Online Evolution for Cooperative Behavior in Group Robot Systems

Dong-Wook Lee, Sang-Wook Seo, and Kwee-Bo Sim*

Abstract: In distributed mobile robot systems, autonomous robots accomplish complicated tasks through intelligent cooperation with each other. This paper presents behavior learning and online distributed evolution for cooperative behavior of a group of autonomous robots. Learning and evolution capabilities are essential for a group of autonomous robots to adapt to unstructured environments. Behavior learning finds an optimal state-action mapping of a robot for a given operating condition. In behavior learning, a Q-learning algorithm is modified to handle delayed rewards in the distributed robot systems. A group of robots implements cooperative behaviors through communication with other robots. Individual robots improve the state-action mapping through online evolution with the crossover operator based on the Q-values and their update frequencies. A cooperative material search problem demonstrated the effectiveness of the proposed behavior learning and online distributed evolution method for implementing cooperative behavior of a group of autonomous mobile robots.

Keywords: Cooperative behavior, distributed evolutionary algorithm, distributed mobile robot system, dxperience-based crossover, Q-learning, reinforcement learning.

1. INTRODUCTION

In distributed autonomous robot systems, a team of mobile robots accomplishes complicated tasks through interactions with environments and other robots. Cooperative behaviors in distributed autonomous robot systems can be implemented using swarm intelligence and intentional cooperation [1,2]. The swarm type cooperation often deals with large numbers of homogeneous robots. The robots do not explicitly work together, but group-level cooperative behavior emerges from their interactions with each other and the environment. Distributed systems of homogeneous robots are usually more fault-tolerant than centralized or leader-follower architectures of mobile robot systems. The overall system performance does not degrade significantly by the malfunction of a small number of robots. In intentional cooperation, robots cooperate explicitly and with a purpose, usually through task-related communications. Distributed robot systems can be easily extended to handle large-scale problems, since the communication complexity does not increase much as the number of robots increases [3,4].

An autonomous robot can demonstrate two types of interactions: sensing and communication. Individual robots sense the existence and recognize the types of objects such as target materials and obstacles. Autonomous robots are required to cooperate with other robots in a dynamic, unstructured environment such as space and deep sea. A set of fixed control rules will not work in such operating environments. The controller must be able to adaptively determine the optimal actions at each step. Cooperative behavior of autonomous mobile robots emerges from local communications between individual robots. A group of mobile robots exchange information with neighboring individuals within a communication range to accomplish the tasks in cooperative manner.

Behavior learning finds an optimal state-action mapping of a mobile robot for a given operating condition. Each robot is required to decide an optimal action for a set of given sensor inputs. In reinforcement learning, an agent effectively learns the behaviors by a reinforcement signal when a prior knowledge on the environment is not available. Popular reinforcement learning algorithms include actor-critic architecture based on time differentiate (TD) method [5,6] and Q-learning [7-10]. Each robot improves the current state-action rules by Q-learning according to the reward or penalty given by the result of an action. In distributed autonomous robot systems,

Manuscript received July 20, 2007; revised December 29, 2007; accepted February 12, 2008. Recommended by Editorial Board member Young-Hoon Joo under the direction of Editor Jae-Bok Song. This research was supported by the Development of Social Secure Robot using Group Technologies of Growth Dynamics Technology Development Project by Ministry of Commerce, Industry and Energy, Korea.

Dong-Wook Lee is with the Division for Applied Robot Technology, Korea Institute of Industrial Technology, Korea (e-mail: dwlee@kitech.re.kr).

Sang-Wook Seo and Kwee-Bo Sim are with the School of Electrical and Electronics Engineering, Chung-Ang University, 221, Heukseok-dong, Dongjak-gu, Seoul 156-756, Korea (e-mails: ssw0511@wm.cau.ac.kr, kbsim@cau.ac.kr).

* Corresponding author.

however, reward and penalty terms may not be calculated immediately due to the delay in evaluation. This paper presents a modified Q-learning algorithm to handle delayed rewards.

Cooperative behaviors of autonomous robots can be developed from evolutionary operations of the information of individual robots. Robots exchange information through local communication with other individuals. Conventional evolutionary algorithms rely on the operations such as selection, crossover, and mutation in a population of individuals. Crossover operation usually finds two offspring chromosomes from two parents. Distributed evolutionary algorithms enable an individual robot to improve the learning ability online through exchanging the acquired information with other robots. In distributed evolutionary algorithms, system components are evolved separately. For example, a population [11,12] or a chromosome [13] can be divided into subgroups and are evolved independently in multiple parallel processors. Each mobile robot retains one of the two chromosomes having more update frequencies of Q-values. Such experience-based crossover operation selects the genes to increase the probability to keep superior genes in the subsequent generations.

This paper presents behavior learning of individual autonomous robots based on reinforcement learning and online distributed evolutionary algorithm for cooperative behaviors of the robots in unstructured environments. Individual robots develop an optimal state-action mapping by the behavior learning. Cooperative behaviors of the robots evolve through the communications with other individuals within a communication range. A group of autonomous mobile robots are required to search and collect target materials scattered in an open space as quickly as possible in a cooperative manner without collisions with obstacles and other robots. Each robot interacts with the environment through the sensors mounted on the perimeter of the body. The sensors detect the existence of objects and recognize target materials and obstacles. The Q-learning finds the best state-action pairs for behavior learning of individual robots. The robots build cooperative behaviors online using the distributed evolutionary algorithms. A robot communicates with neighboring robots within a communication range to exchange information. When a robot encounters superior state-action rules, the robot receives the rules and reproduces new rules using evolutionary operations.

2. BEHAVIOR LEARNING OF AUTONOMOUS ROBOTS

2.1. Autonomous mobile robot

A group of robots are required to search and collect target materials spread over a space in collaboration with other robots. A robot has the abilities such as local communications with neighboring robots and collision avoidance with obstacles or other robots. An individual mobile robot is equipped with two wheels, sensors, actuators, and communication devices. Fig. 1 shows a sensor arrangement of a mobile robot. A robot can detect the existence of near objects and measures the distance to the object having infrared (IR) sensors within a limited sensing range. A robot is assumed to be able to distinguish the target materials from the obstacles and robots based on the color. There are eight sensors around the robot, 45 degrees apart. The sensors are grouped into four directions: Forward (S_0), Right (S_1, S_2, S_3), Rear (S_4), and Left (S_5, S_6, S_7). Only one sensor becomes active at a time for a near object in that direction. Each sensor can have three possible sensing states: No Object (0), Material (1), and Object (obstacle or robot) (2). From the sensor inputs, a robot detects three possible states in each of four directions. The total number of possible states of a robot is 81 ($= 3^4$). Sensing range is usually much smaller than a communication range.

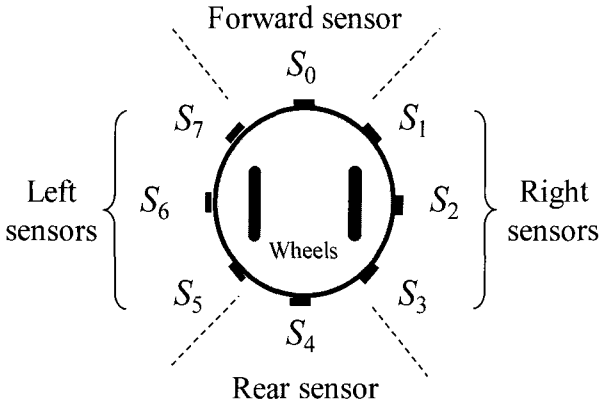


Fig. 1. Sensor arrangement of mobile robot.

The behavior of a robot can be defined by a state-action mapping. Five actions are defined as follows:

- Random Move (RM)
- Move Forward (MF)
- Turn Right (TR)
- Turn Left (TL)
- Approaching Target (AT)

Random Move refers to turning to an arbitrary direction and moving forward. *Move Forward* defines the moving in the forward direction. *Turn Right* and *Turn Left* define the moves that a robot turns 45 degrees to the right and to the left and move forward. *Approaching Target* defines the movement toward a detected object. If more than one object is detected, a robot moves toward the nearest object. If no object is detected, a robot moves forward. A robot has no *a priori* knowledge that an object is useful to approach.

2.2. Robot behavior learning with reinforcement learning

Behavior learning refers to finding an optimal state-action mapping of a mobile robot for a given operating condition. Each robot is required to make an optimal decision for an action given sensor inputs. Reinforcement learning is suitable especially for agent-based applications, since the signal used to learn the model comes from elaboration of the reinforcement function to represent the behavior of agents. Reinforcement learning maximizes the rewards that a learning agent receives to improve the behaviors through the interaction with the environment using a reinforcement signal [6].

Q-learning [7] has been developed as a method of model-free reinforcement learning based on stochastic dynamic programming. Q-learning is suitable in robotics applications since it is applicable to online learning with finite states and actions of a robot. A robot gradually learns the behavior rules through the Q-learning mechanism. A robot can take a set of actions ($\boldsymbol{A}$) given a set of states ($\boldsymbol{S}$). A state and action mapping is stored in the form of Q-table, a collection of all the possible Q-values of state-action combinations. In this paper, the set $\boldsymbol{S}$ consists of 81 states and the set $\boldsymbol{A}$ has five actions that correspond to 405 (= 81×5) Q-values. As the iterations of Q-learning go on, one of Q-values becomes dominant for each state. A state-action pair with a dominant Q-value is regarded as an optimal state-action rule.

In this paper, a modified Q-learning is used for behavior learning of individual mobile robots. In a distributed autonomous mobile robot system, reward (or penalty) for a robot behavior may not be calculated immediately, but after a series of behaviors. Hence delayed rewards for an action must be counted. Algorithm 2 shows a modified Q-learning algorithm with delayed rewards. Delayed reward takes an important role that it enforces the previous steps that affect current action. In (1), a temperature coefficient T reduces the probability of behavior selection as learning proceeds. As the T-value decreases, the difference of the $P(a)$ values of each action for a random state s becomes large, so the probability of choosing the action with the largest Q-value increases. In early stage, the probability of choosing various actions is high (exploration). As learning gradually proceeds, the system uses previously learned results (exploitation). A series of previous actions affect the current action with decreasing influence. The term β^k ($0 < \beta < 1$) is introduced to reduce the effect of previous actions to current actions gradually as the step goes to the maximum K previous steps.

Algorithm 1 (Q-learning Algorithm with Delayed Reward):

1. Initialize $Q(s_i, a_j)$ to small values for all the states $s_i \in \boldsymbol{S}$, $i = 1, \cdots, N_s$, and actions $a_j \in \boldsymbol{A}$, $j = 1, \cdots, N_a$. N_s and N_a denote the numbers of states and actions.
2. Obtain the current state s.
3. Choose an action a in proportion to the probability

$$P(a_i) = \frac{\exp\left(Q(s,a_i)/T\right)}{\sum_{j=0}^{N-1} \exp\left(Q(s,a_j)/T\right)}, \tag{1}$$

where T is a temperature parameter that gradually decreases to zero.

4. Carry out action a in the environment. Let the next state be s'.
5. If a delayed reward r is calculated then update current Q-value $Q(s_0,a_0)$ and past Q-values $Q(s_k,a_k)$, $k = 1, \ldots, K$.

$$Q_{t+1}(s_k,a_k) = (1-\alpha)Q_t(s_k,a_k) + \alpha\left[\beta^k r + \gamma \max_{a_k' \in A} Q_t\left(s_k', a_k'\right)\right], \tag{2}$$

where K denotes the maximum previous steps that affect current action, and β is a constant between 0 to 1.

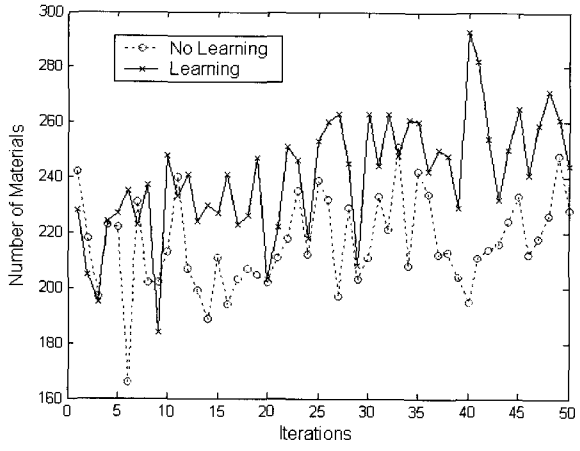
6. Repeat the steps 2-5.

After the learning is completed, we pick the action corresponding to the maximum Q-value. The relation of states and actions can be represented as a Q-table [7,8]. The Q-table consists of 405 Q-values corresponding to 81 states s_i, $i = 0, 1, \ldots, 80$ and 5 actions a_0 (RM), a_1 (MF), a_2 (TR), a_3 (TL), a_4 (AT) in the form of 81-by-5 matrix. Each state s_i is composed of four sensor inputs of [Forward, Right, Rear, Left]. For example, a robot senses a material in the right and an object in the rear, the state of the robot becomes [0 1 2 0].

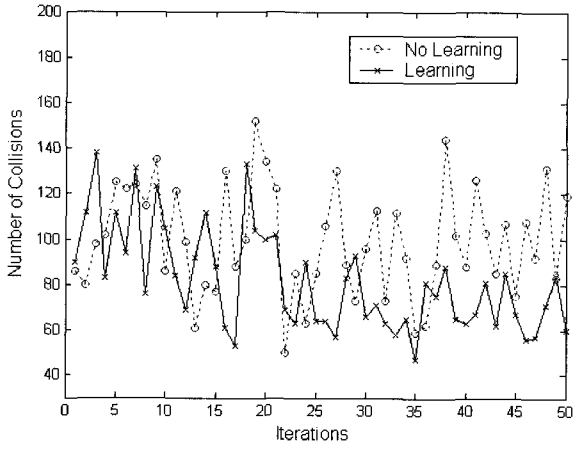
2.3. Material collection experiment

As an experimental setting, the materials and obstacles are randomly scattered in a working space. There are 25 mobile robots used in this experiment of the diameter 0.05m. The sensing range is assumed 0.44m. The actions of RM, TR, and TL involve a robot movement of turning and moving of 0.1 m. In MF action, a robot moves forward of 0.15m. Mobile robots search and collect target materials in a workspace. At each iteration, the workspace is reset with randomly generated target materials and obstacles. Algorithm 2 needs some parameters to be decided by user heuristically. For example, there are T, α, β, γ, and K. The temperature parameter T was chosen as the function $T(j) = 2 - 0.03j$ at j-th iteration. The other parameter values used in this experiment are $\alpha = 0.1$, $\beta = 0.75$, $\gamma = 0.25$, and $K = 3$. r equals +1 for reward, and r equals −1 for penalty.

Fig. 2(a) shows the number of target materials collected by a group of robots.



(a) Number of materials collected.



(b) Number of collisions.

Fig. 2. Performance comparison of with and with-out Q-learning.

3. ONLINE EVOLUTION OF COOPERATIVE BEHAVIOR

This paper demonstrates cooperative behavior of a group of mobile robots through local interactions with neighboring robots. A group of robots are expected to search and collect target materials scattered in a workspace as quickly as possible while avoiding collisions with the obstacles and the other robots. Robots cooperate with each other using local communications to reduce the time to collect all the materials. Robots within communication range exchange information to implement cooperative behaviors. Each robot evolves by exchanging learned information with other robot through local communications.

In distributed evolutionary algorithm, each robot can calculate the fitness value by reinforcement learning and can select and reproduce by communications. The fitness is calculated for all robots under same condition. A robot calculates the fitness value using (3) based on rewards, penalties, and consumed energy during the evaluation time T_{eval}, which has been set to 300 sec. If a robot is not evaluated during T_{eval} after reproduction of its chromosome, the robot cannot exchange the information with other robot because the robot has no fitness value of new generated chromosome. A robot selects the other robot to crossover based on the fitness value computed during the evaluation time.

$$\text{Fitness} = w_1 N_r - w_2 N_p, \tag{3}$$

where N_r and N_p denote the numbers of rewards and penalties. The parameters w_1 and w_2 are positive weight values.

If robot A encounters robot B whose fitness is higher, for example, then robot A receives the chromosome of robot B and reproduces chromosome using the experience-based crossover. In this case, robot B does not change the chromosome. The information is passed from superior robot to inferior robots. A robot improves the performance by combining other robot's chromosome obtained from different environment with the chromosome. The state-action rules in the form of Q-table are encoded in chromosomes for evolution operation.

This paper proposes a new crossover method based on learning times to find a chromosome for a robot. A chromosome consists of Q-values and L-values as the number of updates of Q-values. Therefore the crossover uses learning frequencies (L-values) as well as Q-values. A chromosome of robot can be represented by a pair of $\mathbf{x}$ (Q-value) and l (L-value) of the parents.

$$\left(\mathbf{X}^p, \mathbf{l}^p\right) = \left[\left(\mathbf{x}_1^p, \cdots, \mathbf{x}_m^p\right), \left(l_1^p, \cdots, l_m^p\right)\right], \tag{4}$$

where m is the total number of genes. A gene is a subset of Q-values that have same state. For example, a robot has one chromosome that is composed of 81 genes and a gene is composed of 5 Q-values. New offspring generated by the crossover is represented as

$$\left(\mathbf{X}^o, \mathbf{l}^o\right) = \left[\left(\mathbf{x}_1^{s_1}, \cdots, \mathbf{x}_m^{s_m}\right), \left(l_1^{s_1}, \cdots, l_m^{s_m}\right)\right], \tag{5}$$

where

$$s_i = \begin{cases} 1 & p_i < \dfrac{l_i^1}{l_i^1 + l_i^2}, i = 0, \cdots, m \\ 2 & \text{otherwise,} \end{cases}$$

p_i is a random number from 0 to 1. The chromosomes of offspring are inherited from parents 1 and parents 2 according to the learning frequencies (l). Robots share the information on the environment that they have not yet been in. As a result, a robot obtains learning data on the environment that the robot has not been from

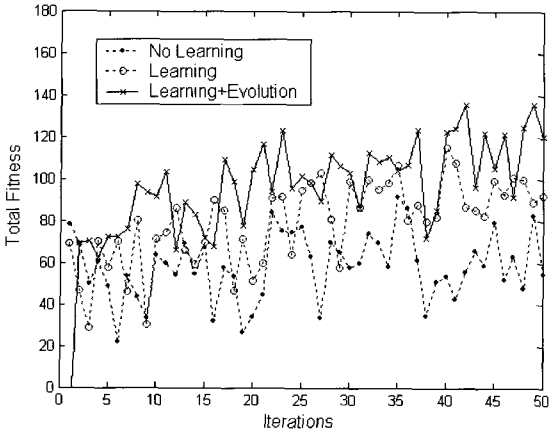
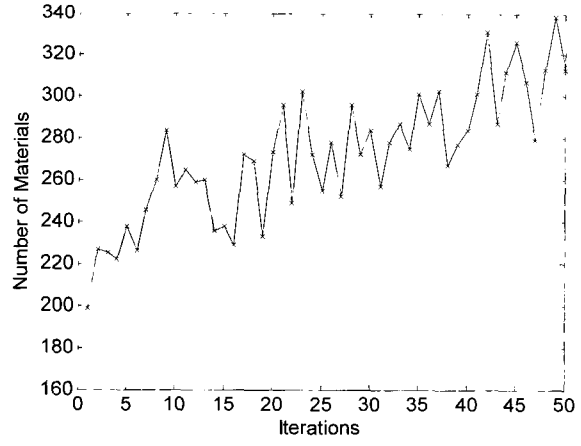
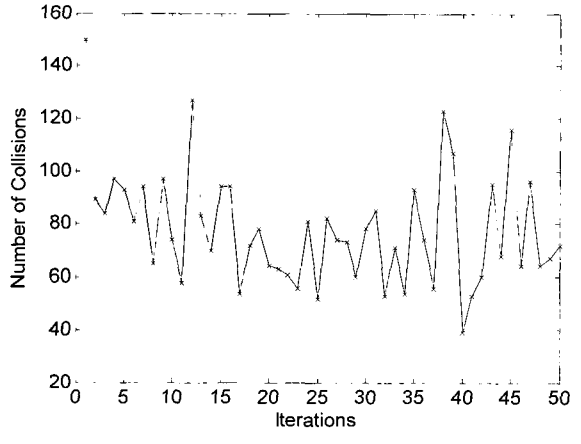


Fig. 3. Relationship between the total fitness variation and iteration numbers.



(a) Number of materials collected.



(b) Number of collisions.

Fig. 4. Evolution trends.

other individual robots by experience-based crossover. A robot gets better chromosomes from other robots so that the robot indirectly learns the environment that it has not been experienced before.

The proposed online evolution method was compared for the three cases: (1) No learning and evolution, (2) Learning only, and (3) Learning and evolution. Case 1 uses the robots with no reinforcement learning for the behavior and no evolution through local communications with the other robots. In Case 2, the robots learn the environment to avoid collisions with other objects using Q-learning. Case 3 involves the robots with behavior learning capability and online evolution. Fig. 3 shows the total fitness variation of the robot system as iteration increases for $w_1 = w_2 = 0.5$. Fig. 4 shows evolution trends when the robots use learning and evolution with experience-based crossover. The robot system with learning and online evolution capability collects the materials more effectively. Total fitness is calculated using the number of collected materials and collisions for iteration. Total fitness is the difference between the number of collected materials and the number of collisions. The total fitness of Case 3 increases faster than the other cases. The performance of robot system is improved as a result of online evolution with experience-based crossover.

4. CONCLUSION

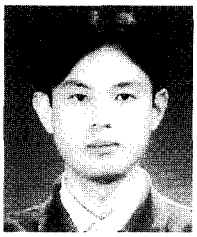
In distributed mobile robot systems, autonomous robots cooperate with each other to accomplish complicated tasks in unstructured environment. This paper presents behavior learning and online distributed evolution for cooperative behavior of a group of autonomous mobile robots. Behavior learning finds an optimal state-action mapping for a given operating condition. A robot develops a set of optimal state-action rules for given operating environments. In behavior learning, a Q-learning algorithm is modified to handle delayed rewards in the distributed robot systems. A group of robots implements cooperative behaviors through local communications with other robots. Individual robots improve the state-action mapping through online evolution with the crossover operator based on the Q-values and their update frequencies. Such experience-based crossover operation selects the genes to increase the probability to retain superior genes in the subsequent generations. A cooperative material search problem demonstrated the effectiveness of the proposed behavior learning and online distributed evolution method for implementing cooperative behavior of distributed mobile robot systems.

REFERENCES

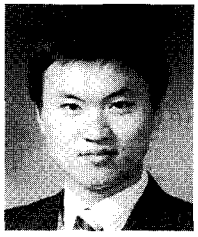
[1] L. E. Parker, "ALLIANCE: An architecture for fault-tolerant multirobot cooperation," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 2, pp. 220-240, April 1998.

[2] P. J. 't Hoen, K. Tuyls, L. Panait, S. Luke, and J. A. La Poutre, "An overview of cooperative and competitive multiagent learning," *Learning and Adaption in Multi-Agent System*, LNAI 3898, pp.

1-46, 2006.
[3] H. Asama, "Perspective of distributed autonomous robotic systems," *Distributed Autonomous Robotic Systems 5*, H. Asama, T. Arai, T. Fukuda, T. Hasegawa (Eds.), Springer, pp. 3-4, 2002.
[4] T. Arai, E. Pagello, and L. E. Parker, "Advances in multirobot systems," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, pp. 655-661, October 2002.
[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.
[6] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, 1997.
[7] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279-292, 1992.
[8] L. P. Kaelbling, "On reinforcement learning for robotics," *Proc. of Int. Conf. on Intelligent Robot Systems*, pp. 1319-1320, 1996.
[9] S. O. Kimbrough and M. Lu, "Simple reinforcement learning agents: Pareto beats Nash in an algorithmic game theory study," *Information Systems and E-Business Management*, vol. 3, no. 2, pp. 1-19, March 2005.
[10] L. E. Parker, C. Touzet, and D. Jung, "Learning and adaptation in multi-robot teams," *Proc. of Eighteenth Symposium on Energy Engineering Sciences*, pp. 177-185, 2000.
[11] M. Nakamura, N. Yamashiro, and Y. Gong, "Iterative parallel and distributed genetic algorithms with biased initial population," *Proc. of Congress on Evolutionary Computation*, vol. 2, pp. 2296-2301, 2004.
[12] A. L. Jaimes and C. A. Coello, "MRMOGA: A new parallel multi-objective evolutionary algorithm based on the use of multiple resolutions," *Concurrency and Computation: Practice and Experience*, vol. 19, no. 4, pp. 397-441, March 2007.
[13] T. Fukuda and T. Ueyama, *Cellular Robotics and Micro Robotic System*, World Scientific, 1994.



Dong-Wook Lee received the B.S., M.S., and Ph.D. degrees in the Department of Control and Instrumentation Engineering from Chung-Ang University in 1996, 1998, and 2000, respectively. Since 2005, he has been with the Division for Applied Robot Technology at Korea Institute of Industrial Technology (KITECH), where he is currently a Senior Researcher. His areas of include artificial life, android, emotion model, learning algorithm, and distributed autonomous robot systems.



Sang-Wook Seo received the B.S. degree in the Department of Electrical and Electronics Engineering from Chung-Ang University, Seoul, Korea, in 2007. He is currently Master course in the School of Electrical and Electronics Engineering from Chung-Ang University. His research interests include machine learning, multi agent robotic system, evolutionary computation, evolutionary robot, etc.

Kwee-Bo Sim received the B.S. and M.S. degrees in the Department of Electronic Engineering from Chung-Ang University, Korea, in 1984 and 1986 respectively, and the Ph.D. degree in the Department of Electronics Engineering from the University of Tokyo, Japan, in 1990. Since 1991, he is currently a Professor. His research interests include artificial life, emotion recognition, ubiquitous intelligent robot, intelligent system, computational intelligence, intelligent home and home network, ubiquitous computing and Sense Network, adaptation and machine learning algorithms, neural network, fuzzy system, evolutionary computation, multi-agent and distributed autonomous robotic system, artificial immune system, evolvable hardware and embedded system etc. He is a Member of IEEE, SICE, RSJ, KITE, KIEE, KIIS, and ICROS Fellow.