

Behavior Control of Autonomous Mobile Robot using Schema Co-evolution

스키마 공진화 기법을 이용한 자율이동로봇의 행동제어

Chi-Sun Joung*, Hyo-Byung Jun, and Kwee-Bo Sim

Robotics and Intelligent Information System Laboratory
Dept. of Control and Instrumentation Engineering, Chung-Ang University
221, Huksuk-Dong, Dongjak-Ku, Seoul 156-756, Korea
Tel:+82-2-820-5319, Fax: +82-2-817-0553
E-mail:kbsim@cau.ac.kr, URL:http://rics.cie.cau.ac.kr

ABSTRACT

The theoretical foundations of GA are the Schema Theorem and the Building Block Hypothesis. In the meaning of these foundational concepts, simple genetic algorithm(SGA) allocate more trials to the schemata whose average fitness remains above average. Although SGA does well in many applications as an optimization method, still it does not guarantee the convergence of a global optimum. Therefore as an alternative scheme, there is a growing interest in a co-evolutionary system, where two populations constantly interact and co-evolve in contrast with traditional single population evolutionary algorithms. In this paper, we propose a new design method of an optimal fuzzy logic controller using co-evolutionary concept. In general, it is very difficult to find optimal fuzzy rules by experience when the input and/or output variables are going to increase. So we propose a co-evolutionary method finding optimal fuzzy rules. Our algorithm is that after constructing two population groups made up of rule base and its schema, by co-evolving these two populations, we find optimal fuzzy logic controller. By applying the proposed method to a path planning problem of autonomous mobile robots when moving objects exist, we show the validity of the proposed method.

I. Introduction

Recently artificial life concept was proposed by C. Langton and has become one of the most popular research area as a solution of intelligent information processing system under uncertain, complex and dynamic environment. Main issue in artificial life is how to implement something lifelike with computer and robots by synthesizing phenomena normally associated with natural living systems. The evolutionary computation based on the natural selection theory plays an important role in artificial life.

The concept of natural selection has influenced our view of biological systems tremendously. Evolutionary Algorithms(EAs) are computational models of living system's evolution process and population-based optimization methods. EAs can provide many opportunities for obtaining a global optimal solution, but the performance of a system is deterministic depending on the fitness function given by a system designer. Thus EAs generally work on static fitness landscapes. But natural evolution works on dynamic fitness landscapes

that change over evolutionary time as a result of co-evolution. And co-evolution between different species or different organs results in the current state of complex natural systems. In this point, there is a growing interest in co-evolutionary systems, where two populations constantly interact and co-evolve in contrast with traditional single population evolutionary algorithms. This co-evolution method is more similar to biological evolution in nature than other evolutionary algorithms.

In this paper, we propose a co-evolution method generating optimal fuzzy rule base, where the fitness of a population changes according to the evolution process of the other population. We presents how to extract fuzzy rules using schema coevolution. In general, it is very difficult to find fuzzy rules by hand when the input-output variables are going to increase. In this paper, therefore, we extract fuzzy rules by co-evolving the fuzzy rules and their schema.

To show the effectiveness of the proposed method, we applied our method to autonomous mobile robotic system, the objective of which is finding a goal and avoiding static/moving obstacles.

II. Co-Evolutionary Algorithm

Recently evolutionary algorithms, including genetic algorithms(GAs), evolutionary strategies(ES), evolutionary programming(EP), genetic programming (GP), has been widely studied as a new approach to artificial life. All of these typically work with a single population of solution candidates scattered on the static landscape fixed by the designer. But in nature, various feedback mechanisms between the species undergoing selection provide a strong driving force toward complexity. And natural evolution works on the fitness landscapes that changes over the evolutionary time. From this point of view, co-evolution algorithms have much attractions in intelligent systems.

Generally co-evolution algorithms can be classified into two categories, which are predator-prey co-evolution and symbiotic co-evolution.

2.1. Predator-Prey Co-Evolution[1]

Predator-prey relation is the most well-known example of natural co-evolution. As future generations of predators develop better attacking strategies, there is a strong evolutionary pressure for prey to defend themselves better. In such arms races, success on one side is felt by the other side as failure to which one must respond in order to maintain one's chances of survival. This, in turn, calls for a reaction of the other side. This process of co-volution can result in a stepwise increase in complexity of both predator and prey[1]. Hillis[2] proposed this concept with a problem of finding minimal sorting network for a given number of data.

2.2. Symbiotic Co-evolution[2][3]

Symbiosis is the phenomenon in which organism of different species live together in close association, resulting in a raised level of fitness for one or more of the organisms. In contrast of predator-prey, this symbiosis has cooperative or positive aspects between different species.

Paredis[3] proposed a symbiotic co-evolution in terms of SYMBIOT, which uses two co-evolving populations. One population contains permutations (orderings), the other one consists of solution candidates to the problem to be solved. And another approach to symbiotic co-evolution is host-parasite relation. Just as do other co-evolutionary algorithms, two co-evolving populations are used. One is called host population which consists of the candidates of solution, the other contains schema of the solution space. This idea is based on the schema theorem and building block hypothesis. The schema theorem is that short, low-order, above-average schemata receive exponentially increasing trials in subsequent generations of a genetic algorithm[4].

The individual of host-population is parasitized by a schema in parasite population. By this process, useful schema generates much more instances in host population at the next generation.

III. Co-evolution Schema Theorem[5]

If a string y in the parasite-population represents a schema H , it is clear that the above parasitizing process can be interpreted, in the context of useful schemata, as a process of increasing the number of instances of a schema H in the host-population. If we recall the original schema theorem, the number of instances of a schema H at the generation k is changed by the amount of newly generated instances of that schema. When the co-evolution is considered the number of instances $m'(H, k)$ of a schema H in the host-population at the generation k is expressed by

$$m'(H, k) = m(H, k) + \widehat{m}(H, k) \quad (1)$$

where $m(H, k)$ is the original number of instances of a schema H in the host-population. And $\widehat{m}(H, k)$ is the increased number of instances by the parasitizing process and can be stated as follows:

$$\widehat{m}(H, k) = \frac{1}{2} \sum_{i=1}^n \{ \text{sgn}[f(\widehat{x}_{iH}, k) - f(x_i, k)] + 1 \} \quad (2)$$

where $\text{sgn}(u)$ is a sign function that equals +1 for positive u and -1 for negative u . Note that since we focus on the newly generated instances after parasitizing, the case that x_i is identified with \widehat{x}_{iH} is excluded from the equation (2). This equation means that since the string x_i is exchanged for \widehat{x}_{iH} in the case that the degree of improvement in the fitness is above 0, the instances of a schema H in the host-population are increased.

Also we can formulate the fitness of a schema H associated with host-parasite co-evolution from its definition. Let us denote by $f'(H, k)$ the fitness of a schema H after parasitized at the generation k . Then

$$f'(H, k) = \frac{\sum_{x \in I_H} f(x, k) + \sum_{x_i \in \widehat{I}_H} f(\widehat{x}_{iH}, k)}{m(H, k) + \widehat{m}(H, k)} \quad (3)$$

where I_H is a set of instances of a schema H at the generation k and \widehat{I}_H is a index set of increased instances of a schema H after parasitized. Combining the above equations, the schema theorem can be rewritten by

$$m(H, k+1) \geq m'(H, k) \cdot \frac{f'(H, k)}{f(k)} \cdot \left[1 - p_c \cdot \frac{\delta(H)}{l-1} - p_m \cdot o(H) \right] \quad (4)$$

Since the fitness of a schema H is defined as the average fitness of all strings in the population matched by that schema H , the fitness $f'(H, k)$ of a schema H after parasitized can be approximated by $f'(H, t) \simeq f(H, t)$. Especially, if the number of strings in the host-population $N_H \gg n$, the above approximation makes sense for the large number of generation sequences[4].

Consequently we obtain an *extended* schema theorem associated with host-parasite co-evolution that is

$$m(H, k+1) \geq [m(H, k) + \widehat{m}(H, k)] \cdot \frac{f(H, k)}{f(k)} \cdot \left[1 - p_c \cdot \frac{\delta(H)}{l-1} - p_m \cdot o(H) \right]. \quad (5)$$

Compared with the original Schema Theorem[4][6], the above equation means that the short, low-order, above-average schema H would receive an exponentially increasing number of strings in the next generation with higher order than SGA. Additionally the parasitizing process gives more reliable results in finding an optimal solution. Because the parasite-population explores the schema space, a global optimum could be found more reliably in shorter time than SGA. When the schema containing a solution does not exist in the population, SGA may fail to find global optima. In the other hand, because the useful schema can be found by the parasite-population, co-evolution gives much more opportunities to converge to global optima.

IV. Schema co-evolutionary Construction of FLC

4.1. Rule base Population.

The individual of rule base population consists of a set of rules, so there are sets of rules in the rule population. And a set of rules is made up of ten different rules. If membership functions are partitioned into five terms and there are n preconditions, then the maximum number of IF-THEN fuzzy rules is 5^n . This means that the input space is divided into 5^n . Therefore, unless we use all of the rules, null set problems occur when the given rule base cannot cover the current input states. So we use a don't-care symbol in addition to linguistic terms for a rule chromosome. This don't-care symbol makes the preconditions so inclusive that a small number of rules can cover the whole input space. An example of encoding scheme for several given rules is shown in Fig. 1.

In order to ensure the character preservingness, we use the Elitism method and a mutation operator only as genetic operators. This selection method is elitist and therefore guarantees a monotonically improving performance.

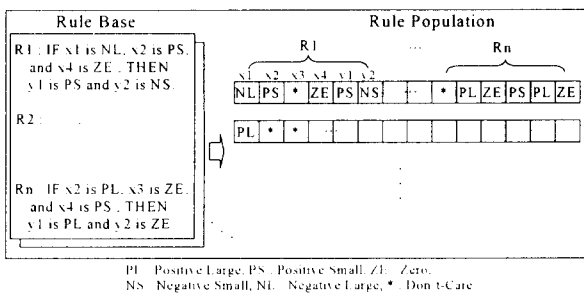


Fig. 1. An Example of fuzzy rules encoding scheme

4.2. Process of schema co-evolution

The process of host-parasite co-evolution is that a useful schema found by the parasite-population is delivered to the host-population according to the fitness proportionate, and the evolutionary direction of the parasite-population is determined by the host-population.

The fitness F_y of a string y in the parasite-population is determined as follows:

Step 1. Determine a set of strings of the host-population to be parasitized. Namely select randomly n strings in the host-population, which are parasitized by a schema y .

Step 2. Let the sampled strings as x_1, \dots, x_n , and the parasitized strings as $\hat{x}_{1y}, \dots, \hat{x}_{ny}$. A parasitized string is a sampled string after parasitized by a schema y .

Step 3. In order to determine the fitness of a string y in the parasite-population, we set a fitness function of one time parasitizing as improvement of the fitness.

$$\hat{f}_{iy}(k) = \max [0, f(\hat{x}_{iy}, k) - f(x_i, k)] \quad (i=1, \dots, n) \quad (6)$$

where $f(x_i, k)$ is the fitness of a string x_i at generation k , and $f(\hat{x}_{iy}, k)$ is the fitness of a string \hat{x}_{iy} which is parasitized by a schema y .

Step 4. Then the fitness F_y of a schema y in the parasite-population is

$$F_y = \sum_{i=1}^n \hat{f}_{iy} \quad (7)$$

By exchanging a string x_i for \hat{x}_{iy} which is a string having maximum value of \hat{f}_{iy} , still one of the strings parasitized by a schema y , the genetic information acquired by parasitizing is delivered to the host-population. And as described in equation (7), the fitness of a schema in the parasite-population is depending on the parasitized strings in the host-population.

V. Path planning of AMR

We verify the effectiveness of the proposed algorithm by applying it to optimal path planning of autonomous mobile robot. The objective of this problem is to find an optimal path when static and moving obstacles exist. For the moving obstacle we assumed that there are two robots with the same FLC at the counterpart corner. Each robot's goal position is set to the other robot's starting point and perceives the other robot as an obstacle. A robot has three sensors (S_0, S_1, S_2) covering $\pm 15^\circ$ to detect the distance to a obstacle. And the direction of its goal (θ) is given, so there are four input variables. For the outputs, FLC gives the directional changes (φ) and speed (v) of AMR.

The input/output variables' ranges are restricted as shown in table 1. And Fig. 2 shows the AMR's sensor configuration and situations of detecting an obstacle.

Table 1. Range of input/output variables

INPUT				OUTPUT	
θ	S_0	S_1	S_2	φ	v
$-180^\circ \sim 180^\circ$	0 ~ 200 mm	0 ~ 200 mm	0 ~ 200 mm	$-90^\circ \sim 90^\circ$	0 ~ 30 mm

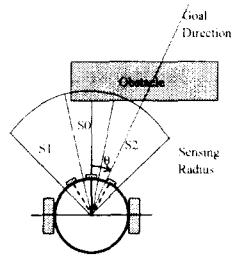


Fig. 2 Sensor Configuration

And the rule fitness measure is formulated by,

$$fit = \left(1 - \frac{D_r}{D_G}\right) \cdot \frac{T_{min}}{T} \cdot \frac{(N_N - N_n)}{N_N} \quad (8)$$

where T is consuming time, N_n is the number of null set, T_{min} is minimum time required to reach the goal, and N_N is maximum number of null set.

In our case, the number of rule and schema populations is set for 30. And the mutation probability of rule is 0.2, the crossover and mutation probability of schema populations are set for 0.5 and 0.02, respectively. Fig.3 shows the resulting fitness compared with SGA.

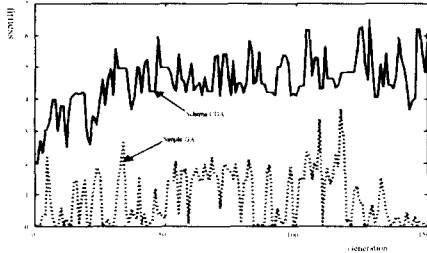


Fig.3. Rule Fitness

The obtained rules after 150 generations are stated in table 2. This rule base means that "Turn to the goal direction, and if an obstacle exist in the direction of moving then turn left or right although opposite direction to the goal position.

Table 2. Rule base after 150 generations

R1	: IF S1 is VL. THEN φ is NL and ν is ME.
R2	: IF S0 is VS. S1 is MS. and S2 is ME. THEN φ is PS and ν is VS.
R3	: IF S0 is ME, and S2 is VL. THEN φ is ZE and ν is VS.
R4	: IF θ is NS, S1 is VL. and S2 is ML, THEN φ is PS and ν is VS.
R5	: IF S0 is VL. and S1 is VS. THEN φ is PL and ν is VL.
R6	: IF S1 is ME. THEN φ is NL and ν is ML.
R7	: IF S0 is ME. S1 is ME. and S2 is ML. THEN φ is ZE and ν is VL.
R8	: IF θ is NL. S0 is ML. and S2 is ML. THEN φ is NS and ν is VS.
R9	: IF S0 is VS. and S2 is ME. THEN φ is ZE and ν is ME.
R10	: IF θ is PS. S0 is ME. and S1 is ME. THEN φ is PS and ν is MS.

VI. Conclusions

In this paper we proposed a Schema-coevolution method to design the rule base of FLC. By applying the proposed method to an optimal path planning problem where moving obstacle exist, the effectiveness of the proposed method was shown. The idea of Schema-coevolution is based on the Schema Theorem and Building Block Hypothesis and on the host-parasite coevolution. The individual of host-population(Rule) is parasitized by a schema in parasite population. By this process, useful schema generates much more instances in Rule populations at the next generation. Also it gives much more chance to find global optima than SGA because the parasite-population searches the schema space.

Acknowledgments

This work was supported by a grant No. KOSEF 96-01-02-13-01-3 from Korea Science and Engineering Foundation.

References

1. Seth G. Bullock, "Co-evolutionary Design: Implications for Evolutionary Robotics," *The 3rd European Conference on Artificial Life*, 1995.
2. W. Daniel Hillis, "Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure," *Artificial Life II*, Vol. X, pp.313-324, 1991.
3. Jan Paredis, "Co-evolutionary Computation," *Artificial Life*, Vol. 2, No. 4, pp.355-375, 1995.
4. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Third Edition, Springer-Verlag, pp. 265-281, 1995.
5. K.B. Sim, H.B. Jun, "Co-Evolutionary Algorithm and Extended Schema Theorem" *J.KSLAM*, vol. 2, no. 1, 1998
6. Melanie Mitchell, *An Introduction to Genetic Algorithms*, A Bradford Book, The MIT Press, 1996