The Third AFSS(1998). 601-606

Co-Evolution of Fuzzy Rules and Membership Functions

Hyo-Byung Jun, Chi-Sun Joung, 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
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. Futhermore proper fuzzy partitioning is not deterministic and there is no unique solution. So we propose a co-evolutionary method finding optimal fuzzy rules and proper fuzzy membership functions at the same time. Predator-Prey co-evolution and symbiotic co-evolution algorithms, typical approaching methods to co-evolution, are reviewed, and dynamic fitness landscape associated with co-evolution is explained. Our algorithm is that after constructing two population groups made up of rule base and membership function, 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.

Keywords : Co-evolution Algorithms, Fuzzy Logic Controller, Autonomous Mobile Robot, Path Planning

1. 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.

Generally co-evolution algorithms can be classified into two categories, which are predator-prey co-evolution[1] and symbiotic co-evolution[2][3]. And the authors derived a schema theorem associated with symbiotic co-evolution[4], and a new fitness measure in co-evolution is discussed in terms of "Red Queen effect"[5].

In this paper, we propose a co-evolution method generating optimal fuzzy controller, where the fitness of a population changes according to the evolution process of the other population. We presents how to extract fuzzy rules and generate membership functions at the same time using co-evolution scheme. 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 and partition membership functions by co-evolving the shape of membership functions and fuzzy rules.

The process of co-evolution is divided into two parts. The first one generates fuzzy rules and the other part determines proper shapes of membership functions. In the part of generating fuzzy rules, each individual comprises a set of rules. There are sets of rules in the population. Both each set of rules and the shapes of membership functions are expressed by genotype. The genetic operators such as selection, crossover and mutation are applied to each chromosome to generate new rule sets and membership functions. 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.

In the next section, co-evolution algorithms are reviewed, and in section 3, we explain how to construct fuzzy logic controller(FLC) using co-evolution algorithms. Simulation conditions and some results are described in section 4, and finally conclusions are followed.

2. Co-Evolution Algorithm

Recently evolutionary algorithms, including genetic algorithms(GAs), evolutionary strategies(ES), evolution -ary 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

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[3] proposed this concept with a problem of finding minimal sorting network for a given number of data. And co-evolution between neural networks and training data was proposed in the concept of predator and prey[6].

And fitness measure in co-evolution is studied in terms of dynamic fitness landscape. L. van Valen, a biologist, has suggested that the 'Red Queen effect' arising from co-evolutionary arms races has been a prime source of evolutionary innovations and adaptations[5]. This means that the fitness of one species changes depending on the other species's.

2.2 Symbiotic Co-Evolution

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[2] 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. A permutation is represented as a vector that describes a reordering of solution genes. 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[8].

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. We have showed mathematically the effectiveness of host-parasite co-evolution by the schema theorem associated with host-parasite co-evolution[4].

3. Co-Evolutionary Construction of FLC

Basically fuzzy logic controller is composed of fuzzifier, inference engine, rule base, and defuzzifier. There are three types of fuzzy reasoning, the first is Mamdani's minimum fuzzy implication rule, the second is Tsukamoto's method with linguistic terms as monotonic membership functions and the third is that the consequent of a rule is a function of input linguistic variables. We use the Mamdani's fuzzy implication rule, max-min compositional rule of inference. A rule is expressed qualitatively and linguistically by fuzzy IF-THEN rules. If there are m input variables and n fuzzy rules, then general fuzzy production rule is

$$R_i : \text{IF } x_1 \text{ is } A_{i1}, x_2 \text{ is } A_{i2}, \cdots, x_m \text{ is } A_{im} \text{ THEN y is } B_i \quad (1)$$

where A_{ik} is a linguistic term associated input variable x_k, and B_i is a linguistic term of output y.

Therefore reasoning value y is as follows:

$$B^0(y) = \bigvee_{i=1}^{n} [\omega_i \wedge B_i(y)]$$
$$\omega_i = A_{i1}(x_1^0) \wedge A_{i2}(x_2^0) \wedge \cdots \wedge A_{im}(x_m^0) \quad (2)$$

And using the center of area defuzzification method, the final inferred consequent y0 is given by

$$y^0 = \frac{\int B^0(y) \cdot y \, dy}{\int B^0(y) \, dy}. \quad (3)$$

A rule base is typically acquired via expert's knowledge. But it is very difficult to find fuzzy rules by hand when the input-output variables are going to increase. It is even impassible when the complex and dynamic environment is considered. And the proper fuzzy partitioning of input and output spaces plays an essential role in achieving a successful fuzzy logic inference engine design. But unfortunately, it is not deterministic and there is no unique solution.

Therefore, automatically generation of an optimal fuzzy rule base and proper fuzzy partitioning is considered as important and a lot of approaches were proposed. Especially there has been a growing interest in genetic based machine learning(GBML) system, in other words classifier system. There are two competing approaches to GBML[7]. One is called Michigan approach, which uses a single set of production rules or classifiers. So each individual rule has a strength which indicates the utility of the rules to the goal of the system. The other is called Pitt approach, the individual of which consists of a set of rules[8][9].

This paper presents a new approach to automatic generation of FLC based on the concept of co-evolution algorithms. Our approach has two parallel evolution processes which are rule base population and membership function(MF) population. The overview of our approach is illustrated in Fig.1.

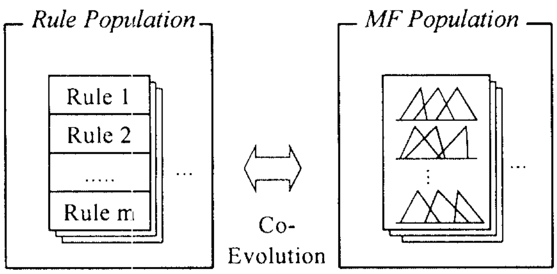
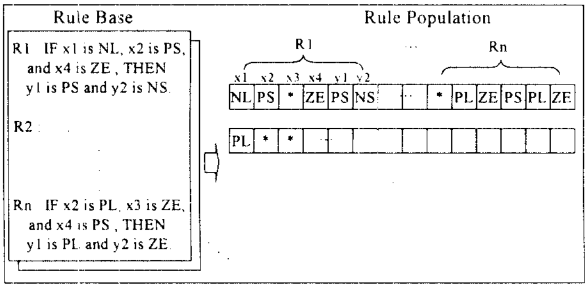


Fig. 1 A block diagram of co-evolution

To apply evolutionary algorithms to any problem, first the solution spaces should be represented by a chromosome. For our case, the encoding methods and genetic operators are explained in the following sub-sections.

3.1 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. 2.



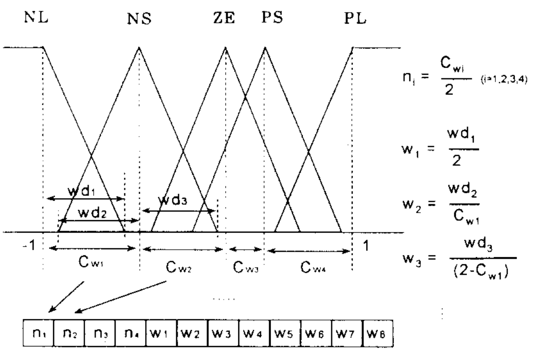
PL : Positive Large, PS : Positive Small, ZE : Zero,
NS : Negative Small, NL : Negative Large, * : Don't-Care

Fig. 2 An Example of fuzzy rules encoding scheme

In order to ensure the character preservingness, we use the $(\mu+\lambda)$-ES selection method and a mutation operator only as genetic operators. This selection method is elitist and therefore guarantees a monotonically improving performance.

3.2 Membership Function Population

As shown in Fig. 3, we use the normalized membership function partitioned with five terms. And the shape of each term is triangular except the two marginal terms. To find the optimal partitions we use the Genetic Algorithms(GAs) proposed by J. Holland in 1975. For our case, the encoding method is illustrated in Fig. 3. The triangular membership function's shape is determined by the three points that are a center point and left/right width points. We assume that the NL and PL terms have fixed center points and the other three center points could be placed any position from -1 to 1 and all the left/right width of each terms could be from 0 to the maximum value from its center point to the margin.



NL : Negative Large, NS : Negative Smal
ZE : Zero, PS : Positive Small
PL : Positive Large

Fig. 3 Membership function and encoding scheme

For a variable the chromosome is consist of (number of terms - 1)×3 bits real-valued string, where the first 4 bits represent the width proportion between the neighbor center points and the last 8 bits represent the width ratio of each term's left and right margin from its center point. For example, w_3 representing NS term's right width ratio is current right width(wd_3) over its possible maximum width($2-C_{w1}$). If there are N terms, N_i input variables, and N_o output variables, then the whole length of one chromosome becomes $3\times(N-1)\times(N_i+N_o)$ bits[10].

This encoding method guarantees the completeness, soundness, and non-redundancy between the solution and the genotype spaces. And fitness proportionate reproduction method and as genetic operators crossover and mutation are used.

4. 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 a 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 coner. Each robot's goal position is set to the other robot's starting point and perceives the other robot as a obstacle. A robot has three sensors(S0,S1,S2) 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(υ) of AMR. The simulation environmental conditions are set as follow:

- Working area : 1500×1500mm
- Robot Size : radius 25mm
- Number of robots : 2 units
- Maximum speed : 30mm/step
- Sensing Radius : 200mm
- Maximum steering angle : 90°/step

And the input/output variables' ranges are restricted as shown in table 1. Fig. 4 shows the AMR's sensor configuration and situations of detecting an obstacle.

Table 1. Range of input/output variables

INPUT				OUTPUT	
θ	S0	S1	S2	φ	υ
-180° ~180°	0~200 mm	0~200 mm	0~200 mm	-90° ~ 90°	0~30 mm

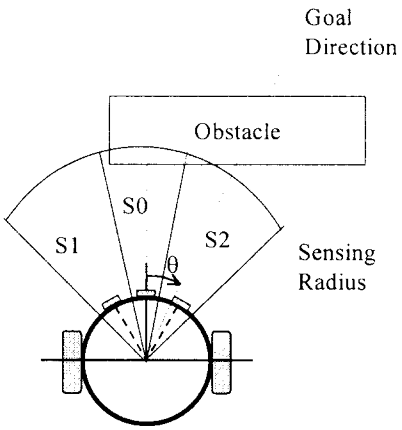


Fig. 4 Sensor Configuration

And the raw fitness measure is formulated by,

$$fit_{ENV} = \left(1-\frac{D_r}{D_G}\right)\cdot\frac{T_{\min}}{T}\cdot\frac{(N_N-N_n)}{N_N} \tag{4}$$

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. The fitness functions of membership function and rule base are set by,

$$fit^j_{MF} = \frac{1}{N}\sum_{i=1}^{N} fit^{ji}_{ENV} \tag{5}$$

$$fit^i_R = \frac{1}{M}\sum_{j=1}^{M} fit^{ij}_{ENV} \tag{6}$$

where N is the total number of MF individuals, fit^j_{MF} is j-th MF individual's fitness, M is the total number of rule base individuals, and fit^i_R is i-th rule base individual's fitness.

In our case, the number of rule and membership function populations is set for 50. And the mutation probability of rule is 0.2, the crossover and mutation probability of membership function are set for 0.5 and 0.02, respectively.

Fig. 5 shows the resulting fitness changes versus generations. And Fig. 6 shows the membership functions obtained after 400 generations

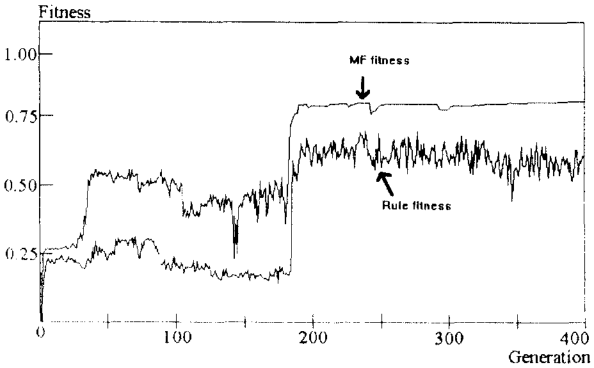
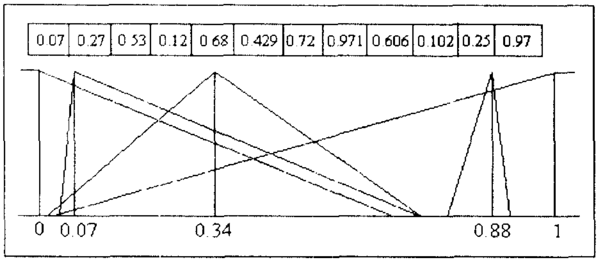
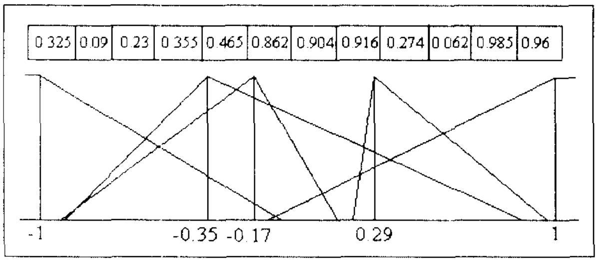


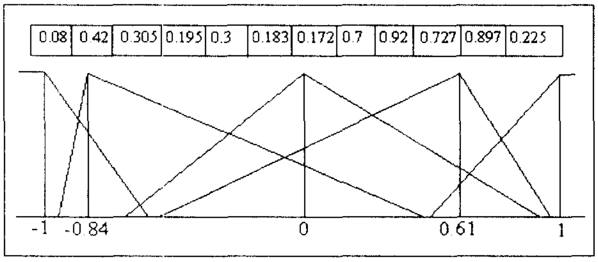
Fig. 5 Fitness curves



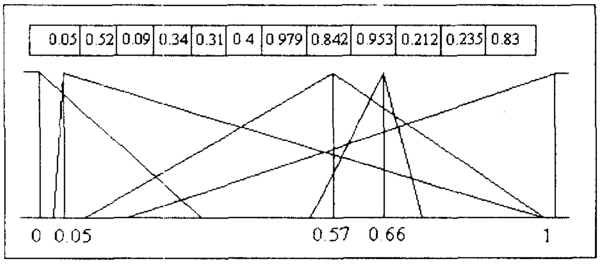
(a) Input variable θ



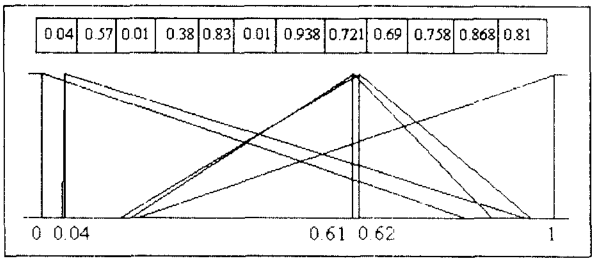
(b) Input variable *S0*



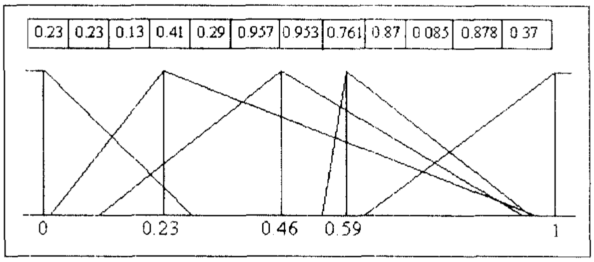
(c) Input variable *S1*



(d) Input variable *S2*



(e) Output variable φ



(f) Output variable υ

Fig. 6 Evolved membership functions

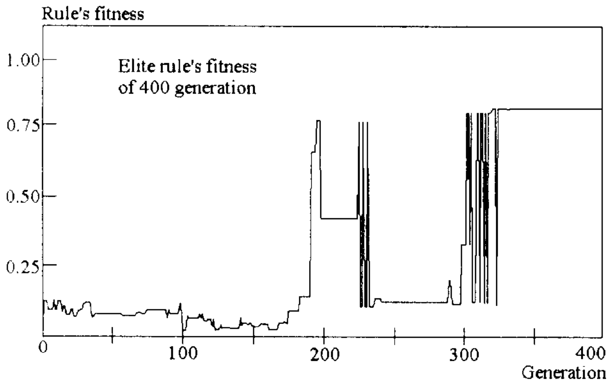
And the obtained rules after 400 generations are stated in table 2. This rule base can cover all possible states and means that 'Turn to the goal direction, and if an obstacle exist in the direction of moving then turn left or right although opposite direction to the goal position.

Table 2. Rule base after 400 generations

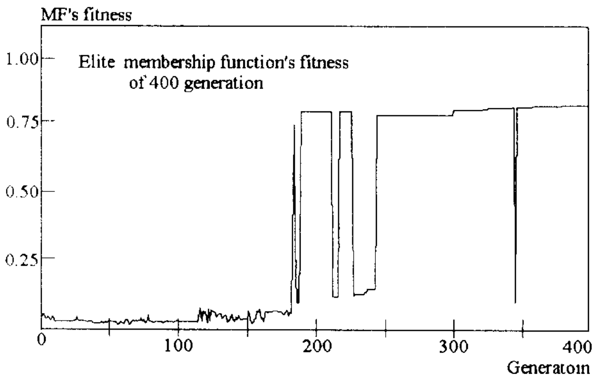
R1	: *IF S1 is VL, THEN* φ *is NL and* υ *is ME.*
R2	: *IF S0 is VS, S1 is MS, and S2 is ME, THEN* φ *is PS and* υ *is VS.*
R3	: *IF S0 is ME, and S2 is VL, THEN* φ *is ZE and* υ *is VS.*
R4	: *IF* θ *is NS, S1 is VL, and S2 is ML, THEN* φ *is PS and* υ *is VS.*
R5	: *IF S0 is VL, and S1 is VS, THEN*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

where NL is Negative Large, NS is Negative Small, ZE is Zero, PS is Positive Small, PL is Positive Large, and VS represents Very Small, MS is Medium Small, ME is MEdium, ML is Medium Large, and VL is Very Large.

Fig. 7 shows the dynamic fitness landscapes, where figure (a) illustrates elite rule's fitness changes according to the changes of membership functions' generations, and figure (b) illustrates fixed elite membership function's fitness changes versus the changes of rules' generation.



(a) Fitness landscape of 400 generation elite rule



(b) Fitness landscape of 400 generation elite MF

Fig. 7 Dynamic fitness landscapes

Using above rule base and membership functions both AMRs find their goal positions in relatively short time and avoid obstacles successfully

5. Conclusions

This paper has proposed a new approach to automatically fuzzy logic controller generation using co-evolution concept. By applying the proposed method to a optimal path planning problem where moving obstacle exit, the effectiveness of the proposed method was shown. And the concept of co-evolution is reviewed on the points of artificial life computation model.

Two main process in co-evolution are optimal rule base generation and proper fuzzy membership function partitioning. Each population evolves according to the other's evolution process. This evolution model is considered as more analogous to natural system.

However, the relation between fitness functions should be extended more generally when more than one population evolve. That remains the future work.

Acknowledgments

This research was supported by Korea Science and Engineering Foundation(KOSEF : 96-0102-13-01 -3)

References

1. Seth G. Bullock, "Co-evolutionary Design: Implications for Evolutionary Robotics," *The 3rd European Conference on Artificial Life*, 1995.
2. Jan Paredis, "Co-evolutionary Computation," *Artificial Life*, Vol. 2, No. 4, pp.355-375, 1995.
3. W. Daniel Hillis, "Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure," *Artificial Life II*, Vol. X, pp.313-324, 1991.
4. H.B. Jun, D.J. Kim, K.B. Sim, "Structure Optimization of Neural Network using Co-evolution," *Journal of the KITE*, Vol. 35, No. 4, 1997.
5. D. Cliff, G. F. Miller, "Tracking The Red Queen: Measurements of adaptive progress in co-evolutionary simulations," *COGS Technical Report CSRP363*, University of Sussex, 1995.
6. D.W. Lee, H.B. Jun, K.W. Sim, "A Co-Evolutionary Approach for Learning and Structure Search of Neural Networks," *Proc. of KFIS Fall Conference* '97, Vol. 7, No. 2, pp. 111-114, 1997.
7. T. Furuhashi, K. Nakaoka, Y. Uchikawa, " An Efficient finding of Fuzzy Rules Using a New Approach to Genetic Based Machine Learning," *FUZZ-IEEE/IFES* '95, pp. 715-722, 1995.
8. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Third Edition, Springer-Verlag, pp. 265-281, 1995.
9. J. Kinzel, F. Klawonn, R. Kruse, " Modifications of Genetic Algorithm for Designing and Optimizing Fuzzy Controllers," *ICEC* '94, Vol. 1, pp. 28-33, 1994.
10. H.B.Jun, K.W.Sim, "Fuzzy Inference-based Reinforcement Learning of Dynamic Recurrent Neural Networks," *Journal of KFIS*, Vol. 7, No. 5, pp. 60-66, 1997.