Integrated Computer-Aided Engineering 10 (2003) 369–385
IOS Press

A quantitative analysis of evolvability for an evolutionary fuzzy logic controller

Seung-Ik Lee[a,b,*] and Sung-Bae Cho[b]
[a]*ATR International, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan*
Tel.: +81 774 95 2687; Fax: +81 774 95 2647
[b]*Department of Computer Science, Yonsei University, 134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, South Korea*
Tel.: +82 2123 2720; Fax: +82 365 2579; E-mail: cypher@candy.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. This paper presents a quantitative analysis of evolvability with evolutionary activity statistics in an evolutionary fuzzy system. In general, one can estimate the performance of an evolved fuzzy controller by its fitness. However, it is difficult to explain how its fitness or adaptability has been obtained. Evolutionary activity is used to measure the evolvability of fuzzy rules and explain why salient rules have higher evolvability. A genetic algorithm is used to construct a fuzzy logic controller for a mobile robot in simulation environments. The quantitative analysis shows that sufficient evolvability is maintained during the evolution and that it contributes to the construction of the optimal controller.

1. Introduction

Fuzzy logic controllers (FLC) [5,6,8,11–13,15,16, 22,26] have been widely used for behavior-based robots such as Khepera because they can easily transform linguistic information and expert knowledge into control signals. While fuzzy logic control has many advantages over traditional methods, it also has some drawbacks at the design stage in that it is difficult to determine optimal parameters. Therefore, many researchers have applied evolutionary algorithms to the construction of FLCs to automate the procedure of determining the parameters [6,7,11,16,22]. While many of the previous works have shown successful results for the problems at hand, no in-depth analysis on the role of evolutionary computation has been done yet. Because the good performance of an evolved solution does not necessarily mean that the evolution has been adaptive [1,4], chances are that subsequent evolution under the same condition would not reproduce the same or similar good performance. Therefore, it is necessary to prove or illustrate that a certain level of adaptability has been maintained during the evolution such that the good performance of the evolved solution is not from chance or necessity but from adaptive evolution.

This paper aims to show the role of evolutionary computation in finding successful rules for solving problems using a measure of evolutionary activity proposed by Bedau and Packard [1,4]. Although adaptive evolution is known to make complex functional structures [10,24], it is, in general, difficult to differentiate adaptive change from other evolutionary phenomena such as chance and necessity. Evolutionary activity is used to quantify the evolvability of an evolution in terms of adaptability so that it can show the adaptability of a system in an objective, quantitative manner. In this paper, we will quantitatively show that the evolved fuzzy controller is not the result of other evolutionary phenomena but the result of adaptive evolution by applying evolutionary statistics. Furthermore, we will analyze the behavioral properties during the evolution that are regarded to be more adaptive than the others.

The organization of this paper is as follows. In Section 2, a method for quantifying adaptive evolution called *evolutionary activity statistics* is described. In Section 3, an approach to evolving a fuzzy logic con-

*Corresponding author.

ISSN 1069-2509/03/$8.00

troller is explained. Section 4 describes experimental set-ups and shows simulation results of the evolved controller. Section 5 presents the quantification of the evolution of the controller and analyzes salient fuzzy rules that are regarded as having higher adaptability. Finally, we will conclude in Section 6.

2. Evolutionary activity sattistics

The goal of evolutionary systems is to find the best solutions to solve the problem at hand. Although these solutions have been proven to work well in applications in many areas, no proof has been shown that the best solution is the result of adaptation to the given problem. As there are many evolutionary phenomena, such as chance or necessity, a good solution does not necessarily mean that the evolution has maintained sufficient adaptability.

Adaptability shows different characteristics compared with other genetic phenomena. Adaptive components in evolutionary systems continuously generate innovative functional structures and these structures persist in the systems because they have high adaptive behavioral characteristics. Based on this idea, Bedau proposed a new measure, called *evolutionary activity* [1].

Evolutionary activity serves to quantify the degree to which a system exhibits the continual spontaneous generation of adaptive forms. Specifically, it measures the degree to which new genetic combinations are persistently used in a population. This quantity is an objective, empirical measure of the level of evolutionary activity in an artificial or natural system. Bedau and Packard [1] proposed an approach to quantifying adaptive phenomena in evolutionary systems. The method for quantifying adaptive evolution is based on evolutionary activity statistics that can be computed from the records of all components' birth, life, and death. The statistics have been applied to, for example, visualizing adaptive evolutionary phenomena [1,3], studying punctuated equilibrium dynamics in evolution [23], identifying long-term evolutionary trends [2,18], and classifying evolutionary dynamics [4].

To measure evolutionary activity, a counter is attached to each component. A component is a unit that can play a significant role in a given problem. A component, therefore, can be at different levels depending on the problem at hand. For example, in a system where each genotype is contending for space over time, a genotype's capability or adaptability can be measured by the amount of space it occupies. In this case, the components are the genotypes and counters are attached to them [2,3]. Individual alleles or taxonomic families can also be components depending on the problem [1, 2].

A counter, $a_i(t)$, of the ith component at time t is attached to each component of a system. A component's activity changes over time as in Eq. (1).

$$a_i(t) = \sum_{k \leqq t} \Delta_i(k) \tag{1}$$

where $\Delta_i(k)$ is the activity increment for component i at time k. Various activity increment functions $\Delta_i(t)$ can be used, depending on the nature of the components and the purposes at hand [1,3,4]. One could increment a component's activity with its concentration in a system [3] or the extent to which it is used or expressed [1], if the concentration or extent is correlated with its adaptive value.

Various statistics can be defined based on the components in a system and their activity counters. The number of components with activity > 0 at time t is *Diversity*, $D(t)$, defined by Eq. (2).

$$D(t) = \#\{i : a_i(t) > 0\} \tag{2}$$

where $\#$ denotes set cardinality. The number of components with activity a at time t is *component activity distribution*, $C(t, a)$, defined by Eq. (3).

$$C(t, a) = \sum_i \delta(a - a_i(t)) \tag{3}$$

where $\delta(a - a_i(t))$ is the Dirac delta function, equal to one if $a = a_i(t)$ and zero otherwise. To measure the continual adaptive success of the components in a system at a give time, *total cumulative evolutionary activity* or *total activity* in short, $A_{cum}(t)$, is defined by Eq. (4).

$$A_{cum}(t) = \sum_i a_i(t) \rightarrow \int_0^\infty aC(t, a)da. \tag{4}$$

Dividing the cumulative evolutionary activity by $D(t)$ gives the cumulative activity per component, called *mean cumulative evolutionary activity* or *mean activity* in short, as follows:

$$\overline{A}_{cum}(t) = \frac{A_{cum}(t)}{D(t)}. \tag{5}$$

Activity statistics can clearly reflect a system's adaptive evolutionary dynamics after they have been normalized by a "neutral" model [4], to screen off the contribution of non-adaptive or maladaptive genotypes.

Adaptive innovations correspond to new components flowing into the system and proving their adaptive value through their persistent activity. Let a_0 and a_1 define a strip through the component activity distribution function, $C(t, a)$, such that activity values a in the range $a_0 \leqq a \leqq a_1$ are among the lowest activity values that can be interpreted as evidence that a component has positive adaptive significance. Then, *new evolutionary activity* or *new activity* in short, $A_{new}(t)$, is the reflection of the evolutionary adaptive innovations by summing the activity per component with values between a_0 and a_1 as:

$$A_{new}(t) = \frac{1}{D(t)} \sum_{a=a_0}^{a_1} C(t, a) \rightarrow \frac{1}{D(t)} \int_{a_0}^{a_1} C(t, a) da. \tag{6}$$

3. Evolution of FLC parameters

A genetic algorithm (GA) is a search technique based on the mechanisms of natural selection and natural genetics [9]. This combines survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of strings is created using bits and pieces of the fittest of the old; an occasional new part is tried for good measure. While randomized, genetic algorithms are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance.

At first, a population of individuals that encode candidate solutions to a given problem is initialized at random. Each individual in the population is evaluated and changed by genetic operations such as crossover and mutation to reproduce a new population. This process goes on until a satisfactory individual appears in the population.

The code of an individual representing the parameters of an FLC is applied to the Khepera robot to measure fitness. After a certain period, each individual FLC is given a fitness value according to its performance in a given problem. Individuals with higher fitness are selected and genetic operations are applied to produce the next population of individuals. Two parameters should be determined to run a GA: how to encode the FLC parameters in gene code and how to estimate the fitness value of each individual. The encoding scheme first proposed in [6] is modified and extended here so

Fig. 1. Encoding of FLC parameters.

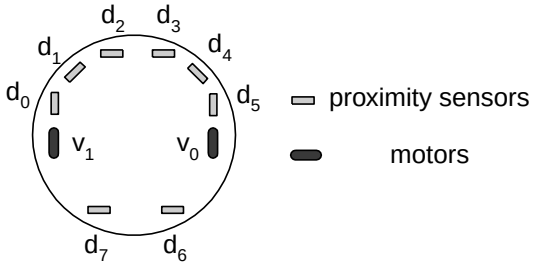


Fig. 2. Linguistic variables corresponding Khepera sensors and motors.

that it can tune the parameters more accurately and include more rules. For the FLC parameters, eight input variables, two output variables, and maximally twenty rules are encoded as shown in Fig. 1, where integer coding is used rather than conventional binary coding.

We first need to define fuzzy sets on both inputs from the sensors and outputs to the motors of the mobile robot. Our FLC uses the sensory information of eight proximity sensors as inputs and controls the speed of the two motors on Khepera named as shown in Fig. 2.

The input linguistic variable $d_i (0 \leqq d_i \leqq 1023, i = 0, \ldots, 7)$ and output linguistic variable $v_i (-10 \leqq v_i \leqq +10, i = 0, 1)$ are expressed by linguistic values (VF, F, M, C, VC) and (BF, B, S, F, FF), respectively. Note that d_i reads 0 when there is no obstacle and 1023 when obstacles are closest. Also note that v_0 corresponds to the right motor and v_1 corresponds to the left motor of the robot. The linguistic terms have the following meanings:

Input		Variable
VF	:	Very Far
F	:	Far
M	:	Medium
C	:	Close
VC	:	Very Close

Output		Variable
BF	:	Backward Fast
B	:	Backward
S	:	Stop
F	:	Forward
FF	:	Forward Fast

The membership functions of D_i and V_i are all in a triangular form defined by Eq. (8).

$$\begin{aligned}&\text{triangle}\,(x,a,b,c) = \\ &\max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right)\end{aligned} \tag{7}$$

where the parameters $\{a,b,c\}$ with $a \leqslant b \leqslant c$ determine the x coordinates of the three corners of the underlying triangular membership function. To reduce the computational complexity, some restrictions are applied to each membership function by Eq. (8).

$$\begin{aligned}\mu_{d_i,VF}(d_i) &= \text{triangle}(d_i,0,0,c^F_{d_i}),\\ \mu_{d_i,F}(d_i) &= \text{triangle}(d_i,0,c^F_{d_i},c^M_{d_i}),\\ \mu_{d_i,M}(d_i) &= \text{triangle}(d_i,c^F_{d_i},c^M_{d_i},c^C_{d_i}),\\ \mu_{d_i,C}(d_i) &= \text{triangle}(d_i,c^M_{d_i},c^C_{d_i},1023),\\ \mu_{d_i,VC}(d_i) &= \text{triangle}(d_i,c^C_{d_i},1023,1023),\\ \mu_{v_i,BH}(v_i) &= \text{triangle}(v_i,-10,-10,c^B_{v_i}),\\ \mu_{v_i,B}(v_i) &= \text{triangle}(v_i,-10,c^B_{v_i},c^S_{v_i}),\\ \mu_{v_i,S}(v_i) &= \text{triangle}(v_i,c^B_{v_i},c^C_{v_i},c^F_{v_i}),\\ \mu_{v_i,F}(v_i) &= \text{triangle}(v_i,c^S_{v_i},c^F_{v_i},+10),\\ \mu_{v_i,FH}(v_i) &= \text{triangle}(v_i,c^F_{v_i},+10,+10)\end{aligned} \tag{8}$$

where $0 < c^F_{d_i} < c^M_{d_i} < c^C_{d_i} < 1023$ and $-10 < c^B_{v_i} < c^S_{v_i} < c^F_{v_i} < +10$. $c^F_{d_i}$, $c^M_{d_i}$, and $c^C_{d_i}$ are three of $c^k_{d_i}, k = 1,\ldots,19$, evenly distributed in the universe of discourse U_d. $c^B_{v_i}$, $c^S_{v_i}$, and $c^F_{v_i}$ are three of $c^k_{v_i}, k = 1,\ldots,19$, evenly distributed in the universe of discourse U_v as shown in Fig. 3. Therefore, only three of the five fuzzy membership functions, $c^F_{d_i}$, $c^M_{d_i}$, and $c^C_{d_i}$, need to be encoded in the case of input and $c^B_{v_i}$, $c^S_{v_i}$, and $c^F_{v_i}$ in the case of output. The membership functions are not symmetric because, in our view, evolution would make them symmetric if that is necessary.

Each rule has eight input variables, $d_0,\ldots,d_7$, and two output variables, v_0 and v_1. Variables having the toggle bit "1" participate in the conditional part in a fuzzy rule. Output variables do not have a toggle flag because all of them should appear in the consequent part. The first bit in Fig. 4 designates whether this rule participates in the fuzzy inference process. Twenty rules at maximum might not completely cover the input space because the complete input space is that of $5^8 = 390,625$ rules. However, the wildcards for input variables dramatically reduce the search space and it is expected that evolution will lead the rules to adaptation so that all occurrences of actual input configurations in a given environment can be covered. Therefore, Fig. 4 can be decoded as follows:

$$\begin{aligned}&\text{IF } (d_0 = \text{M}) \text{ and } (d_1 = \text{VF})\\ &\text{and } (d_6 = \text{C}) \text{ and } (d_7 = \text{M})\\ &\text{THEN } (v_0 = \text{BF}) \text{ and } (v_1 = \text{FF})\end{aligned}$$

The fitness of an individual, Θ, with s run time steps and n checkpoints passed through is defined by Eq. (9).

$$\begin{aligned}\Theta = \left(\frac{n}{N}\right)&\sum_{t=0}^{s} V_t(1-\sqrt{D_t})\\ &(0.5+0.5(1-I_t))\end{aligned} \tag{9}$$

where N is the total number of checkpoints, V_t is average rotational speed at step t, D_t is the normalized absolute value of the difference between the speed of the two wheels, and I_t represents the normalized value of the sensor that presents the highest level of activation.

4. Experiments

4.1. Experimental setup

A genetic algorithm was used for the evolution of fuzzy rules. At the start of evolution, we generated and randomly initialized fifty individuals. Maximum generation was 1,000. The population was overlapped by 50% with elitist strategy. New individuals replaced individuals with relatively low fitness values. The crossover rate was 0.5, and the mutation rate was 0.01.

We also experimented with a neutral shadow of the fuzzy model to screen off the contribution of non-adaptive or maladaptive genotypes and to measure the excess evolutionary activity of genotypes [21]. All the parameters for the neutral shadow model were set identical to the fuzzy model except that the selections for reproduction occurred randomly regardless of the individuals' fitness. The evolutionary dynamics in a neutral shadow is a neutral diffusion process in genotype space. Genotypes arise and become extinct, and their concentrations change over time, but the genotype dynamics is at best weakly linked to adaptation. All selection in the shadow model is random so that no genotype has any adaptive significance.

As the number of whole rules in a population is constant, the number or ratio of the genotype of the rule in the population expresses a rulefls relative significance. Adaptive genotypes will have a relatively high concentration in the population, and poorly adapted genotypes

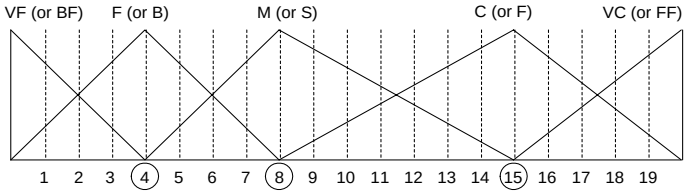


Fig. 3. Encoding of a membership function.

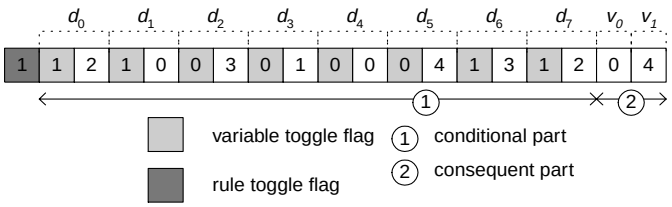


Fig. 4. Encoding of a rule.

will be correspondingly scarce. Therefore, the evolutionary activity $a_i(t)$ of the ith genotype at time t is defined as its instances integrated over the time from its origin up to t, provided it exists as follows:

$$a_i(t) = \begin{cases} \int_0^t n_i(t)dt & \text{if genotype } i \text{ exists at } t \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where $n_i(t)$ is the number of rules that has ith genotype at t.

Figure 5 shows the environment where each individual is evaluated. The environment is set up such that the mobile robot can meet many different situations, such as "left turn", "right turn", "narrow path", and "open area". Here, the goal of the mobile robot is to return to its starting position (depicted as • in Fig. 5) by moving along the lines shown in Fig. 5.

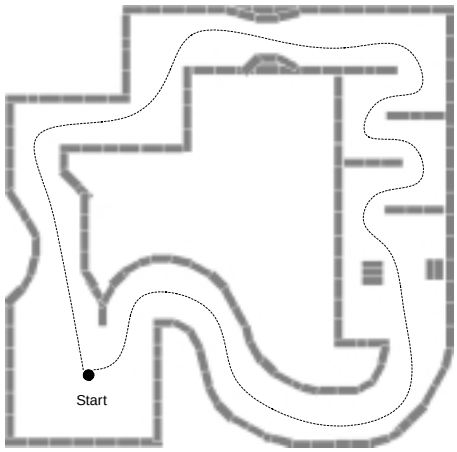


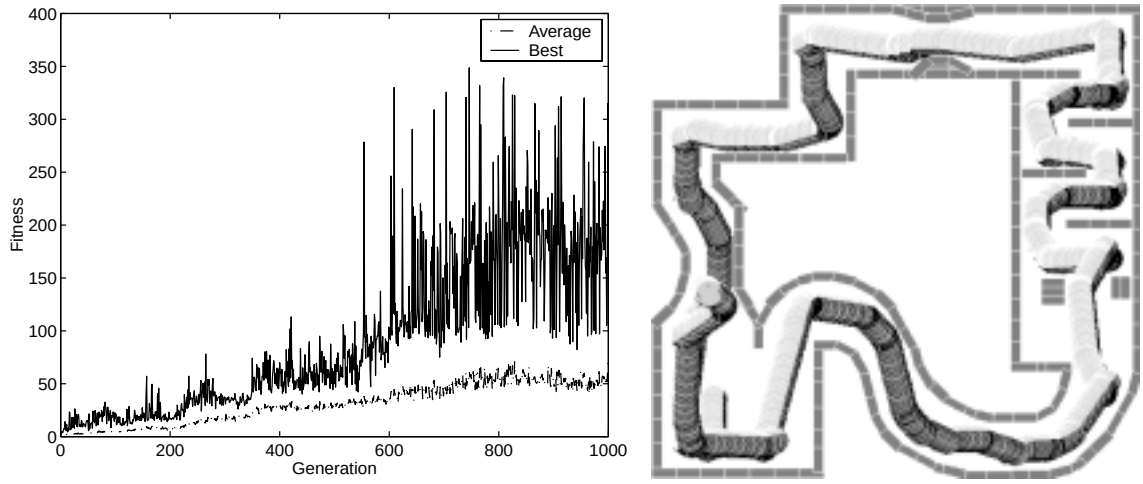
Fig. 5. Environment for mobile robot test.

4.2. Simulation results

Figure 6 (a) shows fitness changes of the simulation. Average fitness increases slowly and best fitness increases as generation goes by with some fluctuation. Even though elitist strategy was used in the evolution, best fitness fluctuates. The reason for this is that the sensory inputs have about 10% noisy information in order to make the simulation more realistic.

From the 554th generation, individuals with an extremely high fitness value start to show up and disappear. Among the individuals, the best individual reaching the goal position has appeared at generation 746. The movement of the best individual is shown in Fig. 6 (b). The best individual shows behaviors such as "left turn", "right turn", "passing through narrow corridor", and "moving in wide area", which we did not specify in the gene code.

The best individual shown in Fig. 6(b) has thirteen rules with its rule-toggle-bit on and only seven out of thirteen get activated during the simulation. These seven rules are shown in Table 1 and their associated fuzzy sets are shown in Fig. 7. Careful viewing shows that there are input combinations for which none of the rules has a degree of membership larger than zero. Fortunately, however, the seven rules prevent those combinations from occurring by driving the robot to follow the route shown in Fig. 6(b). We also have to say that the best individual sometimes behaved differently because other rules rather than the desired rules in a given input configuration were fired. This reduces the robustness of the controller, but does not interfere



(a) Fitness (b) Trajectory of the best individual

Fig. 6. (a) Fitness changes. (b) Snapshot of the best individual.

with measuring evolvability. During the steps shown in Fig. 6(b), the activations of the related rules are shown in Fig. 8. Rules BR_2, BR_4, and BR_5 are activated more frequently than the other rules.

5. Quantification of the evolutionary dynamics

5.1. Measurement of adaptability

In this section, we will show that the performance exhibited by the best individual is the result of adaptive evolution. To do this, we utilize the evolutionary activity concept of Bedau's proposal [1]. Figure 9 shows the component activity distribution function. This graph depicts how evolutionary activity of the genotypes (i.e., genotypes of rules) on the y-axis varies as a function of time on the x-axis. There are myriad lines or waves evident in Fig. 9. Each wave corresponds to a single genotype and shows the variation over time of that genotype's evolutionary activity.

Here, by the definition of evolutionary activity as given in Eq. (1), the slope of a given genotype's activity wave at a given time can be interpreted as the genotype's concentration in the population at that time because the whole number of rules in a population at a time is constant. When a new genotype enters the population, a new wave will arise from the x-axis. As the genotype's concentration in the population grows (or shrinks) over time, the slope of the wave increases (or decreases). When the genotype finally becomes extinct, the slope of its wave falls to zero and the wave ends. In this way, a genotype's activity wave reflects its changing concentration throughout its history in the population. Whenever one genotype drives another to extinction, a new wave arises as an earlier one dies out. The dominating rules during a given epoch of evolution appear as dominating wave(s).

Figure 10(a) shows the time series of diversity D. The neutral shadow's diversity values are generally higher than those of the fuzzy model. This arises from the fact that there is no selective pressure in the neutral shadow model in contrast to the fuzzy model, where more adaptive individuals have higher possibilities of being selected, therefore, producing more children, and persisting over time. Figure 10(b) shows the total cumulative evolutionary activity, A_{cum}, and Fig. 10(c) shows the mean cumulative evolutionary activity, $\overline{A}_{cum}$, of the two models. Both A_{cum} and $\overline{A}_{cum}$ are significantly higher in the fuzzy model than in the neutral shadow. This means that many more adaptive rules are present in the fuzzy model than its neutral shadow.

Figure 11 illustrates the difference between the fuzzy model and its neutral shadow. The distributions shown in Fig. 9 have been summed along the temporal dimension and then divided by the total number of counts in both distributions. The value of each distribution at a given activity value a represents the fraction of activity counts in each distribution that have activity a.

The fraction of activity counts of the neutral shadow at relatively lower activities is higher than that of the fuzzy model. However, at higher activities, the fraction of activity counts of the neutral shadow is much lower

Table 1
Rules of the best individual

Rule	Meaning
BR_1	(d_3 =F) and (d_4 =M) and (d_7 =F)→ (v_0 =F) and (v_1 =F)
BR_2	(d_7 =VF) → (v_0 =BF) and (v_1 =S)
BR_3	(d_1 =VC) and (d_2 =F) and (d_6 =F)→ (v_0 =BF) and (v_1 =B)
BR_4	(d_1 =VF) and (d_2 =VF)→ (v_0 =FF) and (v_1 =FF)
BR_5	(d_0 =VF) and (d_4 =VC) and (d_7 =VF)→ (v_0 =FF) and (v_1 =B)
BR_6	(d_0 =M) and (d_1 =C) and (d_3 =VF) and (d_4 =F) and (d_7 =VF) → (v_0 =S) and (v_1 =S)
BR_7	(d_1 =C) and (d_6 =M) →(v_0 =S) and (v_1 =B)

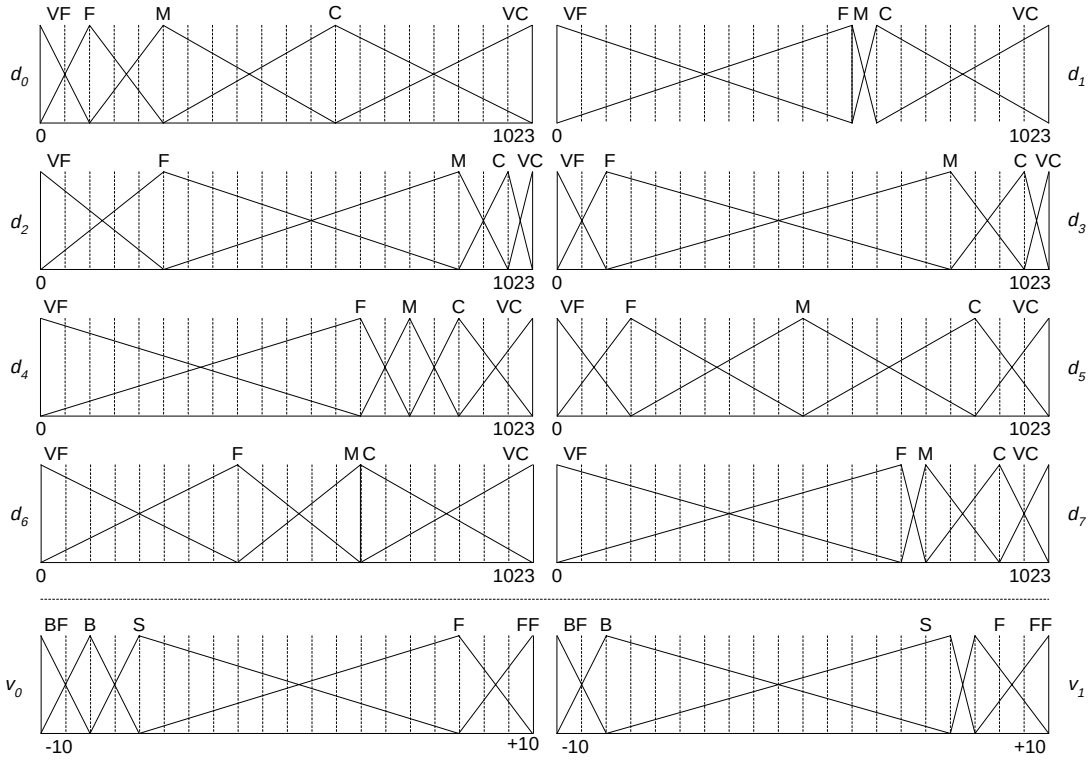


Fig. 7. Fuzzy sets obtained by evolution.

than that of the fuzzy model. This means that high adaptive rules are much more numerous in the fuzzy model than in its neutral shadow.

The point at which the two distributions have the same value, a', is an activity count that is equally likely to have been chosen from either distribution. This value is used to calculate new activity, A_{new}, with a_0 to be a' and a_1 slightly above a' as follows:

$$a_1 = a' + (0.05 \times (a_{\max} - a')) \tag{11}$$

where $a_{\max}$ is the highest activity value at which either distribution is positive and a' is the lowest value at which the two distributions cross. Since the lowest value at which two distributions cross in Fig. 11 is $a' = 145$, we set $a_0 = a' = 1.45 \times 10^1$ and $a_1 = 4.05 \times 10^1$, respectively.

With these values, we can calculate new activity as shown in Fig. 10(d). The new activity values of the fuzzy model are higher than those of the neutral shadow, which means that new activity signaling positive adaptability flows more frequently into the fuzzy model compared with that of the neutral shadow. Therefore, we can conclude from this figure that the fuzzy model continuously generates new adaptive fuzzy rules much more than the neutral shadow does.

5.2. Analysis of salient waves

Figure 12 shows salient waves that have generally higher activities than other waves. The birth and death of a genotype can be seen in this figure. For example,

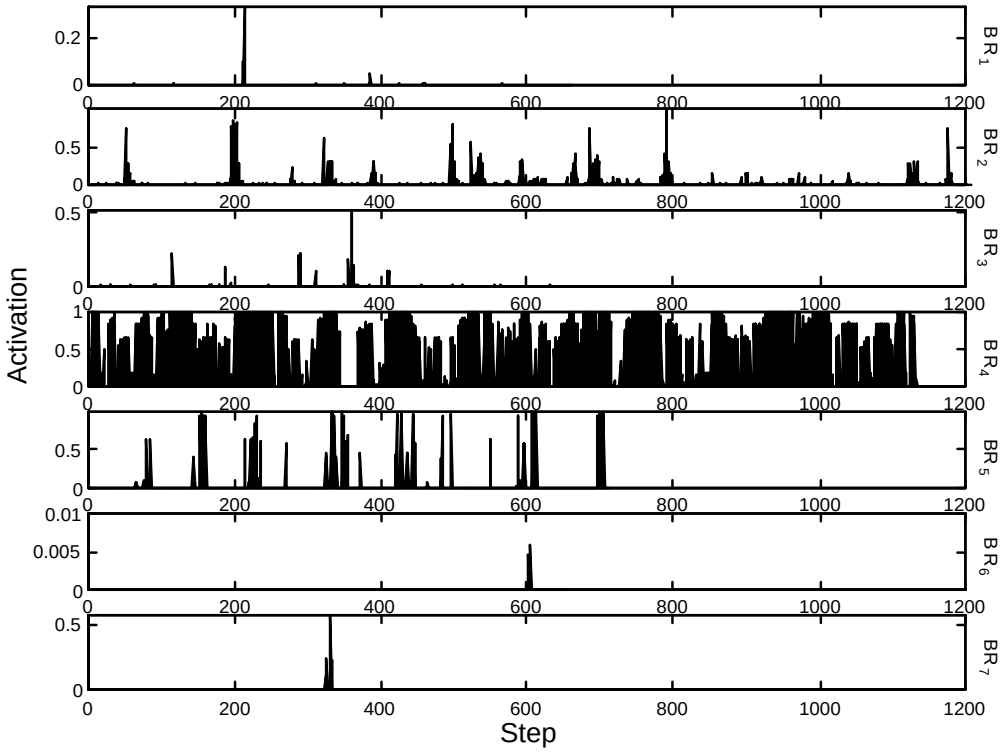


Fig. 8. Rule activations of the best individual.

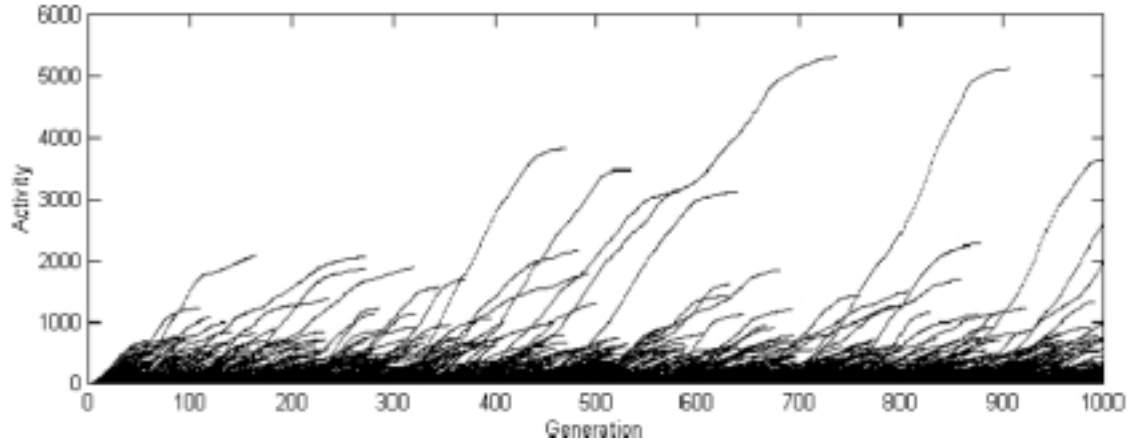
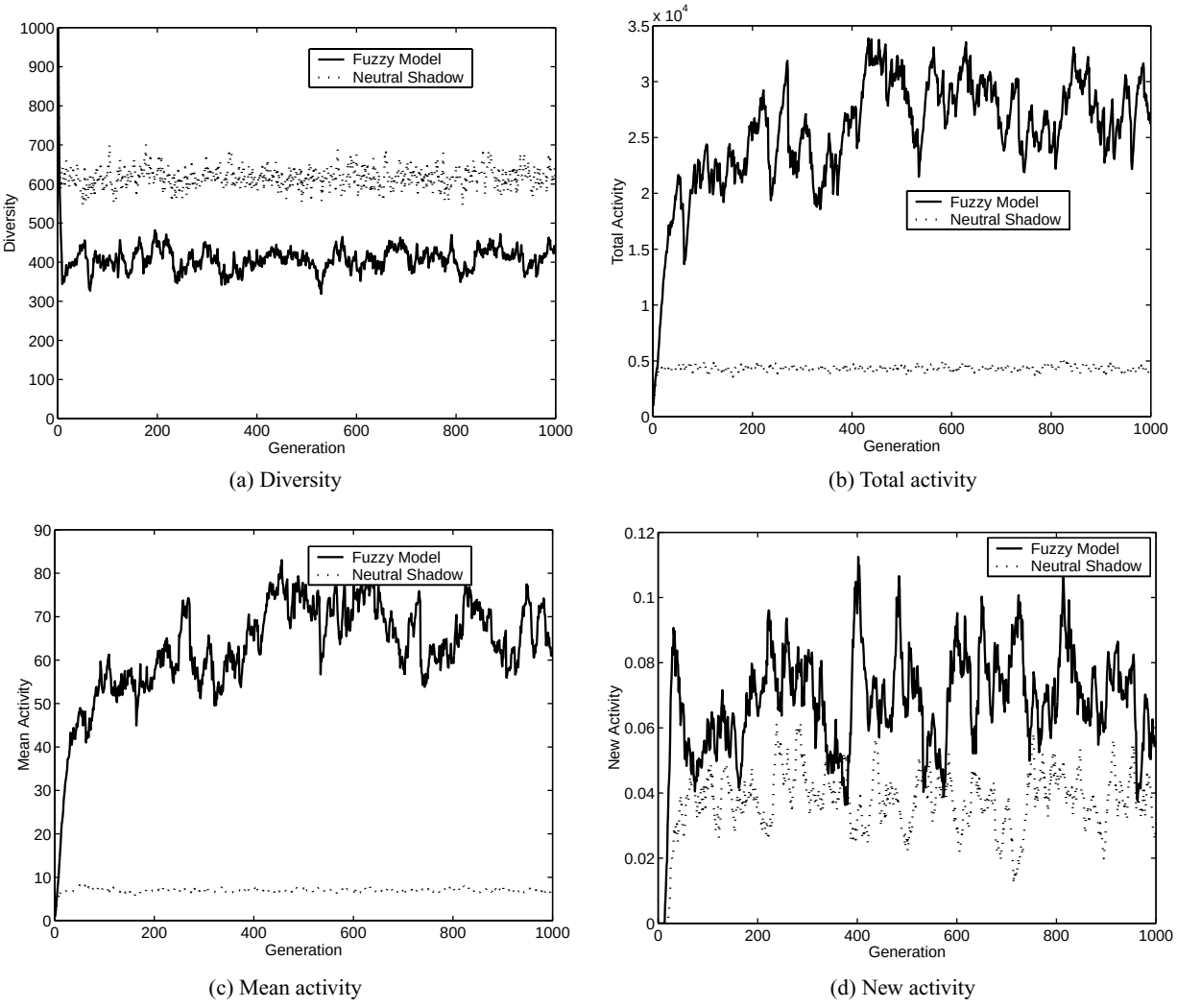


Fig. 9. Activity distribution function $C(t, a)$.

genotype SR_1 occurs before generation 50 and lasts for over 100 generations.

The salient genotypes in Fig. 12 have the rules listed in Table 2. Comparing the salient rules with the rules of the best individual in Table 1, very close similarities can be identified. For example, SR_2 is very similar to BR_4, SR_8 to BR_5, and SR_{10} to BR_2. Considering the close similarities between the salient rules and the rules of the best individual, it can be inferred that the salient rules contributed to the evolution of the best individual.

To find out what properties of the salient rules allow them to last long, some individuals that have salient rules as part of their rule set are selected and analyzed in subsequent sections.



(a) Diversity (b) Total activity

(c) Mean activity (d) New activity

Fig. 10. (a) Diversity $D(t)$. (b) Total cumulative evolutionary activity $A_{cum}(t)$. (c) Mean cumulative evolutionary activity $\overline{A}_{cum}(t)$. (d) New activity $A_{new}(t)$.

5.2.1. Analysis of salient rule SR_2

The SR_2 genotype occurs at about the 60th generation and lasts for about 200 generations. One individual with SR_2 was selected and analyzed. Table 3 shows the rules activated during this test. SR_2 is, in this case, R_{10} of this individual as in Table 3.

Figure 13(a) shows the behaviors with SR_2 included and Fig. 13(b) shows the behaviors with SR_2 intentionally excluded to find out the role of SR_2. Figure 13 indicates that SR_2 drives the robot to move forward. This can be identified by the fact the robot does not move anywhere in Fig. 13(b), whilst it moves forward at the beginning stage of the simulation in Fig. 13(a).

Figure 14(a) shows the speed changes of the two motors with SR_2 and Fig. 14(b) shows the speed changes without SR_2. The speeds of the left motor in Fig. 14(b) are the same as those of the right motor except that their signs are different. Therefore, the behaviors in Fig. 13(b) are actually turning continuously at the same position.

Figure 15 shows the activations of the rules of Table 3. In the case of (a), all the rules are activated during the test steps while only R_6 is activated in the case of (b). By R_6, the robot turns continuously at the same position as in Fig. 13(b). Although R_6 is still fired in the case of Fig. 15(a), the robot moves forward because the activation of R_6 is relatively much lower than that of $R_{10}(= SR_2)$ as shown in Fig. 15. From this analysis, we can conclude that SR_2 drives the robot to move forward when walls or obstacles are very far from it.

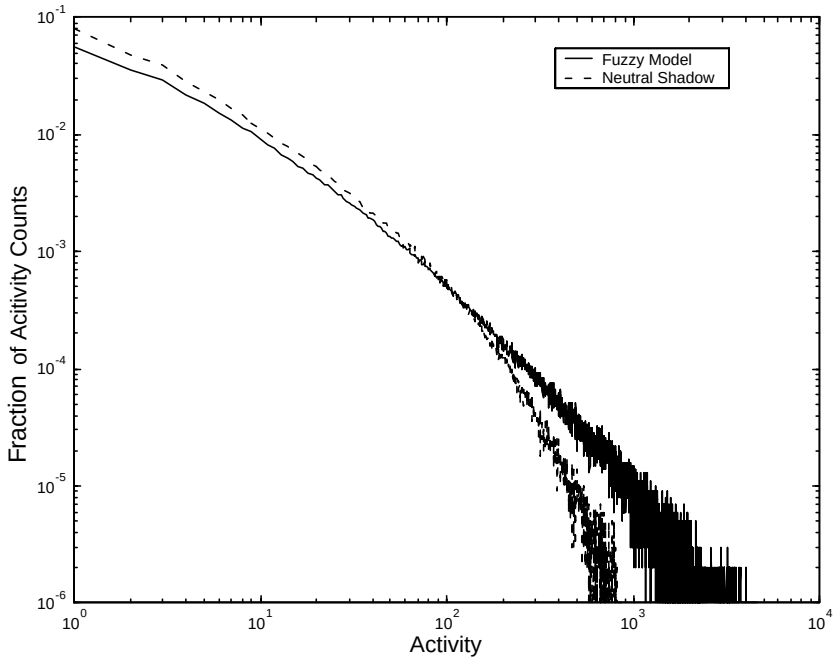


Fig. 11. Log-log plot of the component activity distributions for the fuzzy model and its neutral shadow.

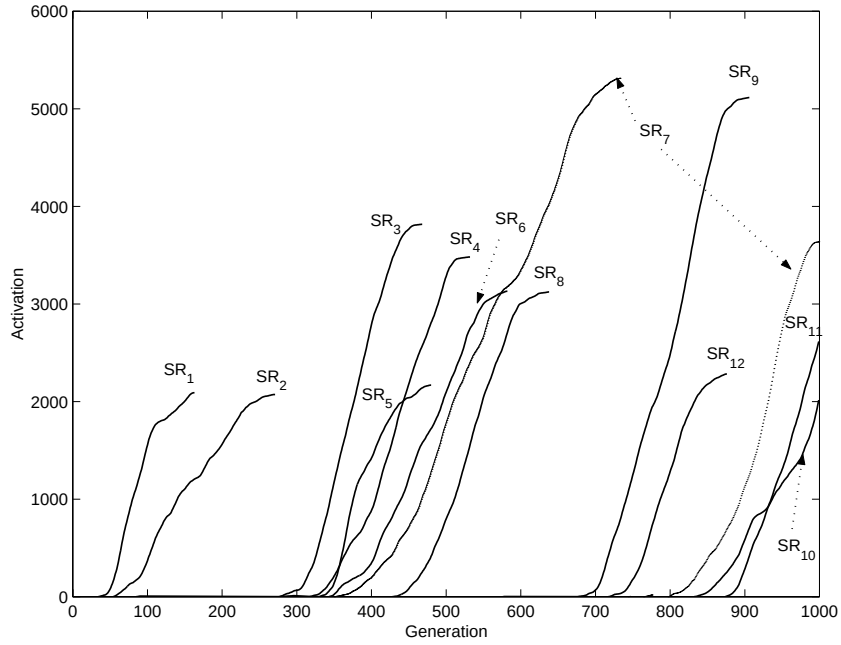


Fig. 12. Salient waves of evolutionary activities.

5.2.2. Analysis of salient rule SR_{10}

The SR_{10} genotype occurs at about the 830th generation and lasts until the last generation. One individual with SR_{10} was selected and analyzed. Table 4 shows the rules activated during this test. SR_{10}, is in this case, R_6 of this individual as in Table 4.

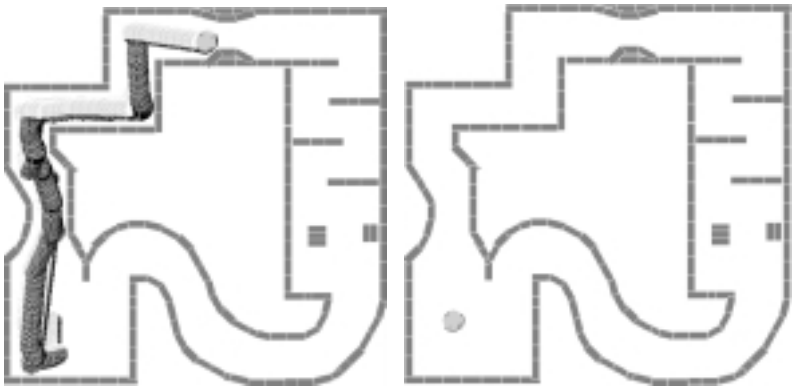
Figure 16 shows the individual's behaviors in two cases. Figure 16(a) shows the behaviors with SR_{10} included, and Fig. 16(b) shows the behaviors with SR_{10} intentionally excluded to find out the role of SR_{10}.

Table 2
Salient rules

Rule	Gene Code	Meaning
SR_1	1030202010010031113	(d_5 =VF) and (d_7 =F) → (v_0 =B) and (v_1 =F)
SR_2	1011010040004010344	(d_1 =VF) and (d_2 =F) → (v_0 =FF) and (v_1 =FF)
SR_3	1031010040002010344	(d_1 =VF) and (d_2 =F) → (v_0 =FF) and (v_1 =FF)
SR_4	1030200011100031113	(d_4 =F) and (d_7 =F) → (v_0 =B) and (v_1 =F)
SR_5	1030201041401010041	(d_4 =VC) and (d_7 =F) → (v_0 =FF) and (v_1 =B)
SR_6	1021410030404030401	(d_1 =VC) and (d_2 =VF) → (v_0 =BF) and (v_1 =B)
SR_7	1001010040002010344	(d_1 =VF) and (d_2 =F) → (v_0 =FF) and (v_1 =FF)
SR_8	1000201011401010041	(d_4 =VC) and (d_7 =F) → (v_0 =FF) and (v_1 =B)
SR_9	1100201011401010341	(d_0 =VF) and (d_4 =VC) → (v_0 =FF) and (v_1 =B)
SR_{10}	1000404040202031102	(d_7 =F) → (v_0 =BF) and (v_1 =S)
SR_{11}	1100001011401010341	(d_0 =VF) and (d_4 =VC) → (v_0 =FF) and (v_1 =B)
SR_{12}	1011010040002010344	(d_1 =VF) and (d_2 =F) → (v_0 =FF) and (v_1 =FF)

Table 3
Activated rules of an individual with SR_2

Rule	Meaning
R_6	(d_4 =F) and (d_7 =F) → (v_0 =B) and (v_1 =F)
R_7	(d_1 =VC) and (d_2 =VF) → (v_0 =BF) and (v_1 =B)
R_{10}(=SR_2)	(d_1 =VF) and (d_2 =F) → (v_0 =FF) and (v_1 =FF)
R_{11}	(d_4 =VC) → (v_0 =FF) and (v_1 =B)
R_{12}	(d_2 =VF) and (d_4 =VF) and (d_5 =C) and (d_6 =F) and (d_7 =VF) → (v_0 =FF) and (v_1 =B)



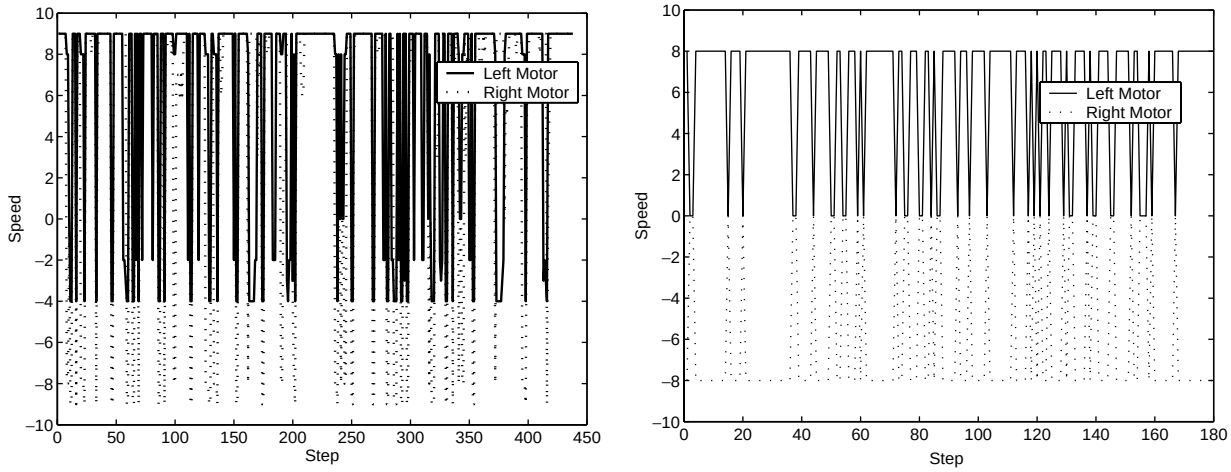
(a) With SR_2 (b) Without SR_2

Fig. 13. Snapshot of an individual in relation to SR_2.

Figure 16 indicates that SR_{10} drives the robot to turn right when it meets the front wall. This can be easily identified by the fact the robot does not move anywhere in Fig. 16(b) when it meets the front wall, whilst it turns right and continues moving in Fig. 16(a).

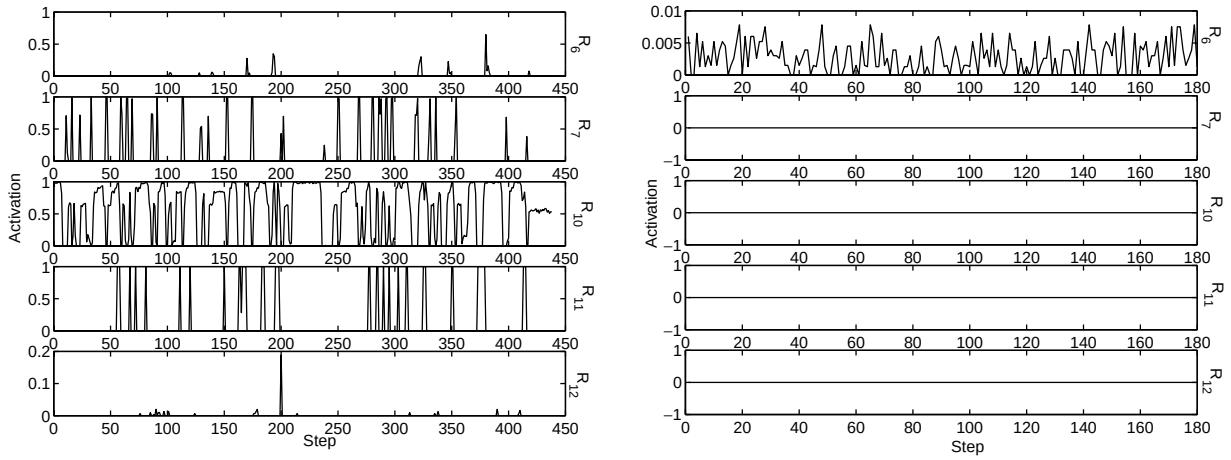
Figure 17(a) shows the speed changes of the two motors with SR_{10}, and (b) shows the speed changes without SR_{10}. The speeds of the two motors drop to zero after about 10 steps in Fig. 17(b), which means that the robot does not move anymore when it encounters the front wall. However, in the case of Fig. 17(a), the robot changes the motor speeds continuously during the steps.

Figure 18 shows the activations of the rules of Table 4. In the case of (a), all the rules are activated during the test while only R_{10} is activated in the case of (b). By R_{10}, the robot moves forward at the beginning. However, when it comes to the front wall, it does not move anymore because no rule is fired as shown in Fig. 18(b). On the contrary, R_6 fires a little just before at step 10, which causes the robot to turn right in Fig. 16(a). After this, R_{10} is fired again and drives the robot to move forward. By the interactions of the two rules, R_6 and R_{10}, the robot turns right and continues moving. From this analysis, we can conclude that SR_{10} makes the robot turn right when it encounters the front wall or an obstacle.



(a) With SR_2 (b) Without SR_2

Fig. 14. Speed of an individual in relation to SR_2.



(a) With SR_2 (b) Without SR_2

Fig. 15. The activations of related rules.

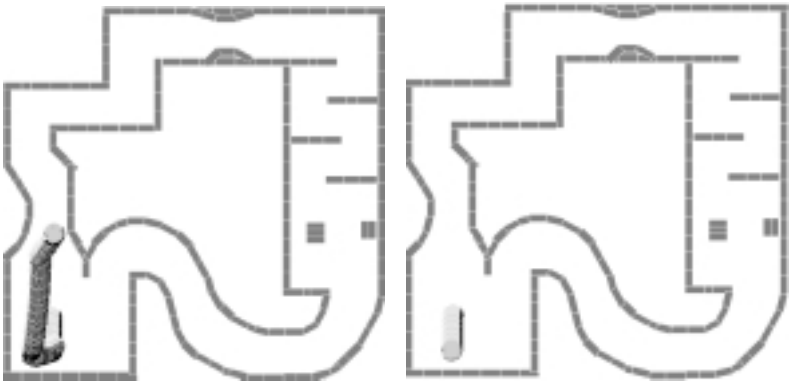
Table 4
Activated rules of an individual with SR_{10}

Rule	Meaning
$R_6(= SR_{10})$	(d_7 =F) → (v_0 =BF) and (v_1 =S)
R_{10}	(d_1 =VF) and (d_2 =VF) → (v_0 =FF) and (v_1 =FF)
R_{11}	(d_4 =VC) and (d_7 =VF) → (v_0 =FF) and (v_1 =B)

5.2.3. Analysis of salient rule SR_8

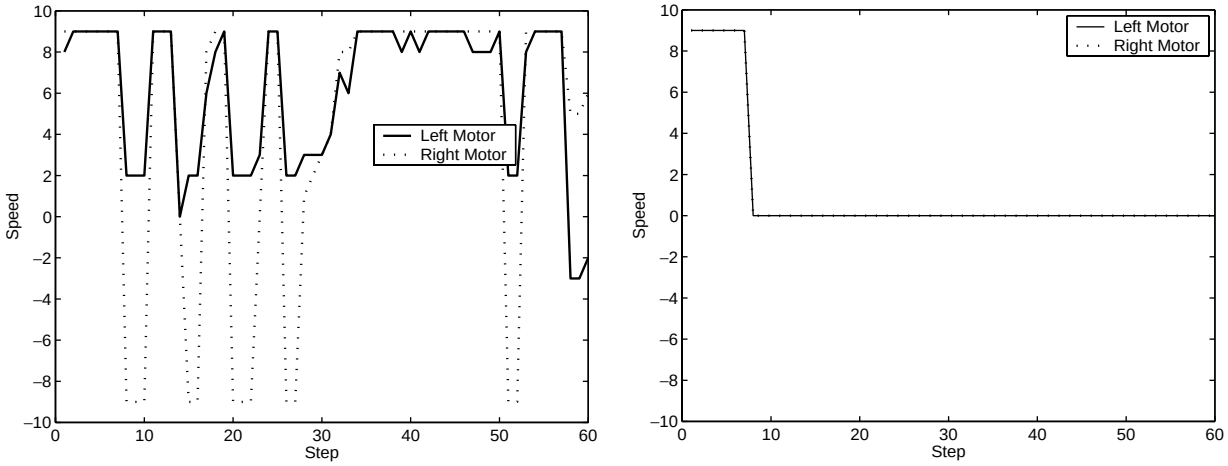
The SR_8 genotype occurs at about the 420th generation and lasts for about 200 generations. One individual with SR_8 was selected and analyzed. Table 5 shows the rules activated during this test. SR_8 is, in this case, R_{11} of this individual as shown in Table 5.

Figure 19(a) shows the behaviors with SR_8 included and Fig. 19(b) shows the behaviors with SR_8 intentionally excluded to find out the role of SR_8. Figure 19 indicates that SR_8 turns the robot left when it is very close to the right wall. This can be identified by the fact the robot does not move away from the right wall, which it bumps into, as shown in Fig. 19(b), whilst it continues moving after turning left at the same situation



(a) With SR_{10} (b) Without SR_{10}

Fig. 16. Snapshot of an individual in relation to SR_{10}.



(a) With SR_{10} (b) Without SR_{10}

Fig. 17. Speed of an individual in relation to SR_{10}.

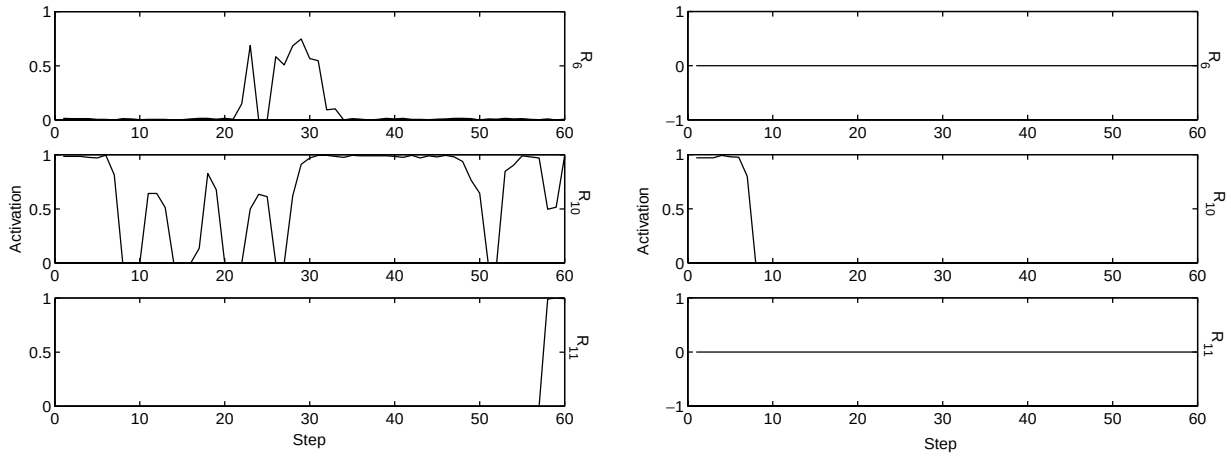
Table 5
Activated rules of an individual with SR_8

Rule	Meaning
R_3	(d_0 =F) and (d_1 =C) and (d_5 =VF) and (d_6 =F) → (v_0 =S) and (v_1 =S)
R_6	(d_4 =F) and (d_7 =F) → (v_0 =BF) and (v_1 =S)
R_7	(d_1 =VC) and (d_2 =VF) → (v_0 =BF) and (v_1 =B)
R_{10}	(d_1 =VF) and (d_2 =VF) → (v_0 =FF) and (v_1 =FF)
$R_{11}(= SR_8)$	(d_4 =VC) and (d_7 =F) → (v_0 =FF) and (v_1 =B)
R_{19}	(d_1 =F) and (d_3 =VF) and (d_4 =VF) and (d_6 =C) and (d_7 =VF) → (v_0 =BF) and (v_1 =B)

in Fig. 19(a).

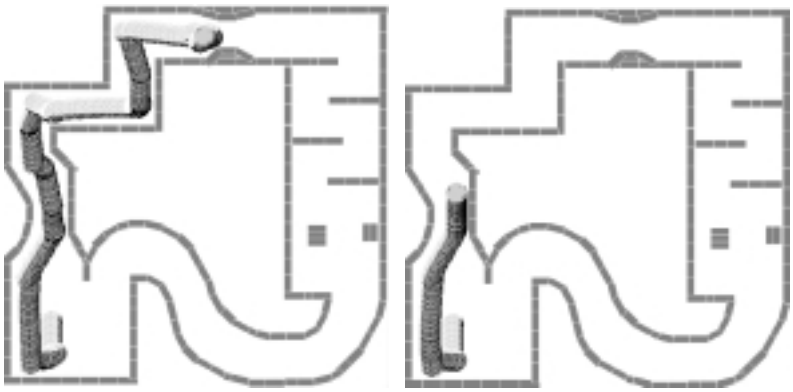
Figure 20(a) shows the speed changes of the two motors with SR_8, and (b) shows the speed changes without SR_8. At step 60 when the robot bumps into the right wall, the speeds of the two motors in Fig. 20(b) are the same. Because of this, the robot could not escape from that situation. On the contrary, with SR_8, the robot easily avoids the right wall and continues moving as shown in Fig. 19(a).

Figure 21 shows the activations of the rules of Table 5. In the case of (a), all the rules are activated during the test while only some of the rules (excluding SR_8) are activated in the case of (b). The activation of SR_8 at about step 60 is over zero and, therefore, it drives



(a) With SR_{10} (b) Without SR_{10}

Fig. 18. Rule activations of an individual in relation to SR_{10}.



(a) With SR_8 (b) Without SR_8

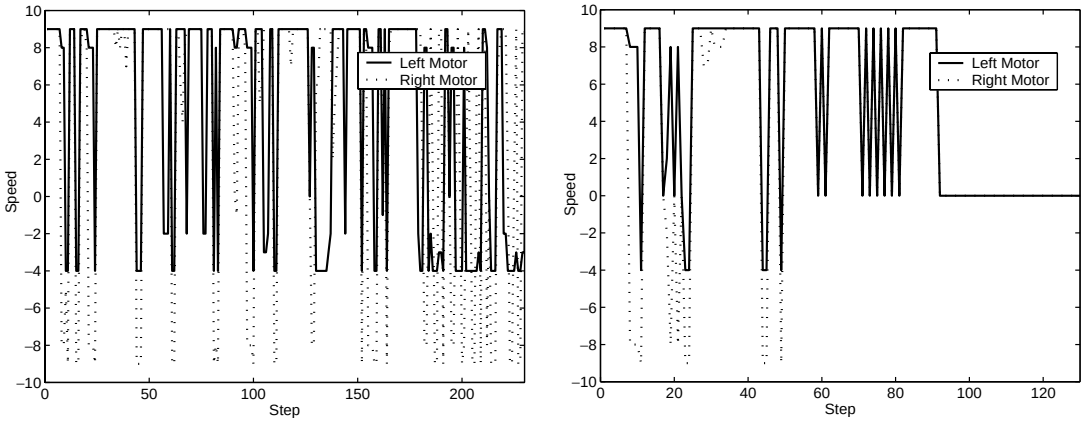
Fig. 19. Snapshot of an individual in relation to SR_8.

the robot to turn left slightly keeping the robot from bumping into the right wall. Without SR_8 as shown in Fig. 21 (b), however, R_{10} causes the robot to bump into the right wall and makes the robot stop when it confronts the front wall as shown in Fig. 19(b). When it encounters front wall, even R_{10} could not be fired. From this analysis, we can conclude that SR_8 turns the robot left when it encounters the right wall.

6. Concluding remarks

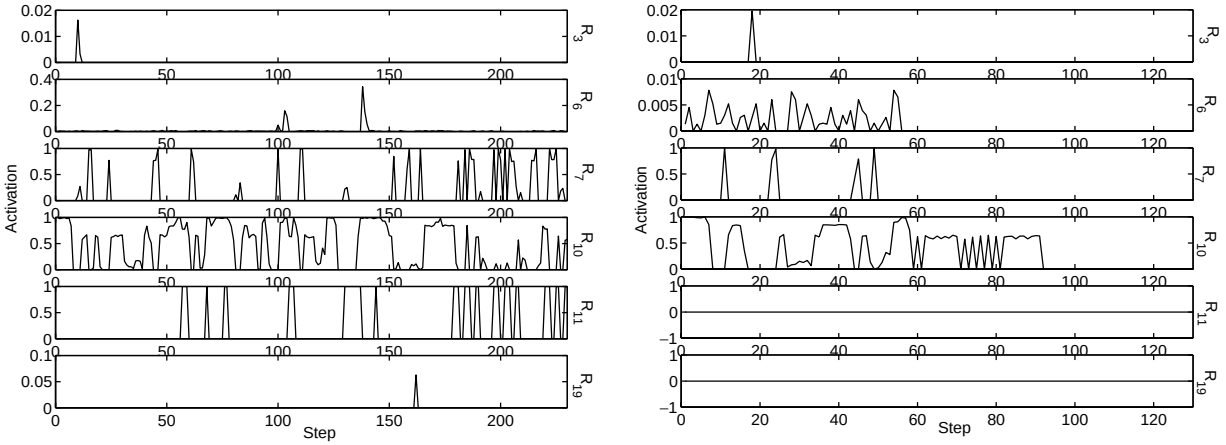
In this paper, we have quantified the evolution of a fuzzy logic controller for a mobile robot and shown that the rules of the best individual are the result of their adaptive capabilities using evolutionary activity statistics. Furthermore, we have also illustrated what behavioral properties were preferred during the evolution by identifying salient rules with evolutionary activities.

The evolutionary activity statistics have shown that the performance of the best individual is not the result of other genetic phenomena, such as chance or genetic drift, but the result of its adaptabilities. Analysis of the salient rules shows that their behavioral properties have considerable significance with regard to the achievement of the goal. Furthermore, the rules of the best individual have close relations with the salient rules that we analyzed in previous sections. Actually, some of the salient rules are directly copied into the best individual's rule set. Others are supposed to be descendants of the salient rules, though we have not clearly shown this in this paper. Our next research topic will be to reveal



(a) With SR_8 (b) Without SR_8

Fig. 20. Speed of an individual in relation to SR_8.



(a) With SR_8 (b) Without SR_8

Fig. 21. Rule activations of an individual in relation to SR_8.

the relations between the salient rules and the rules of the best individual.

In the perspective of evolutionary optimization, however, there is room for improvement. Although the coding scheme used in this paper is intuitive and easily implemented, we have to say that it is not an optimal coding scheme because of its large dimensionality. The dimensions of search space would be reduced with a better coding scheme, such as DNF coding [7,17], that allows both multiple labels associated to the same variable and a wildcard for variables. However, this does not interfere with the focus of this paper. As we have shown until now, maintaining a certain level of evolvability is the key to successful evolution and only that can make repetitive experiments successful. We believe that that the presentation of measured evolvability is necessary for every evolutionary experiment to be more persuasive and objective.

Based on the measured evolvability of the evolutionary fuzzy system, we can say, as a conclusion, that the evolved fuzzy controller is the result of adaptive evolution by the quantification of adaptability, and that evolutionary activity proves to be useful for quantifying adaptability.

Acknowledgments

The research reported here was supported in part by a contract with the Telecommunications Advancement Organization of Japan entitled, “Research on Human

Communication" and by Korea Institute of Science and Technology in 2002.

Appendix

The basic structure of a fuzzy logic controller consists of three conceptual components: fuzzification of the input-output variables, a rule base which contains a set of fuzzy rules, and a reasoning mechanism which performs the inference procedure on the rules and given facts to derive a reasonable output. In this appendix, we present a mathemathical formalization of fuzzy inference for our fuzzy system.

A.1. Fuzzy sets

The nth rule can be represented as a fuzzy relation defined by Eq. (A1).

$$R_n : (D_0^n \times D_1^n \times D_2^n \times D_3^n \times D_4^n \times D_5^n \times D_6^n \times D_7^n) \rightarrow (V_0^n, V_1^n) \tag{A1}$$

where $\rightarrow$ denotes the fuzzy relation. This fuzzy relation can be implemented with each corresponding membership function defined by Eqs (A2) and (A3).

$$\mu_{RV_0^n}(d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7, v_0) = f(\mu_{D_0^n}(d_0), \cdots, \mu_{D_7^n}(d_7), \mu_{V_0^n}(v_0)) \tag{A2}$$

$$\mu_{RV_1^n}(d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7, v_1) = f(\mu_{D_0^n}(d_0), \cdots, \mu_{D_7^n}(d_7), \mu_{V_1^n}(v_1)). \tag{A3}$$

A.2. Fuzzy inference

Here, we define fuzzy rules, fuzzy reasoning, and defuzzifiation. Let D_i^n and $D_i^{n'}$ be the fuzzy sets defined on d_i in the universe of discourse U_d, and $V_0^n, V_1^n, V_0^{n'}$ and $V_1^{n'}$ be the fuzzy sets defined on v_0 and v_1 in the universe of discourse U_v, respectively. To control the actions of the mobile robot, $V_0^{n'}$ and $V_1^{n'}$ should be inferred from $D_i^n, D_i^{n'}, V_0^n$ and V_1^n. The nth rule R_n can be transformed into a fuzzy relation based on Mamdani's fuzzy implication function [19]. Based on Zadeh's compositional rule of inference [27], V_0' and V_1' are expressed as

$$V_0' = (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} (D_0^n \times \cdots \times D_7^n \rightarrow V_0^n) = (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} RV_0^n \tag{A4}$$

$$V_1' = (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} (D_0^n \times \cdots \times D_7^n \rightarrow V_1^n) = (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} RV_1^n \tag{A5}$$

where $\circ$ denotes the maximum-minimum composition. The resulting V_0' and V_1' are expressed as in the following equations.

$$\mu_{V_0'} = \bigcup_{n=0}^{N-1} \underbrace{\left\{\vee_{d_0}\left[\mu_{D_0^{n'}}(d_0) \wedge \mu_{D_0^n}(d_0)\right]\right\}}_{\omega_0} \wedge \cdots \wedge \underbrace{\left\{\vee_{d_7}\left[\mu_{D_7^{n'}}(d_7) \wedge \mu_{D_7^n}(d_7)\right]\right\}}_{\omega_7} \wedge \mu_{V_0^n}(v_0) = \bigcup_{n=0}^{N-1} \underbrace{(\omega_0 \wedge \cdots \wedge \omega_7)}_{\text{firing strength}} \wedge \mu_{V_0^n}(v_0) \tag{A6}$$

Similarly, $\mu_{V_1'}$ is defined as

$$\mu_{V_1'} = \bigcup_{n=0}^{N-1} \underbrace{(\omega_0 \wedge \cdots \wedge \omega_7)}_{\text{firing strength}} \wedge \mu_{V_1^n}(v_1) \tag{A7}$$

where $\wedge$ denotes the minimum operation and ω_i is the maxima of the membership functions of $D_i^n \cap D_i^{n'}$.

Defuzzification refers to the way $\overline{v}_0$ and $\overline{v}_1$ are extracted from a fuzzy set as representative values. Among the many defuzzification methods [19,20,25, 27], the center of gravity method is used because it is widely used and appropriate for our system to control the mobile robot.

$$\overline{v}_0 = \frac{\int_{v_0} \mu_{v_0}(v_0) v_0 dv_0}{\int_{v_0} \mu_{v_0}(v_0) dv_0} \tag{A8}$$

$$\overline{v}_1 = \frac{\int_{v_1} \mu_{v_1}(v_1) v_1 dv_1}{\int_{v_1} \mu_{v_1}(v_1) dv_1}. \tag{A9}$$

References

[1] M.A. Bedau and N.H. Packard, Measurement of evolutionary activity, teleology, and life, *Artificial Life* **2** (1992), 431–461.

[2] M.A. Bedau, E. Snyder, C.T. Brown and N.H. Packard, *A comparison of evolutionary activity in artificial evolving systems and in the biosphere*, in: Proceedings of the Fourth European Conference on Artificial Life, P. Husbands and I. Harvey, eds, MIT Press, Cambridge, MA, 1997, pp. 125–134.

[3] M.A. Bedau and C.T. Brown, Visualizing evolutionary activity of genotypes, *Artificial Life* **5** (1999), 17–35.

[4] M.A. Bedau, E. Snyder and N.H. Packard, A classification of long-term evolutionary dynamics, *Artificial Life* **4** (1998), 228–237.

[5] H.R. Beom and H.S. Cho, A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning, *IEEE Transactions on Systems, Man, and Cybernetics* **25**(3) (1995), 464–477.

[6] S.-B. Cho and S.-I. Lee, Mobile robot learning by evolution of fuzzy controller, *Journal of Intelligent and Fuzzy Systems* **6** (1997), 91–97.

[7] O. Cordón, F. Herrera, F. Hoffmann and L. Magdalena, *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, World Scientific, Singapore, 2001.

[8] T. Fukuda and N. Kubota, An intelligent robotic system based on a fuzzy approach, *Proceedings of the IEEE* **87**(9) (1999), 1448–1470.

[9] D.E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley, 1989.

[10] J.H. Holland, *Adaptation in natural and artificial systems*, MIT Press, 1992.

[11] K. Izumi, K. Watanabe and T. Miyazaki, *Fuzzy behavior-based control for a miniature mobile robot*, in: Proceedings of the 2nd Int. Conf. on Knowledge-Based Electronic Systems, 1998, pp. 483–490.

[12] J.S.R. Jang, C.T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, 1997.

[13] P.J. King and E.H. Mamdani, The application of fuzzy control system to industrial process, *Automatica* **13** (1997), 235–242.

[14] K-Team, *Khepera Simulator Version 5.02 User Manual*, 1997.

[15] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller, part 2, *IEEE Transactions on Systems, Man, and Cybernetics* **20**(2) (1990), 419–435.

[16] M.H. Lim, S. Rahardja and B.W. Gwee, A GA paradigm for learning fuzzy rules, *Fuzzy Sets and Systems* **82** (1996), 177–186.

[17] L. Magdalena and F. Monasterio, *Evolutionary-based learning applied to fuzzy controllers*, in: Proceedings 4th IEEE International Conference on Fuzzy Systems and the Second International Fuzzy Engineering Symposium (FUZZ-IEEE/IFES'95) 3, IEEE, 1995, pp. 1111–1118.

[18] C.C. Maley, *Four steps toward open-ended evolution*, in: Proceedings of the Genetic and Evolutionary Computation Conference, W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. J and R.E. Smith, eds, Morgan Kaufmann, San Francisco, CA, 1999, pp. 1336–1343.

[19] E.H. Mamdani and S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Man-Machine Studies* **7**(1) (1975), 1–13.

[20] N. Pfluger, J. Yen and R. Langari, *A defuzzification strategy for a fuzzy logic controller employing prohibitive information in command formulation*, in: Proceedings of IEEE International Conference on Fuzzy Systems, IEEE, San Diego, CA, 1992, pp. 717–723.

[21] A generic neutral model for measuring excess evolutionary activity of genotypes, in: Proceedings of the Genetic and Evolutionary Computation Conference, W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V.Honavar, M. J and R.E. Smith, eds, Morgan Kaufmann, San Francisco, CA, 1999, pp. 1366–1373.

[22] T.L. Seng, M.B. Khalid and R. Yusof, Tuning of a neuro-fuzzy controller by genetic algorithm, *IEEE Transactions on Systems, Man, and Cybernetics-Part b: Cybernetics* **29**(2) (1999), 226–236.

[23] T. Shannon, Generic behavior in the Lindgren non-spatial model of iterated two player games, *Artificial Life* **4** (1998), 316–325.

[24] J.M. Smith, *The Theory of Evolution*, New York, 1975.

[25] R.R. Yager and D.P. Filev, A simple adaptive defuzzification method, *IEEE Transactions on Fuzzy Systems* **1**(1) (1993), 69–78.

[26] L.A. Zadeh, Fuzzy sets, *Information and Control* **8** (1965), 338–353.

[27] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Transactions on Systems, Man, and Cybernetics* **3**(1) (1973), 28–44.