COORDINATIVE BALANCING IN EVOLUTIONARY MULTI-AGENT-ROBOT SYSTEM USING GENETIC ALGORITHM

Takanori SHIBATA[1] and Toshio FUKUDA[2]

1 Bio-Robotics Division, Robotics Department
Mechanical Engineering Laboratory
Ministry of International Trade and Industry
1-2 Namiki, Tsukuba 305, Japan
Phone: +81-298-58-7299
Fax: +81-298-58-7201
E-mail: shibata@mel.go.jp

2 Dept. of Mechano-Informatics and Systems, Nagoya University,
1 Furo-cho, Chikusa-ku, Nagoya 464-01, JAPAN
Phone: +81-52-781-5111 ext. 4478
Fax: +81-52-781-9243
e-mail: d43131a@nucc.cc.nagoya-u.ac.jp

Abstract

This paper presents a new strategy for motion planning of multiple robots as a multi-agent system. The system has a decentralized configuration. All the robots cannot communicate globally, but some robots can communicate locally and coordinate to avoid conflicts for public resources. In such systems, it is difficult for each robot to plan its motion effectively while considering other robots, because the robots cannot predict the motion of other robots as an unknown environment. Therefore, each robot only determines its motion selfishly while considering a known environment. In the proposed approach, each robot plans its motion while considering the known environment and using empirical knowledge. The robot considers its unknown environment including the other robots in the empirical knowledge. The Genetic Algorithm is applied to optimize the planning of the motion of each robot. Through iterations, each robot acquires knowledge empirically using fuzzy logic, and the system behaves efficient evolutionary. For an illustration, this paper deals with path planning of multiple mobile robots and performs simulations.

I. INTRODUCTION

Autonomous robots which perform tasks without human operators are required in many fields. When an autonomous robot works at its tasks, it is necessary to have knowledge and to decide its motions by itself. In the conventional robot, human operators give the robots knowledge in advance in top down manner. However, there are limitations of knowledge given by human operators and memory of the robots. Furthermore, in unknown environments, human operators can not give knowledge. Therefore, the autonomous robots must have adaptation function to unknown environments and learning function to memorize experiences and utilize them later. It is important to utilize both success and failure. It is also important to update the knowledge. These process proceed in bottom up manner.

For the intelligent control of robots, the hierarchical intelligent control scheme has been proposed by Shibata et al. (1992, 1993). The scheme has three levels: the adaptation level, the skill level and the learning level. Knowledge is acquired by results of adaptation to the environment in bottom up manner and instruction by human operators. The robot memorizes the knowledge at the learning level. The robot utilize the knowledge in planning its motion. This paper deals with the learning level of the autonomous robots. For complex tasks, single autonomous robot can not achieve the tasks alone because of its functional limitations. In this case, it is necessary to work cooperatively with other robots.

When many autonomous robots work together at tasks, configuration of the multi-agent system can be classified into two type. One is a centralized system at which one master robot exists and other robots obey the master. The other is a decentralized system in which each robot performs tasks cooperatively. In the centralized system, the master robot is required to have many functions. If the master robot breaks down, the task cannot be achieved. On the other hand, in the decentralized system, each robot has a function. Since plural robots work cooperatively, it is easy to deal with various tasks. Moreover, if one robot breaks down, other robots can help or replace the broken robot. Therefore, the decentralized system is superior to the centralized one. However, it is difficult to achieve the decentralized system, because interaction among robots influences unfavorably when public resources are limited.

If there is no interaction, each robot has to work optimally for its purpose, so that the total task should be achieved optimally. That is, each robot should work selfishly. Or else conflicts among the robots might occur when using a public resource. The conflicts may cause collisions and deadlock states among the robots in a local area. In order to avoid conflicts, it is necessary for the robots to communicate locally at least and to coordinate among themselves. The coordination among the robots is as important as selfishness. However, if many robots use the same public resource, the robots have to wait for their orders. In such cases, the waiting time is considered as a cost and the coordination is not always effective. Moreover, for the coordination, communication among robots is very important. However, communication itself is a burden for robots, especially global communication in a distributed robot system without a supervisor. Without global communication, it would be difficult to achieve decentralized tasks efficiently. Therefore, it is difficult for each robot to plan its effective motion while considering unpredictable coordination. This paper deals with the decentralized multi-agent system. In this system, it is difficult for each robot to communicate all the other robots globally at the same time.

This paper proposes a new strategy of motion planning for the multi-agent-robot system which has a decentralized configuration. As assumptions, every robot cannot communicate globally at the same time but can locally to avoid conflicts. In this paper, while planning, every robot uses a common static world model to express a common environment including public resources. Each robot also uses a dynamic world model to express an unknown environment including other robots, which is acquired empirically through learning. The static world model is used for selfish motion planning of the robot, while the dynamic world model is for considering coordination with other robots depending not on global communication but experiences.

For an illustration, this paper deals with path-planning of multiple mobile robots. When a mobile robot moves from a starting point to a target point, it is necessary to plan the optimal path or a feasible path for itself by avoiding obstructions in its way and by minimizing a cost which is a function of time, energy for acceleration and deceleration and distance. The major problem for path planning of a mobile robot is how to find a collision-free path from a starting point to a destination. Much research has been carried out for the path planning of a robot with stationary obstacles. Modeling of an environment of a robot is one of the key issues for path planning. Nilson proposed Visibility Graph (VGRAPH) (1969), and Lozano-Perez and Wesley applied the Visible Graph technique in a configuration space (1979). Brooks proposed a representation method of free space by means of figures called generalized cones and applied to a collision-free path planning problem (1983). Noborio et al. applied the quad tree to produce a path graph (1989). Lee and Park used a neural network to find a collision-free path for a robot (1991). Habib and Asama proposed the MAKLINK graphs which are based on a free-link concept to construct available free space within the robot's environment in terms of a free convex area (1991). The graph is less complex for searching a collision-free path than other methods, because the numbers of nodes and links in the graph are less than that in other graphs. Therefore, in this paper, the graph is used to express the known environment of the robot.

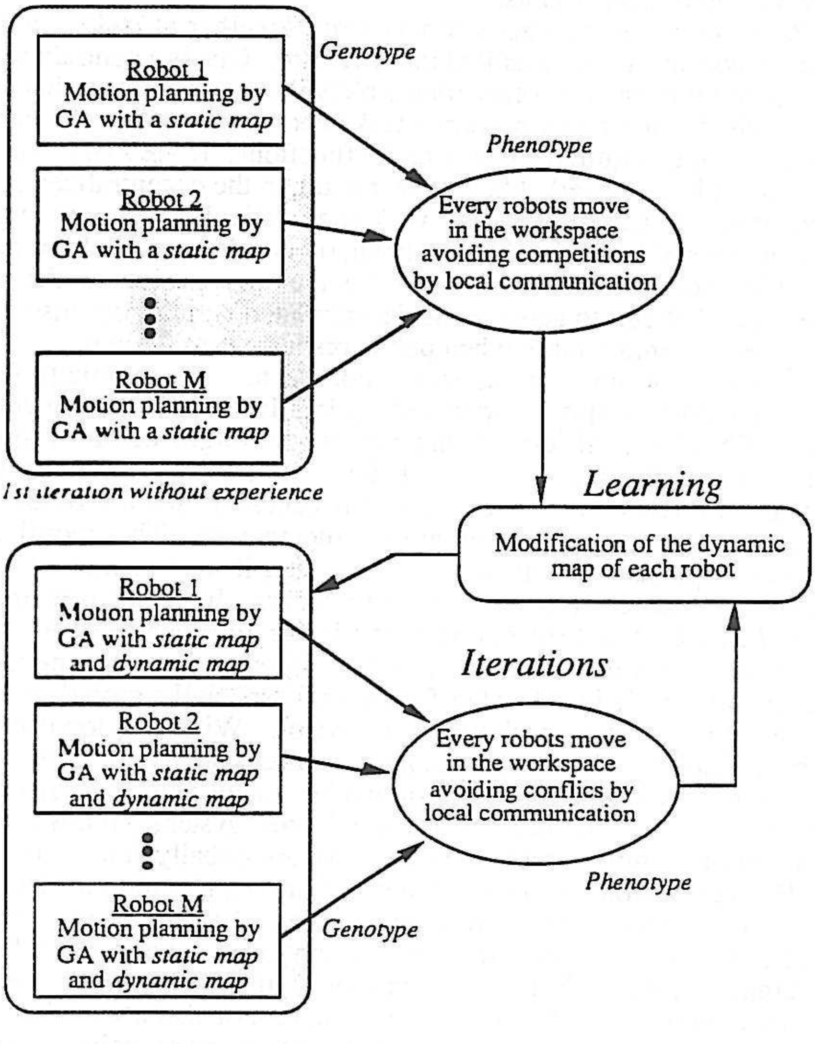

Fig. 1 Evolution in multi-agent system through learning process.
Genotype: unordered set of low-level rules
Phenotype: behaviors that emerge out of the interactions among the low-level rules when they are activated within some specific environment.

Moreover, when many robots move in the same working space, it is necessary for each robot to select the most reasonable path so as to avoid conflicts with other robots and to minimize the total cost of all robots. If each robot selects its optimal path selfishly, the conflicts may occur. Previous works use rules that the robot waits for other robots as coordination to avoid the conflicts (Wang et al., 1991, Yuta et al., 1991). They did not consider total efficiency of the system, though their method could avoid the conflicts. However, if many robots pass the same road or point frequently, the waiting time becomes excessive. Therefore, when multiple mobile robots move in the same work-space, selection of an optimal path for each robot is not always effective.

In this paper, the MAKLINK graph is used as a static map to express the known environment of the robots and the Genetic Algorithm is applied to find out the reasonable pass. The GA was introduced by Holland (1975). The GAs are inspired by adaptation in evolving natural systems (Goldberg, 1989, Davis, 1991). Pearce et al. applied the GA to local robot navigation of a mobile robot (1992) . Shibata et al. applied the GAs to motion planning of a mobile robot and coordination among multiple robots hierarchically (1992). In the previous approach, each robot plans multiple paths while using the first GA and then, a leader robot selects the most reasonable path for each robot using the second GA. There are assumptions that every robot uses only a static map of its environment and can communicate every robot at any position. Therefore, the leader robot can exist. However, in this paper, there is no leader robot as the decentralized multi-agent-robot system.

On the other hand, a dynamic map is defined to predict other robots' motions for coordination while planning motions. This dynamic map is acquired empirically through learning process as shown in Fig. 1. At first, each robot plans a path by the GA using the static map as the world model. The GA can produce many reasonable paths since the GA searches multiple points in a search space at the same time. Then, every robot moves according to the most reasonable path for each robot. The robots may wait for other robots passing by at a public place as the result of local communication to avoid a conflict. At this moment, a cost of waiting time occurs. Each robot memorizes this cost in a dynamic map as an experience while using fuzzy set proposed by Zadeh (1965). After the robots arrive at the target positions, each robot produces the dynamic map depending on time, the number of nodes, and the cost as an empirical map. In this paper, only the waiting time is considered in the dynamic map. This process is defined as one iteration in the learning process and the dynamic map is used as acquired knowledge of the dynamic environment. From the second iteration, each robot plans path by GA using the static map as a fixed world model and the dynamic map as a predicted dynamic world model depending on experience. The dynamic map is modified through iterative learning process. This learning process is a bottom up approach for primitive knowledge acquisition.

Simulations are carried out to show the effectiveness of the proposed approach. Using the proposed approach, each robot gradually acquires experience where and when the public resources are crowded through iterative learning process. Then, depending on the experience, each robot becomes able to plan reasonable path.

II LEARNING SCHEME FOR MOBILE ROBOTS

A. Hierarchical intelligent control of a mobile robot

For mobile robots, basic functions are planner, navigator, servo controller and sensory system. Hierarchical intelligent control is applied to the mobile robot. In hierarchical intelligent control, the planner corresponds to the learning level, the navigator does the skill level, and the servo controller does the adaptation level. The planner plans paths depending on aims of tasks. The navigator navigates the robot on the planned path while avoiding unexpected obstacles. This paper deals with the path planning at the planner.

99

B. Learning scheme

Figure 1 illustrates a learning scheme proposed in this paper. At first, each robot plans a path by the GA using the static map as the world model. The GA produces many reasonable paths since the GA searches multiple points in a search space at the same time. Then, every robot moves according to the most reasonable path for each robot. The robots may wait for other robots passing by at a public place as the result of local communication to avoid a conflict. At this moment, a cost of waiting time occurs. Each robot memorizes this cost in a dynamic map as an experience while using fuzzy set. After the robots arrive at the target positions, each robot produces the dynamic map depending on time, the number of nodes, and cost as an empirical map. This process is defined as one iteration in the learning process and the dynamic map is used as acquired knowledge of the dynamic environment.

From the second iteration, each robot plans the path by the GA using both the static map as a fixed world model and the dynamic map as a predicted world model depending on experience. The dynamic map is modified through iterative learning process. This learning process is a bottom up approach for primitive knowledge acquisition

C. Assumptions

The major problem for path planning of a mobile robot is how to find a collision-free path from a starting point to a destination. Much research has been carried out for the path planning of a robot with stationary obstacles. Modeling of an environment of a robot is one of the key issues for path planning. Habib and Asama proposed the MAKLINK graphs which are based on a free-link concept to construct available free space within the robot's environment in terms of a free convex area. The graph is less complex for searching a collision-free path than other methods, because the numbers of nodes and links in the graph are less than that in other graphs. Therefore, the graph is used to express the known environment of the robot with the following assumptions:

1) The polyhedral obstacles are orthogonal prisms whose heights are all parallel to Z axis, and the entire path lies entirely in a horizontal plane containing X and Y, and the given obstacles will intersect this plane in a collection of polygonal obstacles.
2) To ensure path is not too close to any of the obstacles, the dimension of the robot is represented by a point, and the boundaries of obstacles are expanded accordingly by the maximum distance at the robot's cross section plus a minimum distance required for proper sensing.

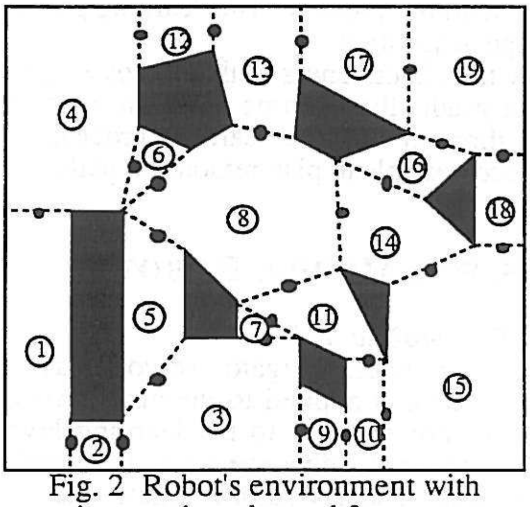

Fig. 2 Robot's environment with stationary obstacles and free-convex areas

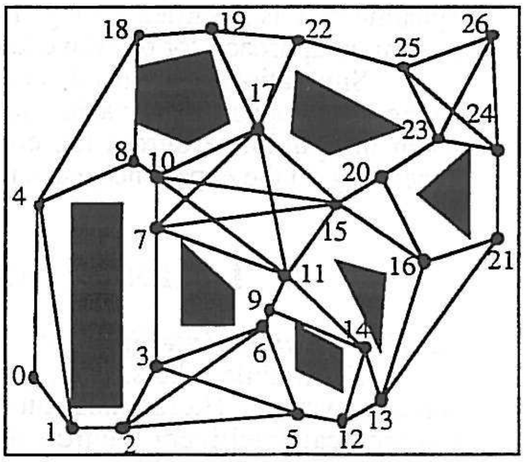

Fig. 3 A graph for a static map of the robot's environment

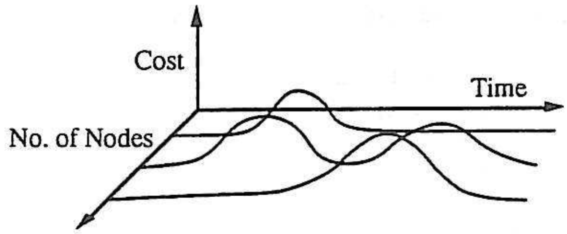

Fig. 4 Dynamic map to express robot's dynamic environment and consider experience in planning.

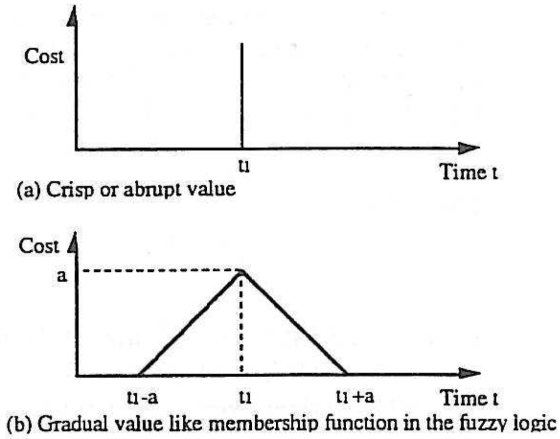

Fig. 5 Fuzzy expression of cost at a public resource for learning of the dynamic map

A simple model of the robot's environment is used. In the model, all obstacles in the environment are represented by a set of vertices to describe each polygonal obstacle. The constructed model represents the knowledge of the robot's environment in terms of a static map. This static map is given to each robot in the same space. An internal representation of the world is built through the map manager as an internal map which is suitable to be used for free space structuring and preparing the required information for path-planning.

D. Static map and rules for every robot

The MAKLINK is based on a free-link concept to construct the available free space between obstacles within an environment of mobile robots in terms of a free convex area. The free-link represents a line whose two ends are either corner of two polygonal obstacles, or one of them is a polygonal corner and other is a point on a working space boundary wall. A connection among the corners of the same obstacle is also edge between two adjacent free-convex areas. A free-link should not intersect with any edge of the obstacles within the robots' environment. The boundary of each free-convex area to be constructed will have at least two free-links as apart of its total number of edges. After calculating all possible free-links, each midpoint of each free-link is used as a node in the graph. The midpoints in the same free-convex area are connected to represent links in the graph.

In order to decide points as nodes in the graph, the free-convex areas are used as *public resources*. Figure 2 shows an example of robots' environment with some stationary obstacles, in which the free-convex areas with free-links are determined and each free-convex area is numbered. Figure 3 shows a graph in which each node is numbered and length of each link is used to calculate fitness of path in Genetic Algorithms (GAs). Rules to avoid conflicts are defined as follows:

3) Only one robot can enter in a free-convex area. Other robots which would like to enter the same free-convex area have to wait at the point on the edge of the area.
4) Plural robots can wait at the same point on the edge of the free-convex area.
5) In order to avoid deadlock states, if some robots arrive at the same free-convex area, the robot which exists the smallest number of nodes has priority to enter the area.

E. Dynamic map for each robot

When a robot waits for other robots passing by a free-convex area, waiting time on time t is considered as a cost for the robot. The robot regards the cost at the free-convex area as an experience. The experience is memorized in a dynamic map while using fuzzy expressions in this paper. Figure 4 shows the dynamic map to express world models and for the robot to consider experiences in planning. Memorized data is the amount of the cost with regard to waiting time, when the robot waited, and at which free-convex area. If a crisp value is used for expressing the cost at a time, the cost changes drastically in the dynamic map (Fig. 5 (a)). For example, the robot waits from time t_1 to t_1+a as in Fig. 5 (a), the cost is considered from t_1-a to t_1+a as in Fig. 5 (b).

$$Cn_{new}(t) = \begin{cases} 0 & t < t_1 - a \\ t - t_1 + a & t_1 - a \leq t < t_1 \\ -t + t_1 + a & t_1 \leq t < t_1 + a \\ 0 & t_1 + a \leq t \end{cases} \qquad (1)$$

Cn is the cost function of time at the node n.
New data from t_1-a to t_1+a is added to the dynamic map multiplied with a learning rate α, while old data from t_1-a to t_1+a is forgotten with an oblivion rate β as follows:

$$Cn(t) = \alpha Cn_{new}(t) + \beta Cn_{old}(t) \qquad t_1 - a \leq t \leq t_1 + a \qquad (2)$$

Through this process, the dynamic map is modified and the experience is accumulated as the learning. The next chapter describes the way of using the MAKLINK graph in the GAs.

III MOTION PLANNING BY GENETIC ALGORITHM

Genetic Algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human problem solving. An occasional new part is tried for good measure avoiding local minima. While randomized, GAs are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance.

In the first step, each robot plans its optimal paths for moving selfishly. At this time, each robot uses the GA with the static map of the known environment which is expressed by using nodes and links in the graph.(Fig. 3). Then, each robot moves along the planned path. At some free-convex areas, the robot may wait for other robots passing by as the result of local coordination and memorizes the cost of waiting time as experiences. For the robot, the other robots are considered as factors of the unknown environment. This process is regarded as the first iteration. After the first iteration, each robot plans paths while using the static map and the dynamic map to predict local coordination in the dynamic environment. Through iterations, each robot acquires approximate knowledge about how and when public resources are often used. Behavior of the system is evaluated by the total cost with regard to waiting time and distances resulting from detours.

A. Basis of genetic algorithms

GAs have traditionally three operations to abstract and rigorously explain the adaptive process of natural systems as follows:

(1) Selection operation,
(2) Crossover operation,
(3) Mutation operation.

The selection process is an operation to select the survival in a set of candidate strings. In this process, the fitness value is calculated for each candidate string by using a fitness function which depends on a goal for searching problems. According to the fitness value, the selection rate is determined for the present candidate strings, and the survival is selected in any rate depending on the selection rate.

The crossover process is a reform operation for the survival candidates. In natural system, a set of creatures creates a new set of the next generation by crossing among the creatures. In the same way, the crossover process is performed by exchanging pieces of strings using information of old strings. The pieces are crossed in couples of strings selected randomly. The mutation process is held to escape the local minima in search space in the artificial genetic approach. The calculation is stopped when the generation is up.

B. Coding method of path for string

Genetic Algorithms for the path-planning of each robot is referred to as GA1. In the GA1, a path for a mobile robot is encoded based on the order of via points. Each robot has a starting point and a target point in the graph under the assumption that each robot passes each point only once or not at all. In Fig. 3, each node has a number. The nodes in the graph are used to encode a path as a string which is expressed by the order of the numbers, for example, *'0-1-2-6-9-11-15-17-22-25-26.'* In this string, '0' is the start point and '26' is the target point for a mobile robot. These points are selected randomly at first, while adjacent numbers must be connected with a link in the graph. Since order-based strings are used, specialized operations of crossover and mutation are used.

C. Calculation of fitness using static map and dynamic map

Distance of the path is indicated by a string used for the *fitness* of each string. Therefore, as the distance decreases, the fitness increases. To calculate the fitness, the length of the link between numbers in the string and the estimated cost at the public resources are calculated as followings:

String: 0-1-2-6-9-11-15-17-22-25-26
Fitness in GA1:

$$F_{1}(k) = d_{0-1} + d_{1-2} + d_{2-6} + \cdots + d_{25-26} + C_1 + C_2 + C_3 + \cdots + C_{19} \quad (3)$$

where $F_{1}(k)$ is the fitness value of the *k*th string and d_{i-j} is distance between nodes *i* and *j*, which is equal to length of the link in the static map. The notation C_n is the estimated cost at *n*th public resource expressed in the dynamic map. Since the C_n depends on time, the search space is huge and the conventional graph search techniques cannot be simply applied to this search problem. However, the proposed approach can be easily applied while considering the cost C_n in the scalar fitness function. Since the GA searches multiple points in the search space at the same time, the GA can obtain the optimal solution and feasible solutions. The robot may use the optimal path for itself but other paths are used in the next iteration, because the dynamic map would have changed through iteration and the fitness of the same string would vary with the dynamic map.

To select strings, following equation is defined.

$$f_1(t) > \frac{\sum_{i=1}^{N} f_1(t)}{N} \quad (4)$$

N is the total number of strings.

D. Crossover and mutation

In the crossover, a string which has an efficient fitness value is randomly selected as a parent, and a number in the string is also randomly selected at first. Then, one more string which also has an efficient fitness value is selected as a second parent at random. If the second parent has the same number selected in the first parent, each string exchanges the part of strings after that number. If not, another string is selected as the second parent, and the same operation is repeated. The followings are examples:

Parent 1: 0-1-2-6-9-11-<u>**15**-17-22-25-26</u>
Parent 2: 0-4-8-10-<u>**15**-20-23-26</u> (5)

Each parent has the point '15' and exchanges the part of each string,

Child 1: 0-1-2-6-9-11-**15**-20-23-26
Child 2: 0-4-8-10-**15**-17-22-25-26 (6)

After the crossover, the children are checked whether each string has same numbers. If so, a part of the string between the same numbers is cut off as follows:

Child 3: 0-1-2-6-9-**11**-15-10-7-**11**-17-22-25-26 (7)

after the cut off operation

Child 3': 0-1-2-6-9-**11**-17-22-25-26 (8)

In the mutation, a position in each string is selected at random based on the probability of mutation which is low. Then, the number of nodes is selected randomly for following positions which are connected sequentially.

IV SIMULATION RESULTS

A. *Selfish planning by Genetic Algorithm with a static map*

The GA1 is applied to the planning of collision-free paths of a single mobile robot as selfish-planning in a space with stationary obstacles. The simulation was carried out to show the effectiveness of the proposed approach. As an assumption, there was one mobile robot in the work space. Therefore, conflicts can not happen. The search space of the graph was very large. In the simulation, the starting point is '0' and the goal point is '26' in Fig. 3 and the size of the search space is about 620,000. The GA1 used 50 strings for the population, and calculated 100 generations with 80% probability of crossover and 20% probability of mutation for each string. 50 cases of random parameters were investigated. The GA1 succeeded in obtaining the optimal path at the rate of 22% and the feasible path of the rest. Figure 6 shows the relationship of the best fitness of the string and generation. The robot succeeded in finding out the optimal path and feasible paths for itself while using the static map.

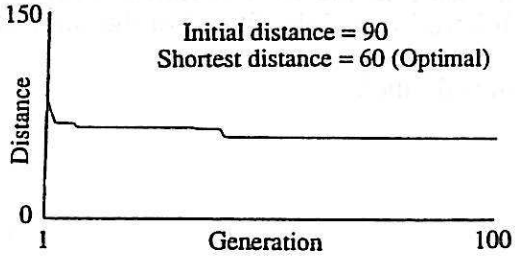

Fig. 6 Simulation results of selfish-planning by using GA1.

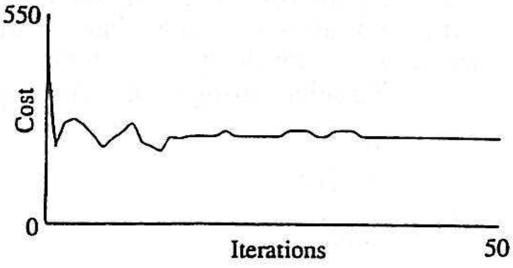

Fig. 7 The relationship between iteration and the total cost of waiting time and distance resulting from detours.

B. *Evolution and coordinative behavior of multiple robots*

The simulation was carried out with 10 robots in the work space, and all of the robots could not communicate with each other globally at the same time but could locally. At first, each robot planned its path selfishly only using the static map and then moved along with their selfish path. During the motion, conflicts occurred at the free convex areas and the robots waited for other robots to pass by according to the rule of local coordination. In order to evaluate system behavior, costs of robots are summed as total cost of the system. However, this total cost is not used for each robot. Figure 7 shows the relationship between the total cost of the system and iterations. At the first iteration, the total cost with regard to waiting time was excessive as the result. Each robot memorized the cost with information of how, when, and at which free convex area in the dynamic map the cost occurred, while using equations (1) and (2).

In the second iteration, each robot planned its path by the GA1 using the static map and the created dynamic map. In the second iteration, the strings were not generated at random as in the first generation, but the strings which were created in the last generation, at the previous iteration, were used as the first generation of the second iteration. Because the strings already had efficient fitness values with respect to the distance of the paths represented in the static map. The fitness values of the strings vary according to the each dynamic map. As a result of the second iteration, the total cost of all the robots was reduced. This is because some robots select detour with respect to length to avoid passing crowded area.

Moreover, the multiple robot system iterated the process, while each robot learned the dynamic environment. After several decades of iterations, the total cost was reduced, although global communication was not used. This is because each robot considers interactions with other robots while planning, using the dynamic map. Table 1 shows the obtained path. The proposed strategy succeeded in decreasing the total cost without global communication. If there were a supervisor, the paths of the least cost in the iteration would be used for moving. Though the supervisor was not exist, the robots behave efficiently after iterations because of its learning ability.

The result from the proposed approach is compared with results from two situations. One is that every robot selects the shortest path for itself as selfish planning. The other is that every robot can communicate each other before moving and there is a leader robot. Shibata et al. have shown hierarchical path planning approach while using two kind of GA. In the previous approach, each robot plans multiple paths while using the first GA and then, a leader robot selects the most reasonable path for each robot using the second GA. These planning are performed once before moving (off-line planning). Each cost is normalized by the cost resulting from selfish planning. Table 2 shows the comparison of the results. Though the proposed approach in this paper does not use global communication, the result is quite reasonable. Therefore, the leaning function is efficient for the autonomous robot to plan its motion in a dynamic environment.

Table 1 Planned paths of ten robots after 50 iterations

Robot	Start	Goal	Optimal Path for itself	Planned Path
1	0	26	0-1-2-6-9-11-15-20-23-26	0-1-2-6-9-11-15-20-23-26
2	26	0	26-23-20-15-11-9-6-2-1-0	26-24-21-13-12-5-2-1-0
3	22	12	22-17-111-14-12	22-25-23-20-16-13-12
4	12	22	12-14-11-17-22	12-13-16-20-23-25-22
5	5	19	5-6-9-11-17-19	5-6-9-11-17-19
6	19	5	19-17-11-9-6-5	19-17-11-9-6-5
7	4	21	4-8-10-15-16-21	4-8-18-19-22-25-23-24-21
8	21	4	21-16-15-10-8-4	21-24-23-25-22-19-18-8-4
9	18	13	18-8-10-11-14-13	18-19-17-11-14-13
10	13	18	13-14-11-10-8-18	13-16-15-17-19-18

Table 2 Comparison of results of the total cost of the multi-agent-robot system

No.	Situation of the system	result
1	All robots move selfishly	1.00
2	All robots can communicate	0.39
3	Coordinative behavior through learning (Robots communicate locally to solve conflicts)	0.36

V CONCLUSIONS

This paper proposed a new strategy for the coordinative motion planning for multi-agent systems. The robot in the system uses the Genetic Algorithm (GA) to search the optimal solution and feasible solutions of motion. While planning, each robot uses the static map of the known environment and the dynamic map of the dynamic environment. The dynamic map is modified by experiences through learning. The experiences are expressed by the fuzzy set in the dynamic map. Though the robots only communicate in a local area for coordination in order to avoid conflicts, the behavior of the system becomes efficient evolutionary through iterative learning.

REFERENCES

R. A. Brooks, Solving the Find Path Problem by Good Representation of Free Space, IEEE Trans. on Sys., Man, and Cyb., Vol. 13, No. 3, pp. 190-197 (1983)

L. Davis (ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold (1991)

D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Welsey (1989)

M. K. Habib and H. Asama, Efficient Method to Generate Collision Free Paths for Autonomous Mobile Robot Based on New Free Space Structuring Approach, Proc. of IROS '91, Vol. 2, pp. 563-567 (1991)

J. H. Holland, Adaptation in Natural and Artificial Systems, An Arbor: University of Michigan Press (1975)

S. Lee and J. Park, Cellular Robotic Collision-Free Path Planning, IEEE Fifth Int'l Conf. on Advanced Robotics, pp. 539-544 (1991)

T. Lozano-Perez and M. A. Wesley, An algorithm for Planning Collision Free Paths among Obstacles, Communication ACM, Vol. 22, pp. 560- 570 (1979)

N. J. Nilson, A Mobile Automation; an Application of Artificial Intelligence Techniques, Proc. of IJCAI, pp. 509-520 (1969)

H. Noborio, T. Naniwa, and S. Arimoto, A Fast Path-Planning Algorithm for a Mobile Robot Based on a Quadtree Representation, Jour. of the Robotics Society of Japan, pp. 3-12 (1989)

M. Pearce, R. Arkin and A. Ram, The Learning of Reactive Control Parameters Through Genetic Algorithms, Proc. of the IROS '92 vol. 1, pp. 130-137 (1992)

T. Shibata, T. Fukuda, et al., Skill Based Control by using Fuzzy Neural Network for Hierarchical Intelligent Control, Proc. of IJCNN '92 - Baltimore, vol. 2, pp. 81-86 (1992)

T. Shibata, T. Fukuda, et al., Selfish and Coordinative Planning for Multiple Mobile Robots by Genetic Algorithm, Proc. of the 31st IEEE Conf. on Decision and Control, Tucson, vol. 3, pp. 2686-2691 (1992)

T. Shibata and T. Fukuda, Hierarchical Intelligent Control for Robotic Motion, Trans. on Neural Networks (to appear, 1993)

J. Wang, Fully Distributed Traffic Control Strategies for Many-AGV Systems, IROS '91, pp. 1191-1204 (1991)

S. Yuta, and S. Premvuti, Consideration on Cooperation of Multiple Autonomous Mobile Robots - Introduction to Modest Cooperation - , Proc. 1991 IEEE International Conference on Advanced Robotics, pp. 545-549, (1991)

L. A. Zadeh, Fuzzy Sets, Information and Control, Vol. 8, pp. 228, (1965)