constraint Satisfaction Problems(CSP). Constraint propagation techniques, which exploit the constraints, are used to limit the search space so that the computation time to solve the CSPs can be greatly reduced (Haoxun, et al, 2001).

The method, what is called collision map algorithm, is a kind of constraint propagation technique and uses a window based approach which is known to give a good solution to job scheduling problem with dynamic constraints(Lee, 1987; Christons and Ali, 2000; Haoxun, et. al, 2001). It is a powerful tool for collision-free scheduling two robots having common activity regions and is recently adjusted to give solution for multiple moving robots.

But the efficient planning method has several problems as follows.

1) Since navigation performance in one priority order differs from that in others, all priority orders of robots should be considered to obtain optimal navigation solution.

2) It makes algorithm's scalability poor to consider all priority orders of robot in that the total number of computation increases very fast in proportion to the factorial of all the number of robots.

A Good Scheduler for multi-agent should give optimal priority selection within a reasonable time and scalable to the total number of robots. To do this, scheduler should have ability to limit the number of significant robots and make complexity of centralized methods small. The method which expresses the paths precedence relations as priority graph and coordinate robots' priority was presented (Maren, 2001). But it doesn't consider dynamic characteristics of robots, so it cannot give an optimal solution.

The purpose of this paper is to give a method to select the optimal priority order for multi-agent task. In section II dynamic properties related to collisions among robots are presented. And then in section III they are expressed as functional parameters so that they are used to select optimal priority order. Here the priority selection problem for collision-free motion planning is then converted to an optimal priority selection problem. In section IV a method based on dynamic approach that gives the solution that minimizes the above specified performance index is presented. The method is based on earliest deadline first (EDF). In section V numerical examples are presented to demonstrate the significance of the proposed method for selecting priority order of multi-agent task.

2. PRIORITY SELECTION PROBLEM

2.1 . Collision Free Condition

It is assumed that a circle mass in R^d (d=2,3) moves from a start position to a goal position along the given path. In the motion, it has bounds on the magnitude of the commanded accelerations ***a(t)*** and velocity bounds for a smooth path. Because each robot has its radius and areas, we define the position of robots as grid set as follows:

$$O_i(p_i(x,y,t))\begin{cases}=1, \text{this cell belongs to } R_i \text{ at time t}\\ =0, \text{otherwises}\end{cases}$$

where $p_i(x,y,t)$ is the position of $\mathrm{R_i}$ (1)

When two robots R_1 and R_2 have a collision, there is a point where both of all occupied function are 1. For example, in figure 1 the occupied functions of two robots for the cell ***i*** are 1 at some time t, so they collided with each other at that time. But the occupied function of R_2 for the cell ***j*** is always 0, so two robots will no collision for the cell ***j***.

We can deduce collision free condition at time t as follows:

$$O_i(p_i(x,y,t))\cap O_j(p_j(x,y,t))\neq\Phi,\ \text{for any pair } \mathrm{R_i} \text{ and } \mathrm{R_j} \qquad (2)$$

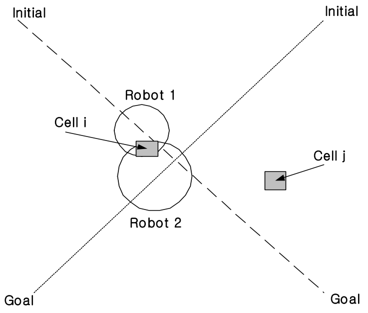

Fig. 1. Two Translating Robots with Collision Cells.

The condition in (2) is powerful, which is not pertinent when the search area is wide and so there are a number of the cells considered. Thus for simplicity we use path segments used in (Roszkowska, 2005). In figure 2, three robots have common regions, and each path is divided several path segments. For example, the path of R_2 is divided to P_{21}, … , P_{27}. A path segment P_{22} is a region in which R_2 may collide with R_1. A collision between R_2 and R_1 may occur when the robots are located on path segments related to collision. So we defined collision free condition by using path segments as follows:

[Collision Free Condition]

No Robot is allowed to be located on the path segments which the other robots are located the segments related to

For example, R_1 is not located on path segment P_{12} when R2 is located on path segment P_{22}.

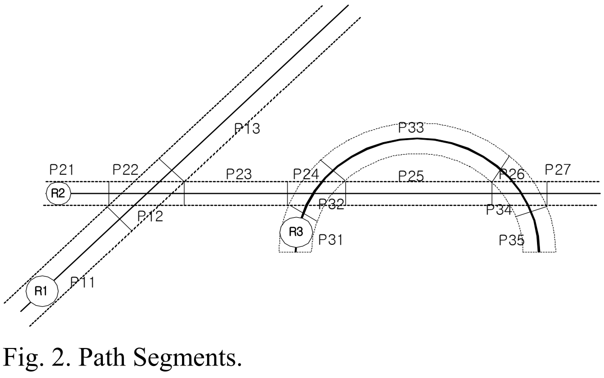

Fig. 2. Path Segments.

Since paths are assumed to be fixed, we can locate a robot from the travel distance of the robot. So we can use traveled length index s to express the place of the robot along the path. Traveled length index s is to be defined to express the place of the robot along the path (Lee, 1987). For example, if R_1 moves along the straight line path χ, s is defined as

$$\chi(kT) = \chi(k_0 T) + s(\chi(k_f T) - \chi(k_0 T)) \qquad (3)$$

Therefore we collision free condition for multi-agents as follows:

[Collision Free Condition]

No robot is not allowed to have a TLI within the interval which those of the other robots are within the interval related to.

For example, in figure 2, if we notate entrance point of P_{22} is S_{21}, exit point of P_{22} is S_{22}, and those of P_{12} are S_{11}, S_{12}, the travel length index of $R_1$is not allowed to be assigned to the value between S_{11} and S_{12} when that of R_2 is located between S_{21} and S_{22}.
So we can guide a robot safely when we know the trajectories of the others which their travel length indexes are drawn from.

2.2 . Collision Map

Collision map is based on time windows on traveled length versus servo time curve (TLVSTC). When the time windows for a robot drawn, constraint satisfaction algorithm is used to find a feasible trajectory for the lower robot (Lee, 1987). In figure 2 a collision happens because some fraction of the lower robot's trajectory lies in collision boxes. This situation is when eq.(2) is false. Therefore the lower robot's trajectory must be shift right in order that trajectory of the lower robot has no cross set with collision boxes.

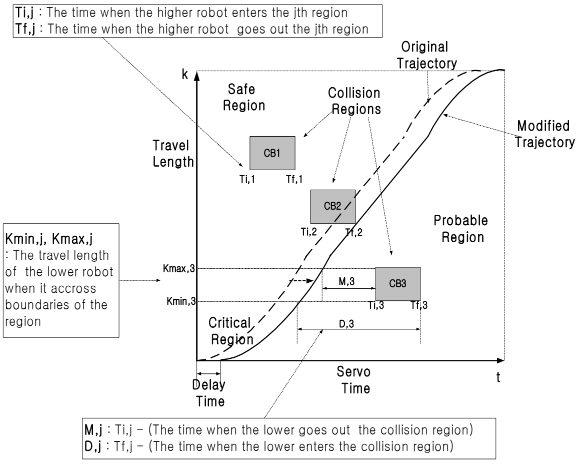

Fig. 3. Collision Map

2.3 . Priority Selection Problem

Priority order of multi-agent is very important because priority order of two robots has effect on the completion time. For Example collision maps for two robots are similar in figure 3 where T1, T2, T3, and T4 are servo time variables defined collision map. And L1, L2, L3, and L4 are travel length indexes. As in figure 4, if one collision box has long width but short height, the other box has short width and long height. To reduce delay time, collision box should have short width. Therefore robot A that is fastest should have the highest priority.

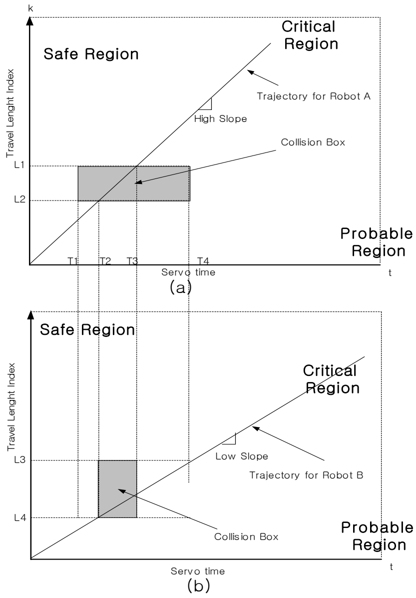

Fig. 4. Collision Maps for Two Robots.

But only robot's speed and travel length in collision region cannot determine optimal priority order. First reason is that the criterion for collision avoidance on collision map can be overly conservative.
Second reason is chain interference among more than three robots. For example in the three robots case in figure 5, Robot 1 is the highest one and Robot 2 is the second, Robot 3 is the least important robot. It is found that collision box between Robot 2 and Robot 3 in (b) is changed by scheduling Robot 2 to avoid the collision region in (a). It makes the problem of optimal priority selection for multi-agent using collision maps complex.

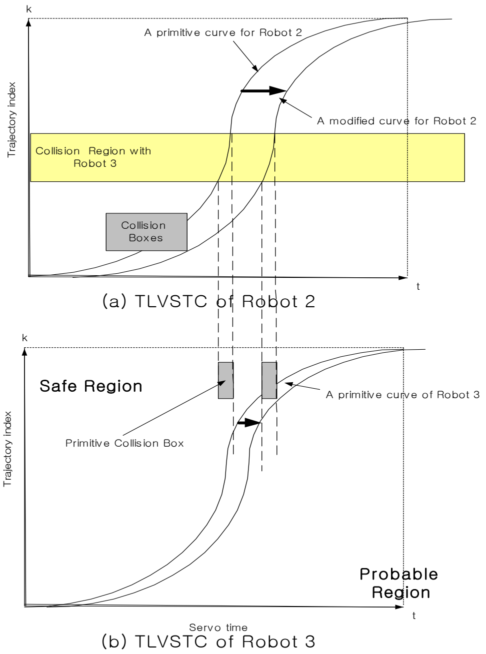

Fig. 5. Chain-Relation among Three Robots

Due to these dynamic properties it is impossible to find optimal priority order by using mathematical methods. Especially as the number of robots increases the complexity of calculation is higher fast. Therefore heuristic methods are needed to obtain optimal priority order for multi-agent task.

3. PROBLEM FORMULATION

3.1 . Collision Free Motion Formulation

In order to formulate the problem of making optimal collision-free motion for multiple robots, it is necessary to define variables related to dynamic characteristics of collision boxes. Two variables for each collision box are defined in order to formulate collisions. One is defined by the time needed to shift the trajectory in critical region to the left to safe region as M_j in figure 3. The other is defined by the time needed to shift the trajectory in critical region to the right to probable region as D_j in figure 3.

With the variables collision regions can be classified into three groups. One is safe region group in which the two robots don't always collide with each other, another is critical region in which those collide with each other now and need the change in the velocity profile of the lower priority robot, the third is probable region in which those don't collide now but may collide with each other due to the changes in the velocity profile of the lower priority robot to avoid collisions in earlier collision regions

Table 1 Classification of Collision Regions

Regions	M_j	D_j	Collision
Safe Region	< 0	< 0	No
Critical Region	> 0	< 0	Yes
Probable Region	> 0	> 0	No

It is also possible to formulate collision-free coordinated motion for multiple robots with two variables. When each robot is assigned one priority number to, they can be coordinated as follows:

Minimize $t_{complete} = \max(t_1^{complete}, \ldots, t_{N_R}^{complete})$

Subject to Collision free condition:

$$(\delta_{p,q,k} - 1)M_{p.q.k} + \delta_{p,q,k} D_{p.q.k} \le 0 \qquad (4)$$

Where,

$N_{p,q,c}$: total collisions between R_p and R_q

$1 \le p < q \le N_R$: Priorities

$0 \le k \le N_{p,q,c}$

$\delta_{p,q,k}$: 1 if $M_{p,q,k} < 0$, otherwise it is 0

Above formulation is called Priority Based Collision-free Linear Programming Formulation and in this procedure, Delay Time is defined as follows:

- Rotate Collision Map until the trajectory of a robot align to Y axis and Project Collision Boxes rotated to X axis. Mark the interval as $M_1, \ldots, M_k$
- If the some Ms has common range, they merge. Mark the linked interval as $D_1, D_2, \ldots D_k$
- If D1 has not 0 or no elements, Delay Time is 0
- Otherwise, determine Delay Time as follows:
  - : Rotate the line (x = Maximum of D_1) by inverse angle of Step 1
  - : Obtain cross at X axis of the rotated line

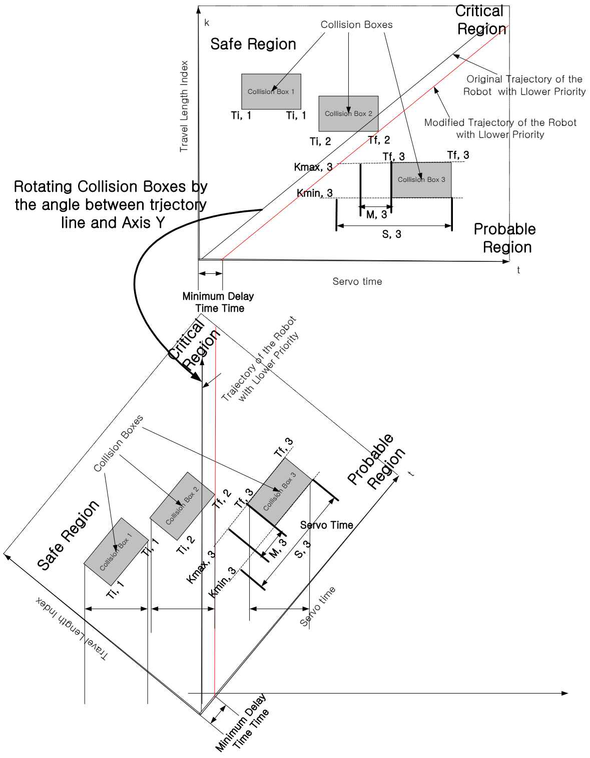

Fig. 6. Determination of Delay Time

3.2 . Object Function *J*

An object function J to express efficiency of priority order as follows:

$J = \max(J_p)$

Where, $J_p = a * T_{t,p} + b * T_{d,p} - c * T_{c,p}$ (5)

P is robot ID number and *a, b, c* is coefficient variables.

Table 2 Variables Related to Object Function

Variable name	**Meaning**
$T_{t,p}$	Traveling time for robot p
$T_{d,p}$	Delay time for robot p
$T_{c,p}$	Time constraints to robot p

Both $T_{t,p}$ and $T_{c,p}$ are to be determined in advance. $T_{d,p}$ is obtained by not mathematical methods but analysis based on collision map. Pessimistic definitions for optimal priority selection are used to draw $T_{d,p}$ in J(.) quickly. When only *c* is 0 or all robots has the same time constraints, optimization of J is equal to minimization of $PT_{a,p}$. When none of all coefficient is not 0, it is equal to maximization of $PT_{r,p}$, so-called Reserved Time Problem(RTP).

Table 3 Variables Related to Performance Index

Variable name	**Meaning**	**Definition**
$PT_{d,p}$	**Pessimistic Delay Time**	$\sum$ (Length of Collision Boxes)
$PT_{a,p}$	**Pessimistic Arrival Time**	$T_{t,p} + PT_{d,p}$
$PT_{r,p}$	**Pessimistic Reserved Time**	$T_c - PT_{a,p}$

4. SOLUTION TO OPTIMAL PRIORITY SELECTION PROBLEM

4.1 . Solution to Reserved Time Problem

It is a problem how to coordinate multi-agent so that all of them meet dead lines assigned to each robots. In this case object function J is defined as maximization of $PT_{r,p}$. Its solution for the RTP is based on the bottom level EDF scheduler algorithm (Park, *et al*., 2003).

Quick Search Algorithm for Optimal Priority Selection(QSA) :

[Step 1] Assign all robots to Urgent Group(UG), one to Iteration Number(I).

[Step 2] Calculate $PT_{r,p}$ for all robots in UG by using collision map.

[Step 3] Classify robots by signs of $PT_{r,p}$'s such that robots with positive $PT_{r,p}$ move to N(I) Group. And increase I by 1.

[Step 4] If there is no robot in UG, go to Step 7.

[Step 5] With no change in UG from previous iteration, go to Step 2.

[Step 6] Examine all the priority order for robots in UG until all robots meet their dead line. Without no priority order in which all robot meet each own time constraint, return FALSE

[Step 7] Assign priority to robots in order UG, N(I), N(I-1),…, N(1). Robot order in any Group is not important, but only priority order in Urgent Group is important. So we determine priority order in UG with the result in Step 6, and return TRUE

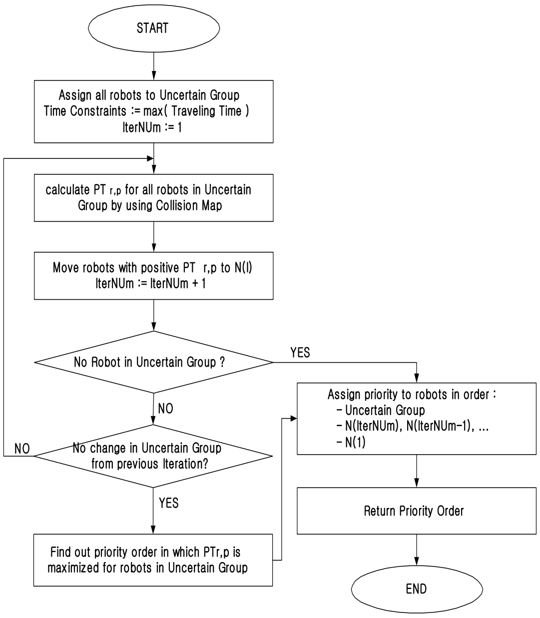

Fig. 7. Block Diagram of QSA

4.2 Optimal Priority Selection Problem for Multi-agent

It is a problem how to coordinate multi-agent to minimize $PT_{a,p}$. The problem is to be converted to RTP by assigning the same time constraints to all robots. The maximum of $T_{t,p}$ was used as the time constraint, and step 6 was modified as follows:

[Step 6'] Examine all the priority order for robots in UG until all robots meet their dead line. Without priority order in which all robot meet each own time constraint, determine an optimal priority order by priority order in which maximum of $T_{t,p}$ is optimized.

5. SIMULATION RESULTS

The following experiments were executed 20 times on PCs with 2.4GHz processor and 512Mbytes RAM.

- Generate robots' tasks. Constraints are that each robot should not collide at start and goal with the others and only one collision happens between two robots.
- case 1 : Coordinate robots for all priority orders.
- case 2 : Coordinate robots based on QSA
- case 3 : Coordinated robots based on QSA with an adaptive deadline

In case 3 we applied QSA iteratively as follows:

[step 1] $E_0 = \max(T_i)$

[step 2] Apply Quick Search Algorithm

[step 3] If there is no robot to meet a Deadline,

$$E_{i+1} = (E_{i-1} + E_i)/2\,,$$

else if there is a number of robots ,

$$E_{i+1} = (E_i + \max(T_i + PD_i))/2$$

else, consider all the priority order in UG.

Success Event is defined as when the arrival time in case 2 is equal to that in case 1. Its probability is named as Success Ratio which represents valid of the results obtained by using the method presented in this paper. In figure 8, Success Ratio was almost 1 when the number of robots was below 12. The success Ratio decreased because there is constraint to computation time. It is concluded that if it is possible to schedule 12 robots quickly, it is also possible to schedule any number of robots within a short time.

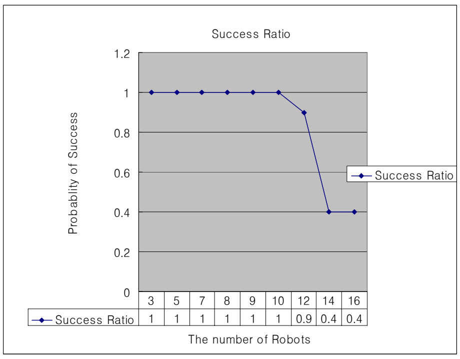

Fig. 8. Success Ratio vs. Number of Robots

In figure 9, thanks to the decreased number of robots in UG, seek time for case 2 was much shorter than that for case 1. But between 12 and 15 robots in UG was with possibilities more than 68 percentages is large for real-time application yet. In case 3 after the specified iterations(we used 5 iteration), the number of robots in UG is lower than that in the case 2. But it increased also.

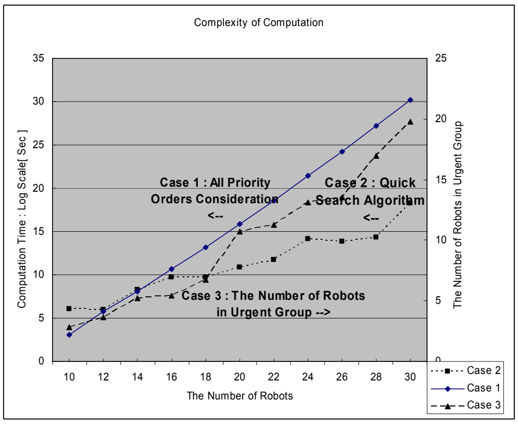

Fig. 9. The Seek Time and the Number of Robots in Urgent Group.

In figure 10, arrival times for two cases were compared. When the computation time is restricted, the arrival time for Case 2 is similar as that for Case 1, and was worse. It is due to local minima of the method in this paper. So in order to optimal solution, all priority order for the robots in Urgent Group should be investigated. However, it takes several hours to coordinate the robots when the number of robots in UG is bigger than 10. Therefore research for the reduction of computation time with below 15 robots is needed in order that the method is applicable to real time coordination of the multi-agent.

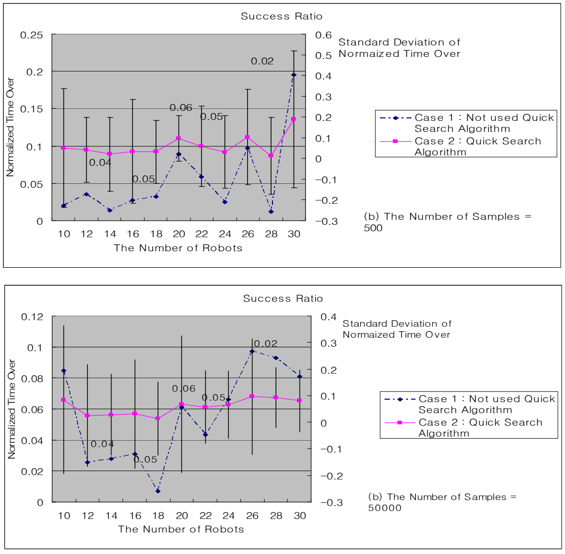

Fig. 10. The Success Ratios for two Cases with (a) 500 Samples (b) 50000 Samples.

6. CONCLUSUION

To select optimal priority for the robots with dynamic constraints, the effect of priority order on navigation performance was analyzed and the multiple linear equations were expressed by using collision map analysis. A solution to the problem was presented by using the dynamic programming approach. This method was applied to the coordination of multi-agents. This method succeeded in reducing the number of significant robots, but is not applicable to on-line scheduler yet. When the method is combined quick coordinator for 15 robots, any number of robots are to be coordinated in real-time.

ACKKNOWLEGEMENT

This research was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea, and by the BK21 Information Technology at Seoul National University.

REFERENCES

Arinivas Akella and Seth Hutchinson(1998), Coordinating the Motions of Multiple Robots with Specified Trajectories, *Proceeding of the 2003 ICRA*, pp. 624-631.

B.H. Lee and C.S.G. Lee(1987). Collision-Free Motion Planning of Two Robots, *IEEE Tran. on Systems, Man, And Cybernetics*, Vol. SMC-17, No.1, pp. 21-31 .

Canny J., Reif J., Donald B. and Xavier(1988). On the Complexity of Kinodynamic Planning, *Foundations of Computer Science, 1988., 29th Annual Symposium on* , pp. 306 – 316.

Christons Dimopoulos and Ali M.S. Zalzala(2000). Recent Developments in Evolutionary Computation for Manufacturing Optimization: Problems, Solutions, and Comparisons, *IEEE Trans. on Evolutionary Computation*, Vol.4, No.2, pp. 93-113.

Guang Song, Thomas S. and Amato N.M.(2003). A General Framework for PRM Motion Planning, *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, Vol.3 pp. 4445 – 4450.

Haoxun Chen, Luh P.B. and Lei Fang(2001). A Time Window Based Approach for Job Scheduling, *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, Vol. 1, pp. 842-847 .

J. Park, M. Ryu and S. Hong(2003). Fair Real-Time Resource Allocation for Internet End System's QoS Support, *Lecture Notes in Computer Science*. vol. 2713, pp. 764-769.

Jufeng Peng, Arinivas Akella(2003). Coordinating the Motions of Multiple Robots with Kinodynamic Constraints, *Robotics and Automation, 2003. Proceedings 2003 ICRA. IEEE International Conference on*, Vol. 3, pp. 4066-4073.

Elzbieta Roszkowska(2002). Supervisory Control for Multiple Mobile Robots in 2D space, *Third International Workshop on Robot Motion and Control*, 187-192.

Maren Bennewitz, Wolfram Burgard and Sebastian Thurn(2001). Optimizing Schedules for Prioritized Path Planning of Multi-Robot Systems, *International Conference on Robotics & Automation*, pp. 271-276.

Sanchez G. and Latombe J.-C.(2002). Using a PRM Planner to Compare Centralized and Decoupled Planning for Multiple-robot Systems, *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on* , Vol. 2 , pp. 2112 – 2119.