www.s2is.org
International Journal on Smart Sensing and Intelligent Systems
ISSN 1178-5608

BOTTOM-UP APPROACH FOR BEHAVIOR ACQUISITION OF AGENTS EQUIPPED WITH MULTI-SENSORS

Naoto Hoshikawa[1], Masahiro Ohka[1], and Hanafiah Bin Yussof[2]

[1]Graduate School of Information Science, Nagoya University

Email: ohka@is.nagoya-u.ac.jp

[2]Faculty of Mechanical Engineering, Universiti Teknologi MARA

Submitted: Oct. 23, 2011 ***Accepted: Nov. 10, 2011*** ***Published: Dec. 1, 2011***

Abstract – While the top-down approach of artificial intelligence encounters the frame problem, the bottom-up approach based on a creature's evolution and behavior is effective for robotic design of intellectual behavior in a specific field. We propose the Evolutionary Behavior Table System (EBTS) using a simple genetic algorithm (SGA) to acquire the autonomous cooperative behavior of multi-agents as the bottom-up approach. In EBTS, a set of rules is expressed as a table composed of sensor input columns and actuator output columns; a row of the table corresponds to a rule. Since each rule is transformed to a string of Boolean values, we treat a long string composed of actuator output strings in the rules as a gene to obtain an optimum gene that adapts to the environment using SGA. In computational experiments, the collective robots could convey an object to a goal through cooperative work; the multi-fingered hands grasped the object and transferred it to the goal. Final truth tables obtained by the gene data do not always assure an optimum solution, but the calculation cost is reduced from astronomical figures to around one ten to twenty thousandth. If we use the top-down methodology, astronomical trials are needed to specify the optimum pattern. Therefore, EBTS is an attractive method

because it is very useful for obtaining general robotic behaviors in both collective and multi-fingered hand tasks.

Index terms*:* Multi-agent, Bottom-up approach, Collective task, Multi-fingered hand, Evolution, Behavior table, Genetic algorithm.

I. INTRODUCTION

Robots are progressing rapidly with recent developments in several technologies related to components and materials, such as sensors, actuators, computers and batteries. These technologies are used for industrial robots, home robots, rescue robots and guard robots. Examples of these include the cleaning robot Roomba [1], which possesses several sensors to detect obstacles and avoid collision; a robot for protection against disasters called PackBot [2], which is equipped with several sensors and long-life batteries and is committed to nuclear and radiation accidents in Japan; and a guard robot called D1 [3], which was developed by ALSOK for patrols and is equipped with several sensors and batteries as well. In addition to these robots, there is a healing robot called PARO [4] equipped with tactile sensors, auditory sensors, light sensors, temperature sensors and posture sensors to respond to persons' speaking and touching. It is reported that PARO is effective in protecting against the advancement of Alzheimer's disease, and in treating depressive psychosis and autism [5]. As mentioned, not only the number and kind of sensors and actuators but also operation time are increased with the progress of components and material technologies.

However, there are innumerable problems to solve in intellectual behavior that is automatically generated based on information acquired by several kinds of sensors mounted on the robot. While robotic hardware for realization of intellectual behavior has progressed, the level of intelligence design is not yet satisfactory. In particular, manual control for each robot is not acceptable for a cooperative task of multiple robots. It is inevitable that design of intellectual behavior should be established to utilize the advantage of multiple robots.

In this paper, we recognize the present problems with the top-down architecture of robotic intellectual behavior to establish intelligence design for multiple robots and a multi-fingered hand. We will discuss why bottom-up architecture is effective for solving the problems based on investigation of intelligence design. Then, we will introduce the Evolutional Behavior Table

System (EBTS) [6][7] proposed by us as one of the effective design methods for intellectual behavior. In this method, after patterns of actuator output for specific sensor inputs are defined as gene information, the actuator output is optimized using a genetic algorithm (GA) to make the robot generate appropriate behavior adapted to its environment.

In our first report, in which EBTS is introduced to achieve artificial tactile affordance (ATAS), we concluded that EBTS is effective for the object transfer problem of multiple robots, but it is not always adequate for manipulation of multi-finger hands [6]. However, in the subsequent report, computer experimental results showed that EBTS was applied to the manipulation problem in a specific workspace [7]. In this paper, we investigate the effectiveness of EBTS for a collective task and multi-fingered hand task according to reexamination of the results of previous papers.

II. TOP-DOWN AND BOTTOM-UP ARCHITECTURES

a. Problems with Classic Artificial Intelligence

Artificial intelligence (AI) was named by J. McCarthy in 1956 and is defined as a technology intending to achieve human intelligence with data processing in computers [8]. Since processing of symbols was a major technique for modeling human intelligence in the early stages of classical AI, there are problems related to the relationship between the symbol and the real world, and discrimination between issues required for intellectual behavior and other issues.

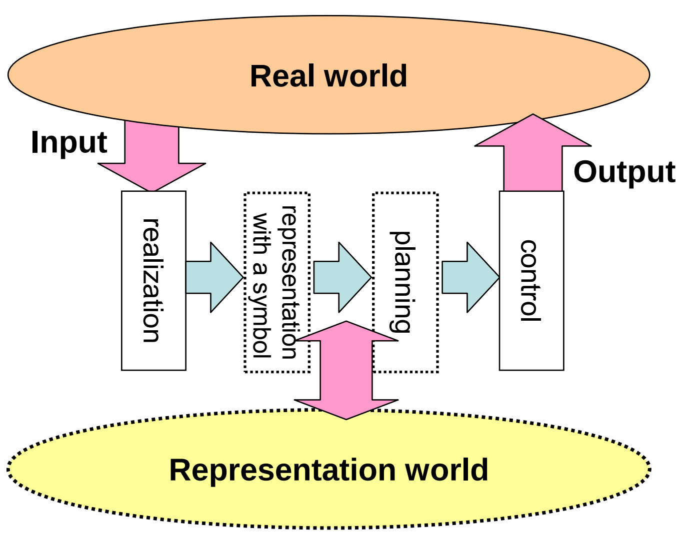


Figure 1. Information processing based on symbolism

In the early stages of AI research, the data processing performed by a von Neumann-type computer that executes in turn a program stored in data storage is applied to the intelligence design. Namely, the intelligence is achieved with symbolizing of the real world and calculating the symbolized world. This idea in AI is called *symbolism* (Fig. 1) [9].

Although people generally assume that if processing speed and memory are increased in the computer, advanced AI is achieved, difficult problems that are not solved by the von Neumann-type computer remain. There are two problems in the symbolizing of the real world. One of them is limitless enlargement and complexity of relationships between concept and symbolization: if the lower level concept is required to explain details of a specific concept, the explanation of the lower level concept recalls a still lower level concept in turn. The other problem is how to connect symbolized information in a system to the real world: this is called the *symbol grounding problem* because a specific symbol needs to be grounded in the real world [10].

The limitless enlargement and complexity of the relationship between concept and symbolization is inevitable to transform any information required for intellectual behavior to frame the form of proposition. Although issues directly related to the intellectual behavior should be distinguished from other issues, all sorts of possibilities should be taken into account. Since this problem was first pointed out by J. McCarthy et al. in 1969, it has been one of the most difficult problems in AI and is called the *frame problem* [11][12].

In present AI, the frame problem is avoided by assuming precise conditions in finite space to restrict the considering environment. Since there are several programs for computer chess [13] and Shogi as representative examples and some of them reach the professional level, we can say that AI superior to human beings has already been achieved in a restricted environment.

Indeed, human beings cannot completely solve this problem either. However, creatures such as human beings can select their behavior to match the restricted environment through naturally removing information unrelated to the behavior. On the other hand, a lot of axioms for solving the frame problem are required in classic AI even if AI judges an obvious truth.

b. Relationship Between Environmental Information and Intelligence

The top-down approach pursues the strict solution as a result of assumption of frames. However, we do not intend to obtain the strict solution through spending a long time, but we intend to respond to a specific issue with appropriate behavior without missing the timing through finding a reasonable solution within the actual time. The bottom-up approach is introduced to accomplish the above appropriate behavior. In this approach, interaction between the real world and an individual is simulated to obtain intelligence through considering a field for the individual and abilities of its sensation and behavior.

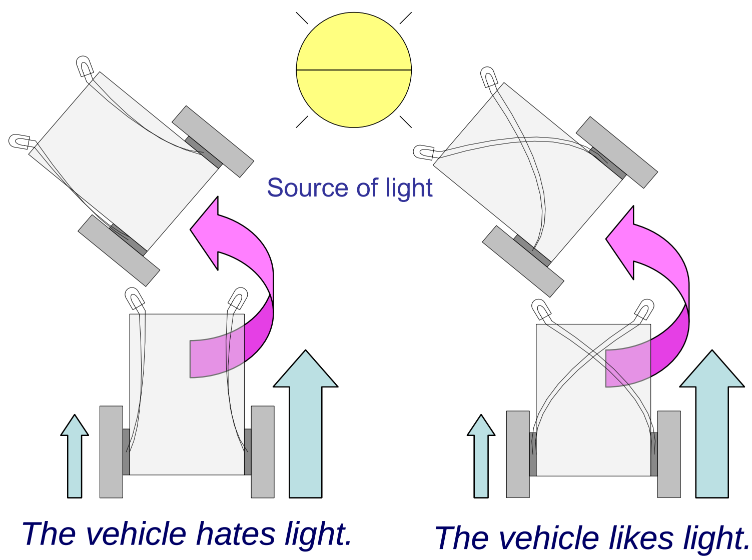


Figure 2. Photophilic and photophobic vehicle

First, a thought experiment performed by Braitenberg (1984) is introduced as an example [14] that explains acquisition of intellectual behavior through interaction between the real world and the individual. Braitenberg's vehicles (Fig. 2) possess two optical sensors with outputting signal being proportional to brightness and two wheels rotating individually with torque being proportional to input signal. These sensors are connected to wheels with simple wiring to drive the vehicle. As shown in Fig. 2, if connection between the sensor and the wheel is parallel, the vehicle goes away from the light source because the wheel on the light side rotates faster. If the connection is crossed, the vehicle approaches the light source because the wheels rotate oppositely.

The behavior of approaching/avoiding light is not accomplished by symbolization such as “if the vehicle accepts a specific directional light in the assumed field and lighting environment, it turns to the opposite direction of the light source”. It sends only the information accepted by the sensor to the specific circuit to output the behavior on the effector. However, an observer seems to believe that it is like a photophilic/photophobic creature. Braitenberg insists that creatures' various behaviors are produced not through the analysis of complex principles but by simple rules, and that these behaviors seem to be achieved by simple principles because they look complex.

R. Brooks emphasizes that the way of symbolizing the real world as representation causes several problems, and introduces subsumption architecture (SSA) on the basis of behavior of social insects such as bees and ants [15][16]. Creatures have acquired their present biological functions through evolution. They do not acquire intrinsically sensory organs such as vision and tactile organs and effectors such as uppers and lower limbs but their primitive sensory organs have progressed to obtain the present high-level functions through the long period of evolution. Brooks takes into account this situation to accomplish SSA, in which basic functions are accumulated to form high-level organs, and the old structure is wrapped (subsumed) with a new function in turn to acquire the intellectual behavior through the bottom-up approach.

Since the real world is recognized as representation in SSA, i.e. SSA does not have any recognition system, which is a common term in cognitive science, software designers do not worry about the frame problem because there is no process for symbolization and modeling. Brooks describes how the complexity of system behavior does not depend on the complexity of the creature (robot), and that human beings do not consider the real world as representation. According to SSA, he produced an insect-type robot called *Genghis* [17], which can show various behaviors similar to an insect or small animal because Genghis rapidly responds to complex environment geometry through switching reflex movements for variation in the environment.

Although SSA is derived from the behavior of social insects, a similar idea is derived from the behavior of animals. Psychologist J. J. Gibson introduced a concept called *affordance* [18] for study of an animal's complex behavior depending on the environment. Affordance is defined as a quality of an object, or an environment, that allows animals the possibility to perform an action. In SSA introduced by Brooks, since creatures do not need intelligence to perform intellectual behavior, which depends on complexity of the environment, SSA is closely related to affordance. In recent research, application of ethology, which treats animal behavior, has progressed in robotics. Study of interaction between animals and human beings is applied to enhancement of robotic intellectual behavior [19].

As mentioned, SSA, affordance and ethology treat similar concepts, in which intellectual behavior is obtained through interaction between creature and environment based on bottom-up architecture. EBTS explained in the subsequent chapter is produced based on this bottom-up architecture as one of the achieved artificial affordance systems [6].

c. Methodology of Intellectual Behavior Design Based on Bottom-up Approach

According to the discussion in the preceding section, the bottom-up approach that is established through discussion of a creature's evolution and behavior is effective for robotic design of intellectual behavior. Although there are several methods in the bottom-up approach, methodology based on evolution is effective because of obtaining a useful solution. In methodology based on evolution, data structures and equations are modeled through mimicking the mechanism of creature evolution. For methodology based on evolution, there are genetic algorithm (GA), evolution strategy (ES), evolutionary programming (EP) and genetic programming (GP).

Since we adopt simple GA (SGA) [20] produced by D. Goldberg as the methodology of evolution in EBTS in this paper, the procedure of GA is summarized in this section. In GA, gene structure possessed by an individual is assumed as the optimum solution, and calculation of optimization is performed through mimicking the method of evolution in biology. To use GA as the optimum solution, three modeling designs are required as follows.

(1) *Genetic representation of solution*

Since one individual possessing a specific genetic feature corresponds to a solution in GA, it is important how to represent the solution as an individual. Information of biological genes is represented through combinations of four bases, adenine, cytosine, guanine and thymine. This combination and the individual expressed by the combination are called genotype and phenotype, respectively. In GA, the determination variable in the optimum solution is represented as a vector $\mathbf{x}$ including N elements x_i ($i = 1, 2, \cdots, N$). The solution is represented as the phenotype composed of genotypes. The value of x_i corresponds to the base of the biological gene and the place storing the value is called the locus.

(2) *Evaluation of solution*

In biological natural selection, the concept of *fitness* is defined as the superiority or inferiority of the individual in the living environment. If the specific individual adapts itself well to environment, the value of fitness takes a high value to obtain high survival probability. In GA, the individual possessing higher quality is provided a higher value of fitness. If GA is applied to the optimization problem, the candidate of solution $\mathbf{x}$ and object function $f(\mathbf{x})$ correspond to individual and fitness g, respectively. The relationship between object function $f(\mathbf{x})$ and fitness g is determined to satisfy the following condition: if the object function of $\mathbf{x}$ takes a higher value, the individual corresponding to $\mathbf{x}$ takes higher fitness g.

(3) *Search of solution*

In GA, a group of individuals is defined, and search of solution is performed within the group. Operation for the group of individuals is called the *genetic operator*, and three kinds of genetic operators are used as follows.

Selection

Superior individuals are reserved to make genes of individuals possessing higher fitness spread in the group of individuals, while inferior individuals are reduced. More than one strategy is usually used to perform this selection. One of the strategies is *roulette wheel selection*. If g_i is assumed as fitness of individual i in generation t, the individual's probability of being selected p_i is calculated as follows:

$$p_i = \frac{g_i}{\sum_{k=1}^{N_p} g_k} \qquad (1)$$

where N_p is total number of individuals in the population.

The other selection is *elitist preserve strategy*, which preserves excellent individuals recoding high fitness toward the next generation. This strategy is used with another strategy.

Crossover

Although there are several crossover techniques, three major techniques, *one point crossover*, *multiple point crossover*, and *uniform crossover* are described as follows.

The *one point crossover* is applied under crossover rate P_c after pairs composed of individuals are selected from the population. After that, a single crossover point is selected on both parents' gene strings. All data beyond that point in either gene string are swapped between the two parent genes.

In the *multiple point crossover*, multiple points are randomly selected on the parent gene strings under crossover rate P_c after pairs of individuals are selected from the population. Everything between two points of every pair is swapped between the parent genes to render two child genes.
In the *uniform crossover*, each bit in the parent strings randomly selected from the population is exchanged with a probability of 0.5.

Mutation

Mutation is used to maintain genetic diversity from one generation of a population to the next generation. For each individual, the value of a gene on the specific locus selected randomly is reversed with mutation probability P_m. The mutation probability should be set low because mutation sometimes destroys a good gene to introduce low average fitness.
Figure 3 shows the state transition diagram of the present system. In EBTS described in the next chapter, elitism, roulette wheel selection, uniform crossover and mutation are used.

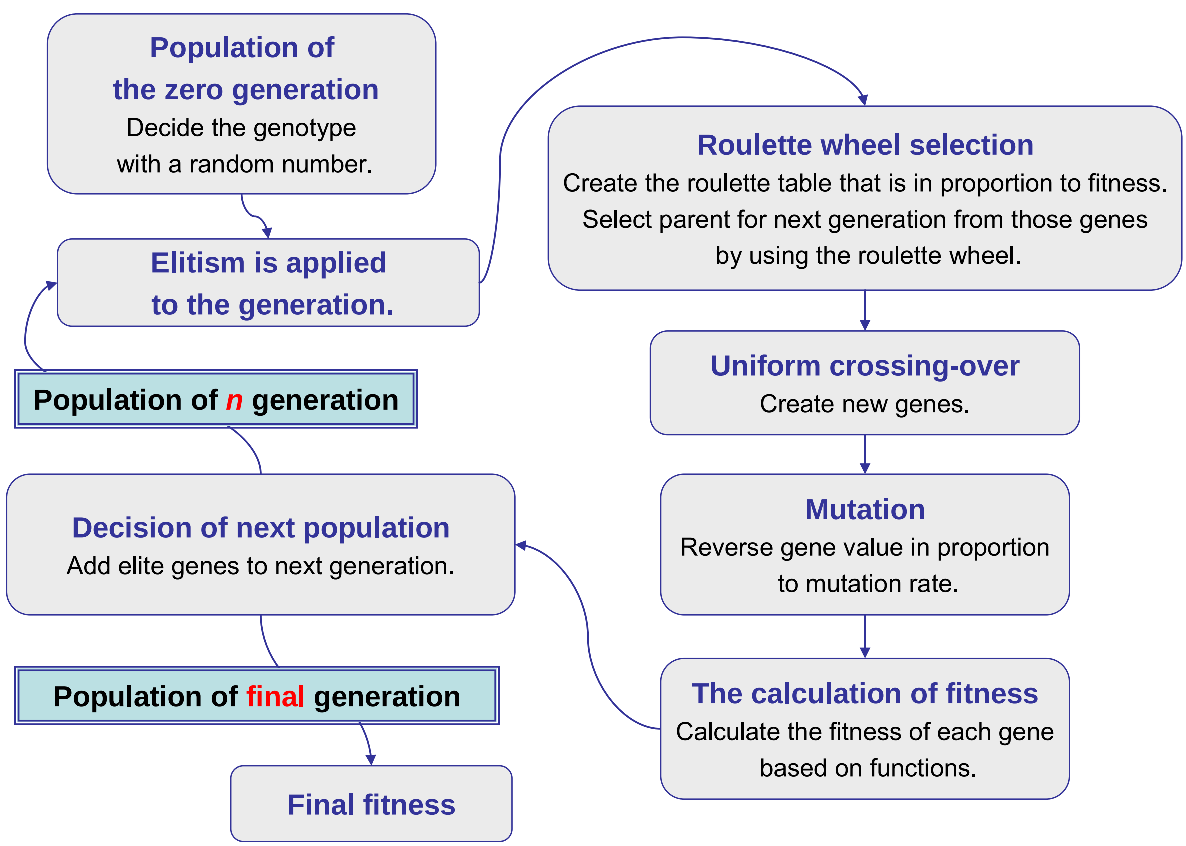


Figure 3. State transition diagram

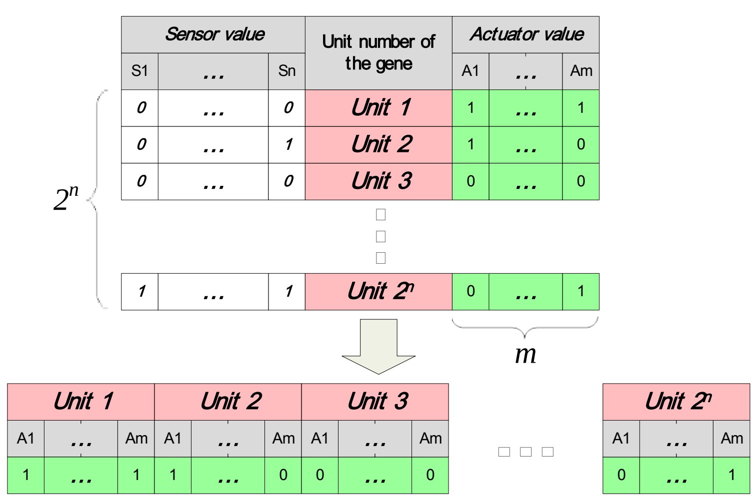
Table 1 Evolutional behavior table

Num	Sensor value					Actuator value			
	S1		…		Sn	A1	…		Am
1	0	0	0	0	0	1	0	1	1
2	0	0	0	0	1	1	0	0	0
3	0	0	0	1	0	0	0	0	0
□ □ □			□ □ □				□ □ □		□ □ □
2^S	1	1	1	1	1	0	0	0	1

III. EVOLUTIONAL BEHAVIOR TABLE SYSTEM

a. Behavior Table

We must define the relationship between environmental information and the behavior rules in a single-layered reflection behavior system to control the behavior of a robot. For this purpose, we introduce a behavior table composed of sensor and actuator status as shown in Table 1. S_1, S_2, …, S_n show the status of sensors mounted on the robot; A_1, A_2, …, A_m show the status of the actuators. Since the sensor and actuator statuses are described by Boolean values (1 or 0), the number of total patterns of sensor status is 2^n, where n is the number of sensor columns.



Length of gene locus : $2^n \times m$

Figure 4. Genotype model

b. Behavior Acquisition Method

Lines of actuator in the behavior table can be expressed with a one-dimensional array like genes because all of the truth-values are Boolean data. Therefore, the behavior table can be designed as a model of SGA comprised of Boolean data. Since the behavior table is evolvable, we call it the Evolutionary Behavior Table System (EBTS). The design procedure of the behavior table using SGA is shown in the following.

First, we design a genotype array that has information about the behavior table. Since the genotype array is formed as shown in Fig. 4, the length of the genotype array is shown with the next formula:

$$G = 2^{S} \times m \qquad (2),$$

where the output gradation is m.

The agent's fitness value is calculated in a simulator that is equipped with internal functions that evaluate the efficiency and the task accuracy degrees of the robot. Then, the simulator generates a behavior table from the genotype array of a one-dimensional vector composed of G elements (Fig. 4). The robot behavior is evaluated on the basis of the task achievement degree in the simulator field during a specified period. The evaluation value obtained by this simulation is sent to the calculating system for genetic algorithms as fitness.

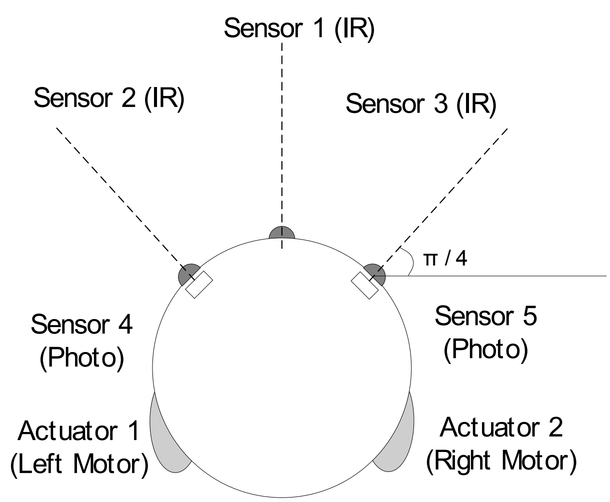


Figure 5. Model of circular mobile robot

IV. MODELING FOR COLLECTION ROBOT AND MULTI-FINGERED HAND

a. Actual Behavior Tables

To examine EBTS's effectiveness for manipulation of multi-fingers hands, we compare the fitness obtained from the multi-fingered hand task with that obtained from the collective task.

(1) *Collective task*

In the task, a circular mobile robot [21]-[25] equipped with infrared (IR) and photo sensors transports an object-emitting radiation light to a goal-emitting radiation light, as shown in Fig. 5. Photo sensors are used for searching for the goal; both IR and photo sensors are used for searching for the object. Each speed of the robot's two wheels is expressed by 2-bit data. "00", "01", "10" and "11" correspond to "stop", "slow", "medium" and "fast", respectively. Since it has 5 sensors and 4-bit output gradation,

$$n = 5,\ m = 4 \qquad (3)$$

If these values are substituted into Eq. (2), the number of elements of the genotype array G is calculated as follows:

$$G = 2^5 \times 4 = 128 \qquad (4)$$

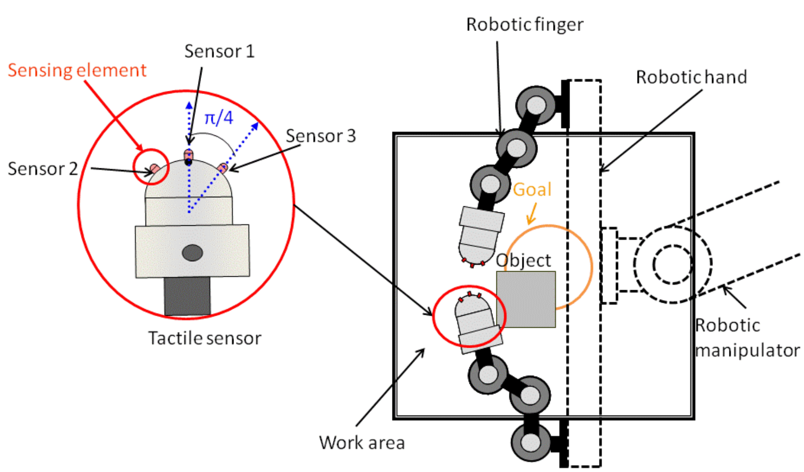


Figure 6. Model of robotic finger equipped with tactile sensors

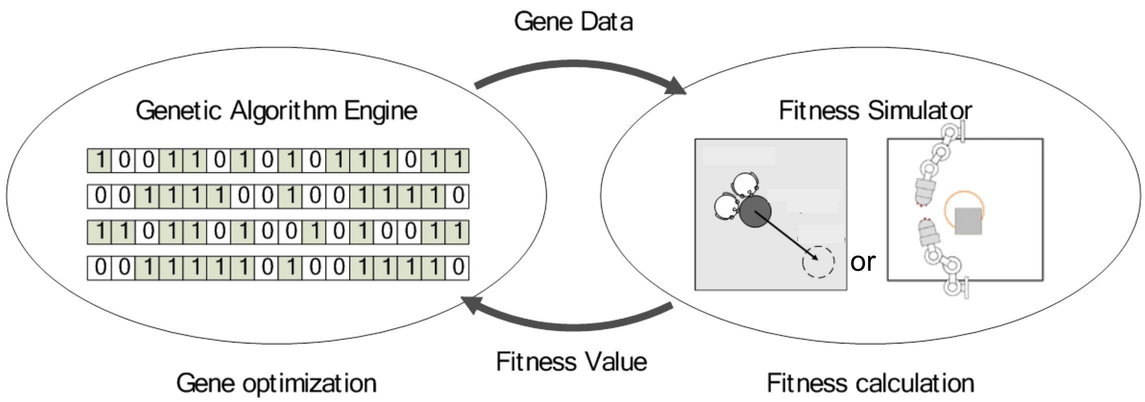


Figure 7. Evolutionary Behavior Table System (EBTS)

(2) *Multi-fingered hand*

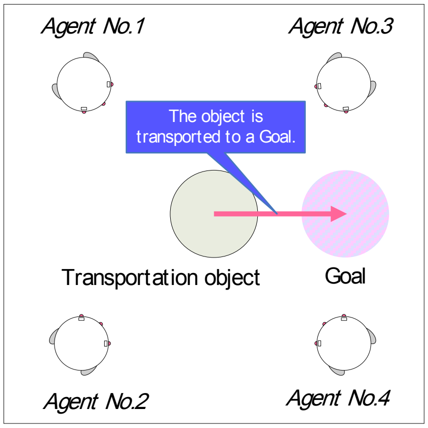
We assume that EBTS is applied to the two-hand-arm system described in previous paper [26]-[28]. Each finger is equipped with a tactile sensor with several sensing elements. In this paper, the hand-arm robot is modeled as shown in Fig. 6. Each finger of this model possesses three sensors, of which graduation is expressed by 2-bit to measure four grades of contact force. A_1, A_2, A_3 and A_4 show the status of the actuators. While A_1 shows that the agent movement is stopping (0) or active (1), A_2, A_3, and A_4 show the fingertip's movement direction (Table 2). For a multi-fingered hand, S, M and G are

$$n = 6, m = 4 \tag{5}$$

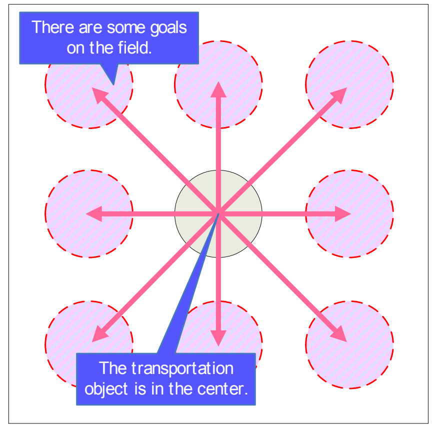
$$G = 2^6 \times 4 = 256 \tag{6}$$

Table 2 Relationship between actuator status and movement direction

A_2	A_3	A_4	Direction
0	0	0	0
0	0	1	$\pi/4$
0	1	0	$2\pi/4$
0	1	1	$3\pi/4$
1	0	0	$4\pi/4$
1	0	1	$5\pi/4$
1	1	0	$6\pi/4$
1	1	1	$7\pi/4$

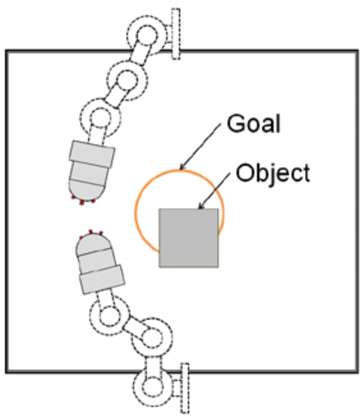


(a) object transfer task

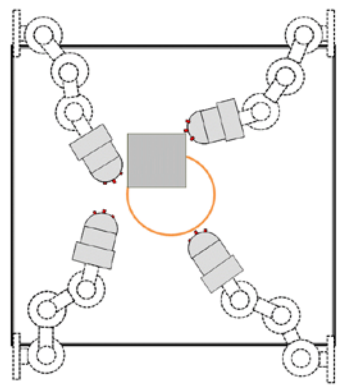


(b) relationship between object and goals

Figure 8. Map field and objects of collective task



(a) two-agent case (one hand)



(b) four-agent case (two hand)

Figure 9. Map field and objects of object handling task

In this agent model, planar movement of fingers is assumed because grasping and manipulation movement of fingers is designed. Although this agent model is similar to that of the collective task, this object transfer task is accomplished only by contact force, the goal is not emitting light, and the movement range of each hand is restricted based on the finger's kinematics.

b. Simulator Fields

An overview of the EBTS mechanism is shown in Fig. 7. Optimization of genes is performed by SGA and fitness calculation. Since we prepared a different fitness simulator for the collective task and multi-fingered hand, we explain these simulators as follows.

(1) *Collective task*

The simulator is composed of a map field and objects on it (Fig. 8). The objects are categorized into two types: an autonomous mobility agent defined as a robot and a non-autonomous object defined as the transportation object. The agent can be equipped with multiple sensor inputs and behavior outputs as its module, which is formed to function as a suitable module based on the assumptions of numerical experiments.

In this object transportation problem, the agent is equipped with three IR sensors and two photo sensors that have a fixed sensing range. The IR sensors detect walls and other objects by Boolean values in the map field. The photo sensors detect the light intensity of the fixed range that is irradiated by the transportation object and the goal.

(2) *Multi-fingered hand task*

The motion of the modeled robotic hand is restricted within two dimensions. Since the wrist is assumed to be fixed, it only manipulates an object with its fingers. We presume that the robotic fingertip transports the object to the goal (Fig. 9).

A simulator is composed of a map field (work area) and the objects on it. They are categorized into two types: an autonomous mobile agent defined as a fingertip and a non-autonomous object defined as the transportation object. The autonomous mobile agent can be equipped with multiple sensor inputs and behavior outputs as its module, which is formed to function as a suitable module based on the assumptions of numerical experiments.

c. Optimization Procedure and Fitness Calculation

The optimization procedure of the genetic algorithms (Fig. 7) is summarized as follows:

1) The population of the random gene data is produced as an initial value.

2) The evolutionary computation engine sends gene data to the simulator to evaluate the gene fitness.

3) Elite genes are selected based on their fitness.

4) A set of individuals is chosen based on the roulette wheel selection.

5) A pair of individuals is selected and used for uniform crossover.

6) The newborn children from the pair mutate under a certain probability.

7) The children’s gene data are sent to the simulator to evaluate their fitness.

8) The fitness of the elite group is compared with that of its children group. The group of the next generation population is selected from the high score group in descending order.

9) If it is not a final generation, it returns to the above procedure 3).

10) Evolutionary computation is finished.

After the agents in the collective task or multi-fingered task performed object transportation, the achievement degree of the task was evaluated as a fitness value of the gene. Evaluation functions for task achievement are composed of transportation accuracy value as E_1, evaluated from the geometrical relationship between agents and the transportation object, and transportation efficiency value as E_2, decided by reaching the goal. Evaluation function E_1 is defined as follows:

$$E_1 = \frac{K_1}{\left\| \mathbf{c}_A - \mathbf{c}_B \right\|} \tag{7}$$

where $\mathbf{c}_A$ and $\mathbf{c}_B$ are position vectors of the object and the goal, respectively. Coefficient K_1 is assumed to be 1,000 for both tasks.

On the other hand, evaluation function E_2 is defined by

$$E_2 = \frac{K_2}{t} \tag{8}$$

using time spent to transfer the object to the goal area. In this study, we assumed as $K_2 = 600$ and 160 for the collective task and the multi-fingered hand task, respectively. This is because the time consumption for each object conveyed in the multi-fingered hand task is shorter than that in the collective task.

The fitness value of the genetic algorithm is calculated as the summation of E_1 and E_2. In the fitness calculation, the agent often accidentally brought the transportation object into the goal. To avoid such accidents, we divided the simulation time into task execution and task evaluation, and compared the positioning between the agent and the transportation object at the termination of the execution time with the termination of the evaluation time.

V. EXPERIMENTAL RESULTS AND DISCUSSION

a. Collective Task

(1) *Experimental condition*

The map field of the collective task is assumed to be a Gaussian plane of 800×800 [pixel^2] with an origin at coordinates (0, 0) in a series of numerical experiments (Fig. 8). In the following, each mobile robot is called an agent because of the computer experiment. The transportation object and the agents are set at the origin and specified positions, respectively. In Fig. 8(a), four agents are placed near each corner of the rectangular field, and the agents are numbered 1-4. To relax over-adaptation in the task procedure, eight goal positions are assumed, as shown in Fig. 8 (b).

An individual attempts the object transportation eight times for each simulation. Evolution calculation, which continues for 2,000 generations, is repeated 10 times for each numerical experimental condition. The number of individuals, keeping the elite number, and mutation probability per 1-bit of gene P_m are 10, 2, and 1%. The execution time of the simulation is divided into 180 frames of the task execution time and 90 frames of task evaluation time per goal. Since the simulation is performed for eight goals, 2,160 frames are calculated. One hundred points for one goal are provided to evaluate the largest score.

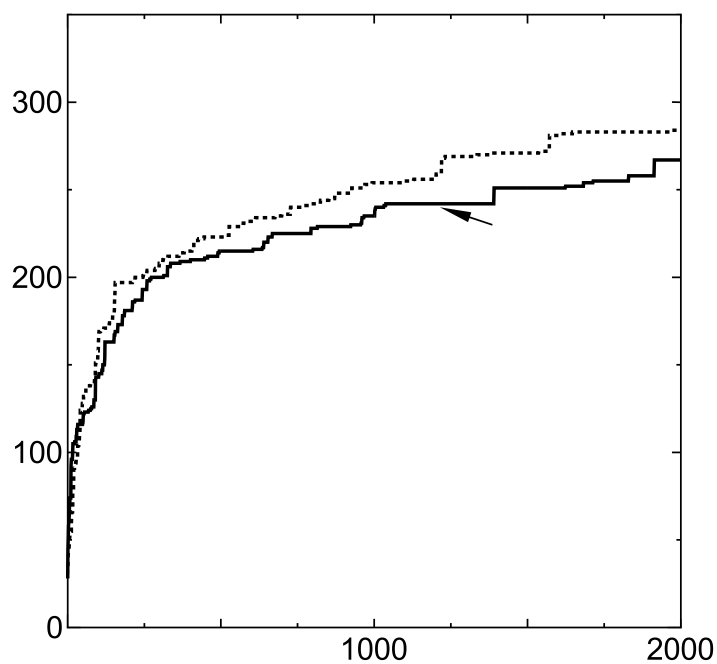


Figure 10. Relationship between maximum fitness and generation for two- and four-agent cases in collective task

(2) *Experimental results and discussion*

Figure 10 shows the relationship of two and four agents between maximum fitness and generation. Since two and four-agent tasks are performed in the multi-fingered task, we examine these two cases in the collective task as well. These two cases show that fitness increases with increase of generation, and the rate of fitness increase decreases with increase of generation. Although fitness value does not seem to become saturated even at the 2,000th generation, a fitness value over 250 is high enough to complete this task because around 5% alignment error and a few frames for task termination after task execution time are accomplished. Both two and four agents can obtain sufficient behavior performance at the 2,000th generation.

On the other hand, the fitness value of the two agents is larger than that of four agents. In the case of four agents, the agents sometimes obstruct each other. In the designed field and in the condition of this numerical experiment, the case of two agents is more efficient than that of four agents.

b. Multi-fingered Hand Task

(1) *Experimental condition*

In numerical experiments, we assume one- and two-hand cases (Fig. 9). The map field of each case is assumed to be a Gaussian plane of 800×800 [pixel^2] with an origin at coordinates (0, 0) in a series of numerical experiments, which are the same as the collective task. In the following, each fingertip is called an agent because the result of a computer experiment for the multi-fingered hand task is compared with that for the collective task. The transportation object and the agents are specified positions. In Fig. 9(b), four agents are placed near each corner of the rectangular field. The eight initial positions of the objects are assumed to relax during adaptation in the task procedure.

An individual mobile agent attempts the object transportation eight times in each simulation. Evolution calculation, which continues for 1,000 generations, is repeated ten times for each numerical experimental condition. The number of individuals, maintaining the elite number, and the mutation probability per 1-bit of gene P_m are 10, 2, and 10%. The reason for larger mutation probability than that of the collective task is that fitness value does not increase under low mutation probability.

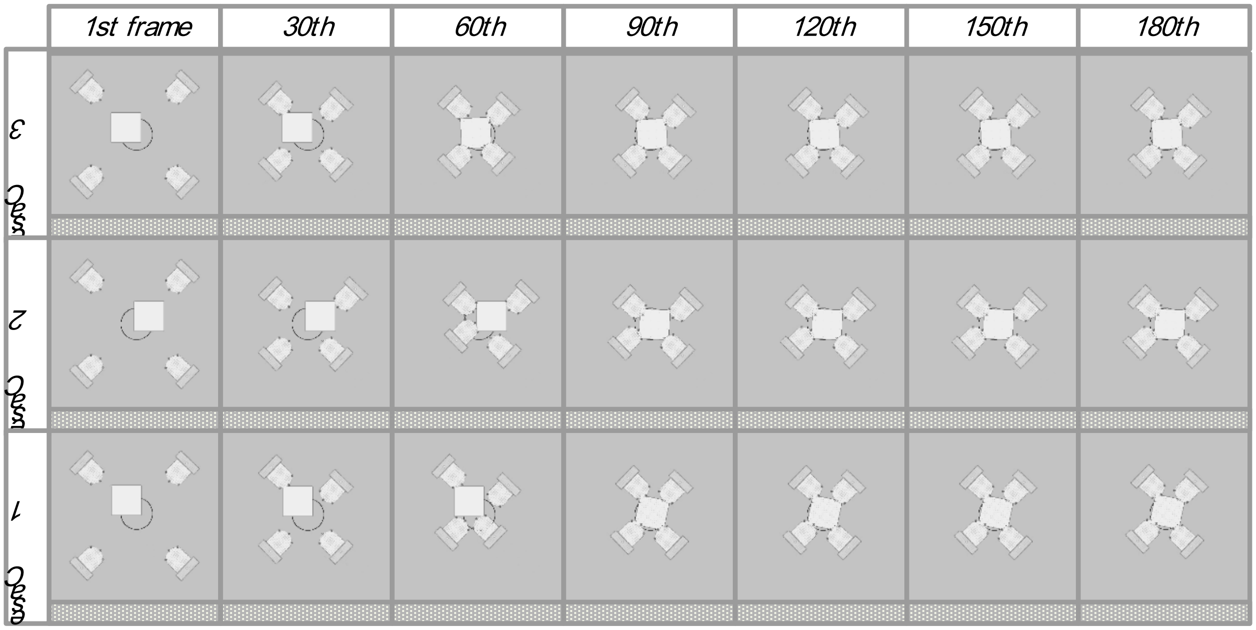


Figure 11. Four fingertips acquired cooperative behavior to convey object

(2) *Experimental results and discussion*

Figure 11 shows the task process using two hands with the four fingers shown in Fig. 9(b). Although initial position of the object is different among three cases, the two hands successfully transported the object to the center circle, which is its goal. After examining other computational experiments with several initial conditions, we verified that this task does not depend on the initial position of the object.

Figure 12 shows the maximum fitness value of the one-hand and two-hand grasping tests. The variation in the fitness value of the one-hand grasping test does not become saturated by the 1,000th generation. The efficiency of object transportation with one-hand grasping seems to be enhanced with more generations.

The four-agent (two-hands) result is shown in Fig. 12 as well. Although the fitness value of the four-agent case is increased with an increase of generation, similar to the two-agent case (one-hand), it almost becomes saturated at 1,000 generations. The fitness value of the two-hand case at 1,000 generations is around 560 and is smaller than that of the one-hand case. Therefore, the efficiency of object transportation with two-hand grasping became saturated very rapidly.

Since object grasping and transportation can be performed by one hand, the two-hand case is obviously redundant. Although redundant grasping induces stability, it decreases the efficiency of grasping and transferring.

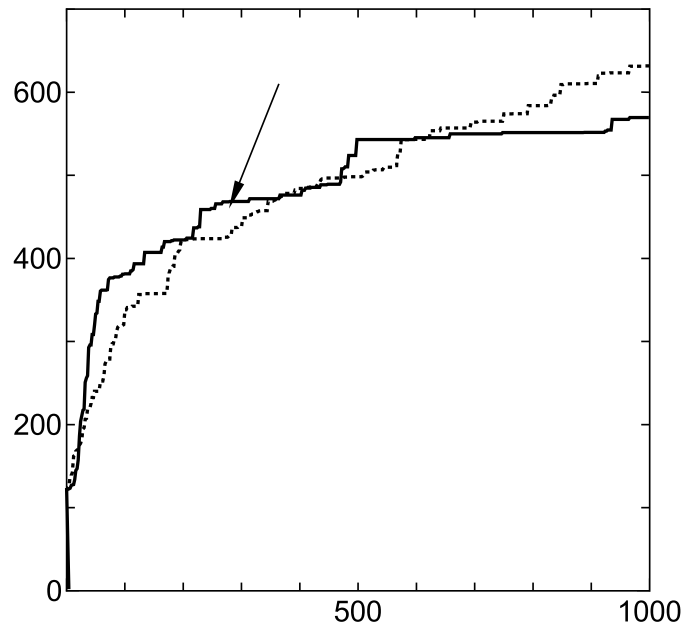


Figure 12. Relationship between maximum fitness and generation for two- and four-agent cases in multi-fingered hand task

c. Comparison Between Collection Robot and Multi-fingered Hand

We will discuss the efficiency of acquiring autonomous cooperation behavior of multi-agents in this EBTS using GA for the collection task and multi-fingered hand task. In computer experiments on both tasks, population in GA was ten and elitism selections were two in the numerical experiment for both tasks. In these experiments, the number of simulation trials was 20,000 for the collective task and 10,000 for the multi-fingered hand task as a result of optimizing the gene data until 2,000 and 1,000 generations, respectively. The final truth table obtained by the gene data doesn't always assure an optimum solution, but the calculation cost is reduced from 3.4×10^{38} to 2.0×10^{4} for the collective task and 1.2×10^{77} to 1.0×10^{5} for the multi-fingered hand task because the combination numbers of input and output patterns are calculated as $2^{128} \cong 3.4\times10^{38}$ and $2^{256} \cong 1.2\times10^{77}$, respectively. If we used the top-down methodology, 3.4×10^{38} and 1.2×10^{77} trials were needed to specify the optimum pattern because all combinations of actuator patterns should be evaluated in the top-down methodology. Therefore, we accomplish the automatic design of cooperative behavior of multi-agents for the collective task and multi-fingered hand task.

In the previous study, we concluded that EBTS is not always effective for such fine control as handling multi-fingered hand tasks. For such fine-control tasks, the relationship between sensor input and actuator output should be connected with program modules via procedural programming. However, this paper shows that EBTS can be applied to the object transfer task of the multi-fingered hand.

In both the collective task and the multi-fingered hand task, the redundancy of agents is decreased because the fitness value of four agents is smaller than that of two agents. The decreased fraction of the largest fitness in the collective task is about 10% and that in the multi-fingered task is about 3%. Even in the case of redundancy, the decreased fraction is not always large. Although the fitness value is decreased in redundant agents, EBTS is effective for the multi-fingered hand task because of automatically solving the complicated kinematics of the hand and obtaining a large enough fitness value.

VI. CONCLUSION

The bottom-up approach that is established through discussion about a creature's evolution and behavior is effective for robotic design of intelligent behavior. We proposed the Evolutionary Behavior Table System (EBTS) using an SGA to acquire the autonomous cooperation behavior of multiple agents. In EBTS, a set of rules is expressed as a table composed of sensor input columns and behavior output columns, and a row of the table corresponds to a rule. Since each rule is transformed to a string of Boolean values, we treat a long string composed of rule strings as a gene to obtain an optimum gene that adapts to the environment using SGA.

In validation experiments, EBTS was applied to two robotic tasks. One of them was a collective task, which required macro motion control of each mobile robot: in this task we assumed multi-agents equipped with the behavior table conveying an object to a specified goal with higher scores than the four-agent condition. The other was a multi-fingered hand task, which required fine motion control of each finger: we assumed that a robot grasped and transferred an object with one or two hands; each hand was equipped with two articulated fingers. The former experiment was performed for evaluating whether EBTS is applicable to the macro motion such as the collective task of social robots; the latter was performed for evaluating whether EBTS is applicable to fine motion such as handling tasks of multi-fingered hands.

In computational experiments, the collective robots could convey the object to the goal through cooperative work; the multi-fingered hands grasped the object and transferred it to the goal. Final truth tables obtained by the gene data did not always assure an optimum solution, but the calculation cost was reduced from astronomical figures to around one ten to twenty thousandth. If we used the top-down methodology, astronomical trials are needed to specify the optimum pattern. Therefore, we accomplished the automatic design of cooperative behavior of multi-agents for collective tasks using the bottom-up methodology.

These tasks were performed more efficiently in the two-agent case than in the four-agent case. Although the redundant agents caused lower fitness in both the collective task and multi-fingered hand task, the decrease of fitness was not always large. Therefore, EBTS is an attractive method because it is very useful for obtaining general robotic behaviors in both macro and fine tasks.

In future work, the effectiveness of EBTS for other tasks should be evaluated because only the object transfer task is examined in this paper. Furthermore, since EBTS seems to have some difficulty with finer tasks such as cap twisting and assembling, conventional procedural programming for a specific task is used for these tasks. Smooth connection between EBTS and the procedural programming becomes one of the important issues.

REFERENCES

[1] http://www.irobot.com/

[2] G. Dudek & M. Jenkin, Computational Principles of Mobile Robot (Second Edition), Cambridge University Press, 2010.

[3] http://www.alsok.co.jp/company/rd/robot/robot_02.html

[4] http://www.parorobots.com/index.asp

[5] T. Shibata, T. Mitsui, K. Wada and K. Tanie, “Subjective Evaluation of Seal Robot: Paro”, Journal of Robotics and Mechatronics, Vol.14, No.1, pp.13-19, 2002.

[6] M. Ohka, N. Hoshikawa, J. Wada, and H. B. Yussof, “Two Methodologies Toward Artificial Tactile Affordance System in Robotics”, International Journal on Smart Sensing and Intelligent Systems, Vol. 3, No. 3, pp.466-487, 2010.

[7] N. Hoshikawa & M. Ohka, “Object Handling Using Artificial Tactile Affordance System”, International Journal of Computer and Network Security, Vol. 2, No. 10, pp. 32-37, 2010.

[8] P. H. Winston, “Artificial Intelligence (second edition),” Addison-Wesley, pp. 159-204, 1984.

[9] J. Haugeland, “Artificial Intelligence: The Very Idea”, MIT Press, 1985.

[10] S. Harnad, “The Symbol Grounding Problem”, Physica, D 42: 335-346, 1990.

[11] J. McCarthy and P. J. Hayes, “Some philosophical problems from the standpoint of artificial intelligence”, Machine Intelligence, 4:463-502, 1969.

[12] J. McCarthy, “Applications of circumscription to formalizing common-sense knowledge. Artificial Intelligence”, 28:89-116, 1986.

[13] D. Levy & M. Newborn, “How to computer play chess”, W. H. Freeman and Company, 1991.

[14] V. Braitenberg, “Adaptation in Natural and Artificial Systems,” MIT Press, 1984

[15] R. A. Brooks, “A robust layered control system for a mobile robot,” IEEE Journal of Robotics and Automation, RA-2(1), pp. 14-23, 1986.

[16] R. A. Brooks, “Intelligence without representation,” Technical Report MIT AI Lab, 1988.

[17] R. A. Brooks, “A Robot That Walks; Emergent Behaviors from a Carefully Evolved Network”, MIT AI Lab Memo 1091, 1989.

[18] J. J. Gibson, “The Ecological Approach to Visual Perception,” Houghton Mifflin Company, 1979.

[19] M. Niitsuma, W. Beppu, T. Ichikawa, S. Kovács, P. Korondi, “Implementation of Robot Behaviors Based on Ethological Approach for Monitoring Support System in Intelligent Space”, H., 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2011), pp. 536-541, 2011.

[20] D. E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning,” Addison Wesley, 1989.

[21] T. Fukuda, T. Ueyama, and F. Arai, “Control strategy for a network of cellular robots,” IEEE International Conference on Robotics and Automation, pp. 1616-1621, 1991.

[22] R. C. Arkin, “Cooperation without communication: Multi-agent schema-based robot navigation,” Journal of Robotic Systems, 9(3), pp. 351-364, 1992.

[23] M. K. Habib, H. Asama, Y. Ishida, A. Matsumoto, and I. Endo, “Simulation environment for an autonomous and decentralized multi-agent robotic system,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1550-1557, 1992.

[24] C. R. Kube and H. Zhang, “Controlling Collective Tasks With An ALN,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 289-293, 1993.

[25] T. Hoshino, D. Mitsumoto, and T. Nagano, “Evolution of Robot Behavior and Its Robustness,” Transactions of the Society of Instrument and Control Engineers (Transactions of SICE), Vol. 33, No 6, 533-540, 1997.

[26] H. B. Yussof, J. Wada, and M. Ohka, Object Handling Tasks Based on Active Tactile and Slippage Sensations in Multi-Fingered Humanoid Robot Arm, 2009 IEEE International Conference on Robotics and Automation, pp. 502-507, 2009.

[27] M. Ohka, H. Kobayashi, J. Takata, and Y. Mitsuya, An Experimental Optical Three-axis Tactile Sensor Featured with Hemispherical Surface, Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol. 2, No. 5, pp. 860-873, 2008.

[28] M. Ohka, J. Takata, H. Kobayashi, H. Suzuki, N. Morisawa, and H. B. Yussof, Object Exploration and Manipulation Using a Robotic Finger Equipped with an Optical Three-axis Tactile Sensor, Robotica, Vol. 27, pp. 763-770, 2009.