

## LEARNING OF BIOLOGICAL BEHAVIOUR BY CLASSIFIER

Hideaki Kanoh<sup>1)</sup> and Akihiro Hosokawa<sup>2)</sup>

1) School of Science and Technology, MEIJI University  
1-1-1. Higashimita, Tama-ku, Kawasaki, 214-8571, JAPAN  
E-mail : kanoh@isc.maiji.ac.jp

2) Research and Development Department, DAINIPPON SCREEN MFG.CU.LTD.  
322 Furukawa-cho, Hazukashi, Fushimi-ku, Kyoto, 612-8486, JAPAN  
E-mail: hosokawa@screen.co.jp

**Abstract:** *The objective of this paper is to investigate biological features of a virtual creature on the computer. It is assumed that the creature does not have a complex judgment, a sophisticated detector and has simple basic actions. We call it a bug. As a lower animal, an insect does not have complex brain, while it is seemed to be able to do complicated works. Specifically complex behaviour of an insect is considered to be based on a reflex action for the external stimuli. The classifier system is used for the generation of bug's behaviour rules in a complicated environment made in the computer. The classifier system is used for the generation of bug's behaviour rules in a complicated environment made in the computer. A classifier system is a reflex system, which generates an action soon after receiving a stimulus from the environment. This is quite similar to the feature of lower animal's function of processing the information.*

*The bug is released at some spot in the virtual world where baits are randomly located and there are obstacles and enemies. The classifier is rewarded only when the bug succeeds to obtain the bait. The more the bug learned, the more new classifier rules emerge: such as a rule to take a step forward or another rule to change the direction towards bait. At this stage, the bug learns the way to obtain the bait avoiding the obstacles and enemies by itself.*

*The results of the observation of simulation experiments show that the chain of the continuous action which consists of multiple classifiers is arising.*

**Keywords:** - learning, classifier, genetic algorithm, biological behaviour, bucket brigade

### 1. INTRODUCTION

The objective of this paper is to investigate biological features of a virtual creature on the computer. It is assumed that the creature does not have a complex judgment, a sophisticated detector and has simple basic actions. We call it a bug. As a lower animal, an insect does not have complex brain, while it is seemed to be able to do complicated works. Specifically complex behaviour of an insect is considered to be based on a reflex action for the external stimuli. The classifier system is used for the generation of bug's behaviour rules in a complicated environment made in the computer. A classifier system [1] [2] is a reflex system, which generates an action soon after receiving a stimulus from the environment. This is quite similar to the feature of lower animal's function of processing the information.

The bug is released at some spot in the virtual world where baits are randomly located and there are obstacles and enemies. The classifier is rewarded only when the bug succeeds to obtain the

bait. The more the bug learned, the more new classifier rules emerge: such as a rule to take a step forward or another rule to change the direction towards bait. At this stage, the bug learns the way to obtain the bait avoiding the obstacles and enemies by itself.

The results of the observation of simulation experiments show that the chain of the continuous action which consists of multiple classifiers is arising.

### 2. SETTING OF VIRTUAL ENVIRONMENT AND VIRTUAL BUG

Firstly we consider an environment where a bug and one bait (▲) are being put in the field surrounded by a wall as shown in Fig.1. The bug can take three kinds of movement of advance, left turn and right turn as shown in Fig. 2.

As means for obtaining information from the environment, the bug has a direction sensor for knowing the direction of the bait and a contact sensor for knowing that the obstacle to be contacted (Fig.3).

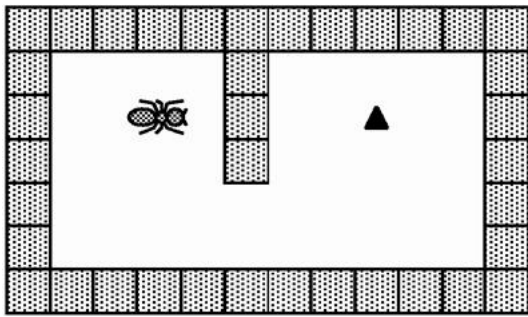


Fig 1 - Virtual environment.

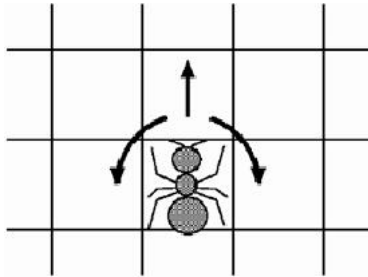


Fig.2 - Action of bug.

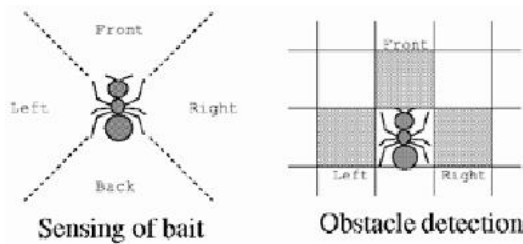


Fig.3 - Detection of the bug.

### 3. CLASSIFIER SYSTEM

The classifier system (CFS) is a set of rule called a classifier (CP). Each CF has the following form  
if < condition > then < action >

and also is given a weight called strength that represents the degree of creditability for the rule. New rule is generated from other rules by the genetic algorithm using the strength of CF as a fitness.

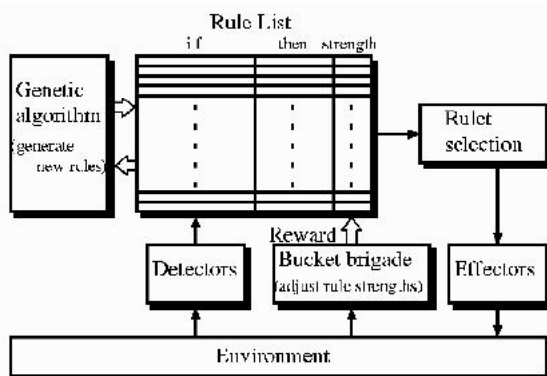


Fig. 4 - The composition of the classifier system.

### 3.1. System constitution

The composition of the classifier system is shown in the Fig.4. The black arrow shows the flow or information and the white arrow suggests optimization of rule lists. The elements of the system are as follows:

**Rule List** A set of rules that is called classifiers, the <condition> part of which has the form  
<condition >= {0, 1, #}<sup>n</sup>, n: length of condition, where # can take either 0 or 1.

**Detector** It converts input information from the environment into the code used in the system. The designer makes the conversion table, which is called coding.

**Effector** It converts an output of CFS into the command which gives the bug a motion in the environment. The designer makes the conversion table as coding.

**Roulette selection** When many conditions match to an input from environment, one CF is chosen at the probability in proportion to the strength and fitness. We say that the CF is activated. The activated CF is identified by the number which is numbered in order of generation.

**Genetic algorithm** Genetic algorithm is used to generate a new CF from CF's that have high strength.

**Credit allocation** The mechanism for distributing the reward to CF's acting in the current iteration and the payoff to CF's acting in the previous iteration. Generally the bucket brigade algorithm is used to pass strength value as payoff to CF's that caused new CF's to act as a result of the internal messages sent in the previous iteration of the CFS.

### 3.2. Flow of the algorithm

The routine of CFS is shown in the following.

[Step 1] The condition part of the message which has come from the environment through the detector and all rule is compared.

[Step 2] One CF is selected from CF's for which the condition match at the probability in proportion to the strength and fitness and the bucket brigade algorithm is performed.

[Step 3] The action code of the selected CF is decoded in the effector and output it to the environment.

Back to Step 1

### 3.3 Credit allocation

When CF s matched at step  $k-1$  and CF  $c$  matches step  $k$ , a bid is given as follows:

$$B(c, s, k) = C \frac{R(c)}{b} \{p(c, k) - p(s, k)\} \quad (1)$$

where  $C$  is a constant coefficient,  $p(c, k)$ ,  $p(s, k)$  are the strength values of  $c$  and  $s$  respectively.  $b$  is the bit length of condition,  $R(c)$  is the total bit length of condition except #. If  $c$  gets the largest bid among CF's that match to the input,  $c$  has to pay the bid to CF's that paid the bid in previous iteration and the strength value of  $c$  becomes as follows.

$$p(c, k) = p(c, k - 1) - B(c, k) \quad (2)$$

The strength values of previous winners become

$$p(s, k) = p(s, k) - 1 + B(c, k) \quad (3)$$

#### 4. CONSTRUCTION OF CLASSIFIER AND FIELD

##### 4.1. Coding of classifier

The hug has a direction sensor by which it is possible to generously know the direction of a bait of every 90 degrees and a contacting sensor by which the hug can detect contact with left, right and front obstacles. One CF is composed of the binary number of total 7 bits; the condition part of 5 bits and the action part of 2 bits as shown in Table 1.

**Table. 1 Allocation of hit**

a	b	c	d	e	f	g
1	0	1	1	0	1	0

condition	action
condition a, b	direction of bait
c	obstacle (left)
d	obstacle (front)
e	obstacle (right)
action f, g	acting pattern

**Table 2. a,b coding and f,g coding**

a b	bait	f g	Action
00	Front	00	Advance
11	Back	11	Advance
10	Left	10	Left turn
01	Right	01	Right turn

The allocation of the value to the each bit is given in Table 2. As to the c, d and e, if there is a contact to an obstacle in the direction, 1 is taken and else it takes 0.

#### 5. SIMULATION

The simulation is carried out in order to examine the behavior of the classifier system on case of the simple obstacle (shown in Fig. 1) and on case of a more complicated obstacles.

##### 5.1. Rule of simulation

1. As starting positions, the bug is put on the left part of the field and the bait is located in the right part of the field.

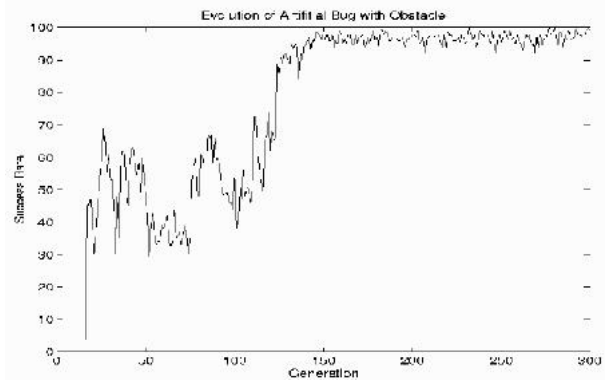
2. If the bug gets the bait within 50 steps, it gets a reward and the simulation is finished. The bug and the bait are relocated at the starting positions, then the simulation restarts.

3. If the bug can not get the bait within 50 steps, the simulation is finished. The bug and the bait are relocated at the starting positions, then the simulation restarts.

4. The genetic algorithm is started in the every 100 simulations to create new classifiers.

##### 5.2. Simple obstacle

The action of avoiding the obstacle is a difficult problem for actual organism. In spite of a bait being in the front, it is necessary to change the direction to right or left, when there is an obstacle between the bug and the bait. That the bug receives the reward is only the case in which it reaches a bait, and CF which contributes to avoiding the obstacle, does not receive the direct reward. To grow such classifier, the classifier which received the direct reward must distribute the part of the reward to the classifier which contributes to avoiding the obstacle. The result is shown in Fig. 5. It is shown that the rapid learning is happening in 15 generation and 122 generation. It is proven that next four classifiers (Table.3) rapidly increased (he strength, when classifiers of 15 generation are examined. These show the fundamental action in which the hug approaches the direction of a bait.



**Fig.5 - Learning of the bug.**

**Table. 3 Rapid grown classifiers of 15 generation**

CF	strength	hail	obs.	action
0 0 0 0 0 0	2950	front &	no	advance
1 0 0 0 1 0	1570	left &	no	L turn
0 0 0 0 1 1	1510	front &	no	advance
0 1 0 0 0 1	870	right &	no	R turn

The rapid learning of 122 generation is based on next classifier. This classifier arises in 115 generation, and the

CF	strength	hail	obs.	action
0 0 1 1 0 0 1	3230	front	& L&F	R turn

strength has reached 10000 in 132 generation. This operation shows the motion which changes the direction in the direction along the wall, when it collided with the obstacle (Fig. 6). The classifier which becomes a cause of the rapid advance in 122 generation arises in 115 generation, and the strength is rapidly increased afterwards. By the generation of this CF, the continuous action which avoids the obstacle by changing the direction in colliding with the obstacle arises.

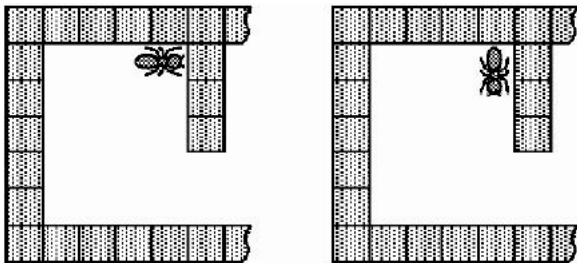


Fig. 6 - Turn along the wall.

5.3. More complicated obstacles

For further simulation, environment with more complicated obstacles is set as shown Fig 7 where bait is put on the field in the right part and the hug is released in the left part at random position. The pattern of partition plates is randomly changed in the every simulation.

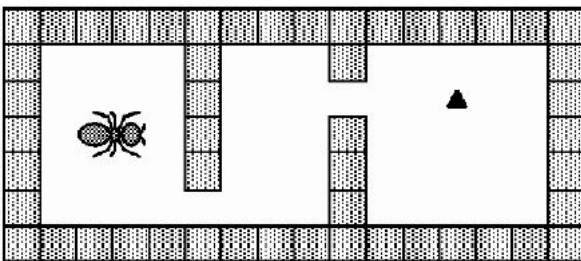


Fig.7 - Setting or environment.

Simulation result is shown in Fig.8. The upper figure shows the change of the success rate of the bait acquisition. Horizontal axis is number of generations, and the vertical axis is success rate. The lower figure shows the change of the strength of each CF. The learning advances in the stepwise (points a and b) and the learning once goes down at point c. Even if the success rate progresses calmly grow to the point c, the strength of each CF intensely changes. In the meantime, curves gently change after the point c not only success rate but also strength. In collecting multiple CF, the bundle-like patterns have been formed.

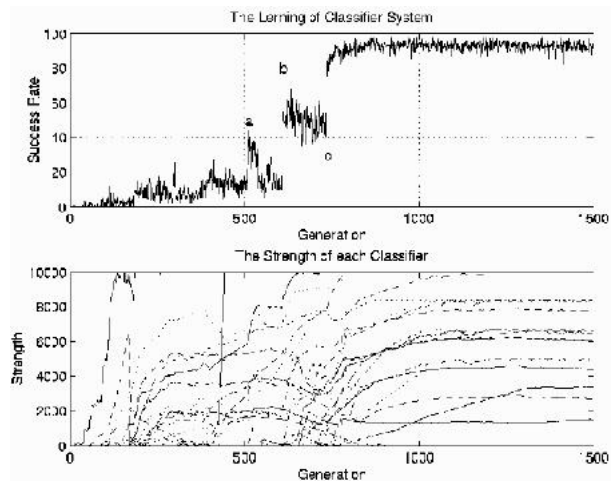


Fig. 8 - Progress of learning and changes of CF's.

5.4 appearance of new cf

Fig.9 and Fig.10 show magnified figures of the vicinities of point a and point b respectively. The success rate increased temporarily at point a, and it has returned to the success rate of the origin in 539 generation. It rises again near the point b. CF [2558] increases the strength near the point a and CF's [3020] and [3020] increase the strengths near the point b.

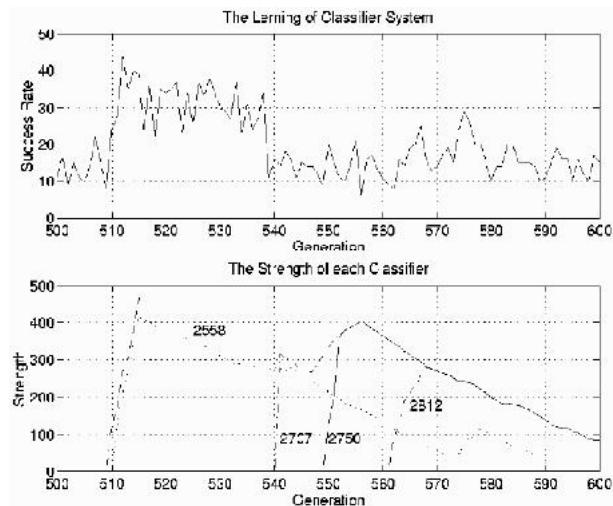


Fig.9 - Progress of learning and changes of CF's magnified near the point a.

As [3020] and [2558] have the completely equal bit sequence, this CF seemed to become a cause of the advance in point s and point b. CF [2558] that arises at the point a and once increases the strength gradually reduces the strength and is finally extinguished. However, [3020] smoothly increases the strength, after it arises in the b point, though it has the bit sequence equal to [2558], and the strength in 1499 generation becomes 8349. The strength of [3020] once decreased after the generation in 604 generation, and it rapidly increases the strength with the generation of [3041].

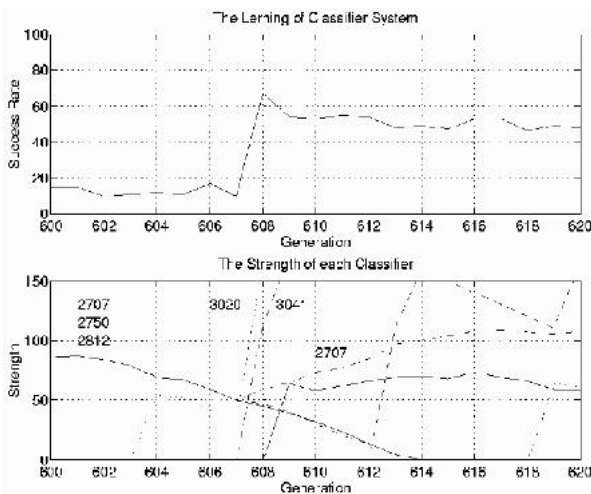


Fig. 10 - Progress of learning and changes of CF's magnified near the point b.

5.5. Forming of sequence

The success rate is rapidly improved near the point c, and it rises about 90%. However, newly generated CF is not found.

In the lower figure of Fig.8, the strength of each CF intensively changes in the interval between point b and point c and it gently changes after the point c. Multiple CF's curves form together the bundle-like patterns, which shows emergence of sequence. Because the phenomenon that the strength of multiple CF is averaged cannot occur except for the case where the bucket brigade algorithm averages strengths of CF's relating deeply to each other.

Table 4 shows the elements of the sequences linked strongly in 1499 generation. This sequence is interpreted using Fig. 11. When the bug is put at the position (a) and bail is in the direction of the right, if [3928] is activated, the bug turns to the 90 degree right. Since in next step, activated CF does not exist, the same action is continued, and the bug becomes a condition of (b). If [3096] or [1047] is activated, the bug takes forward one step. After the advance was repeated, if there are obstacles in right and left like (d). [3677] is activated and the bug turns to the left (e). This last action seems to be no meaning, but at (e) there is high possibility in which [1063] is activated and the bug turns to the right direction.

Table 4. Elements of sequence (1499 generation)

NO.	ID	condition	action	strength
4	1063	0 1 0 1 0	0 1	10000
21	3096	0 0 1 0 0	0 0	4374
22	3928	1 1 0 1 1	0 1	4374
23	3677	0 0 1 0 1	1 0	4373
24	1047	0 0 1 0 0	1 1	4372

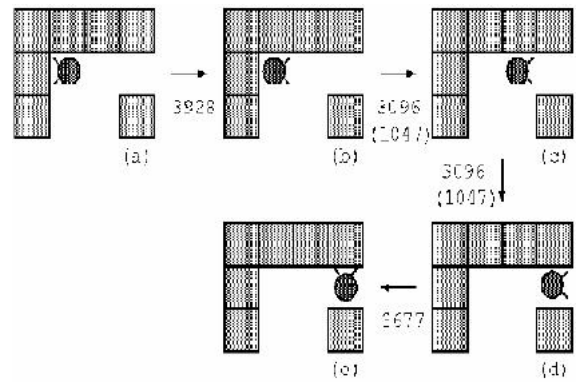


Fig. 11 - Element of bug by sequence.

When [1063] is activated, [3677] that was activated one step before receives the strength from [1063] and this strength also spreads to CF's belonging to the same sequence. As the strength of [1063] is 10,000, the sequence gets large profit. This profit spreads in all classifiers of the sequence.

5.6. Selection of sequence

In Fig.10, three CF's [2707], [2750] and [2812] arise in the interval from 540 generation to 570 generation and the strengths of the three rules agree from the middle point, which indicates emergence of a sequence. This sequence reduces the strength value in the every generation and finally extinct. This fact shows that the series of behaviours of this sequence has not been connected with the profit. By the rule of bid, the sequence connected with the profit increases the strength more and more. and the strength decreases on the sequence not connected with the profit. From this result, it is proven that emergence and selection of a sequence are also carried out in sequences.

6 ENVIRONMENT WITH ENEMIES

6.1 setting of classifier system

Environment with the enemies of Fig.12 is made to confirm whether it can learn an appropriate action pattern without getting confused in information in the case in which sensory information is enormous then bit string of classifier is lengthened. The bit string of the classifier is shown in Table.5 where the bug has both a bait sensor and an enemy sensor. And both sensors can output information of the distance in three steps.

The field is endless, that is, the edge of the top is connected with the edge of the bottom and also the edge of the right is connected with the edge of the left.

6.2 rules of simulation

1. As starting positions, the bugs, enemies and baits are put randomly on the field.

2. If the bug gets the bait within 50 steps, it get a reward.
3. If the bug is captured by the enemy, the bug gets a negative reward.
4. Fixed number of baits are always placed in the field.
5. The genetic algorithm is started in the every 5,000 simulations to create new classifiers.

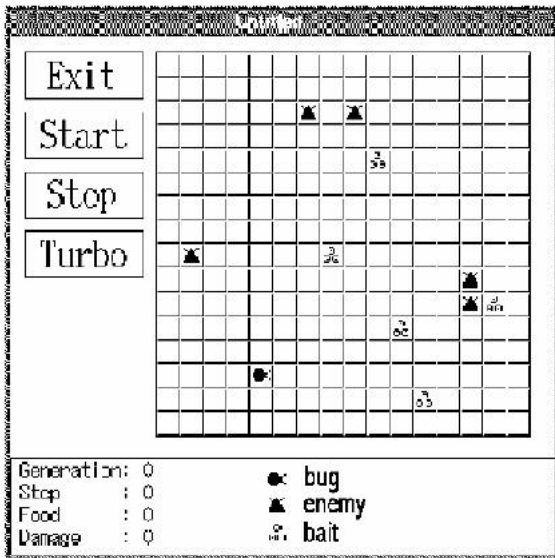


Fig. 12 - Environment with the enemies.

Table.5 Bit string of classifier

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
1	0	#	1	0	1	0	#	#	0	1	0	#	1	0	0	1	0
condition										action							

**Allocation of bit**

condition	a, b	distance from bait (front)
	c, d	distance from bait (right)
	e, f	distance from bait (back)
	g, n	distance from bait (left)
	i, j	distance from enemy (front)
	k, l	distance from enemy (right)
	m, n	distance from enemy (back)
	o, p	distance from enemy (left)
action	q, r	acting pattern

**6.3 Simulation**

Simulation is performed for cases of five baits and five enemies and case of live baits and no enemy. The result is shown in Fig. 13 where the horizontal axis represents the number of start of the genetic algorithm and the vertical axis represents the number of acquired baits.

In comparison with the case in which there is an enemy, it is shown that about double baits are acquired, when there is no enemy. But observation shows that there is an action of passing through

the front of a bait, and we cannot say that the action of the bug is efficient. What is the cause of the badness of the efficiency? As the information is too much and it seems to be able to not extract important information. In actual insect, information from the sense organ is not used for the direct action decision. It is considered that the action has been decided on the basis of features of the information which is extracted in passing many layers of neurones. Then, we modify the detector of the classifier system to extract the features of the information artificially by adopting only information on nearest bait or nearest enemy as shown in Fig. 14. The simulation experimental result is shown in Fig. 15.

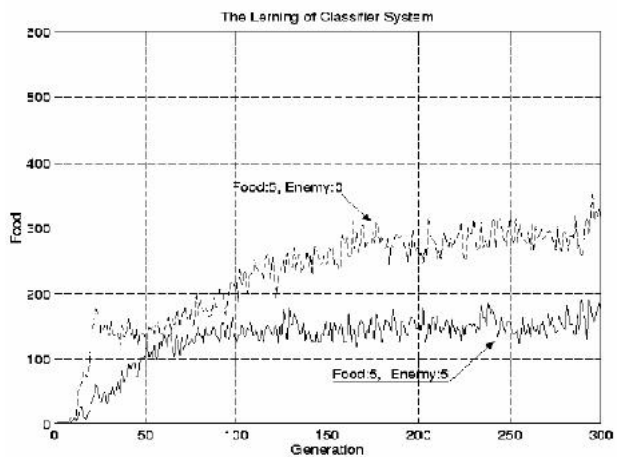


Fig. 13 - Learning in presence of enemy (I).

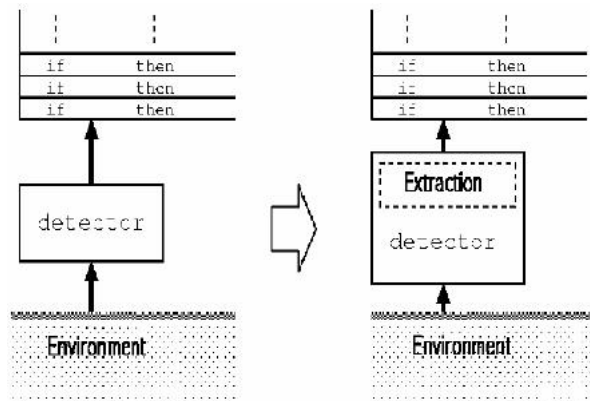


Fig. 14 - Modification of detection.

As a result of adding the circuit which extracts features, the acquisition number of a bait is nearly doubled, and movements of bugs are not so much futile. This result shows the importance of extracting features of information.

**7. CONCLUSION**

The results show that the complicated action is generated by a series of simple actions, and that

the complex action depends on a complex environment. By analyzing classifier in view of the strength, the several series of classifier are discovered. In the case of presence of enemy, information becomes too much to extract important information. By modifying the detector of the classifier system to extract the features of the information artificially adopting only information on nearest bait or nearest enemy, the acquisition number of a bait becomes nearly doubled, which shows the importance of extracting features of information.

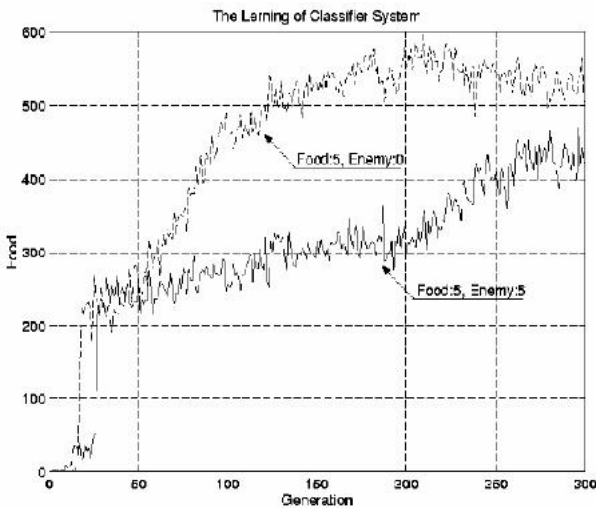


Fig. 15 - Learning in presence of enemy (II).

## 8. REFERENCES

[1] David E Goldberg: "Genetic Algorithms in search, Optimization & Machine Learning", Addison-Wesley Pub. Co. 1989

[2] David Jefferson: "Evolution as a Theme in Artificial Life: The Genesys/Tracker System", Artificial Life 2



Professor Hideaki Kanoh was born in Osaka, Japan on 1937. He received a B.E. degree from National Yokohama University and a D. Eng. degree from Osaka University. From 1961 to 1962, he was with the Mitsubishi Petrochemical Company as a research staff. From 1962 to 1969 he was with National Yokohama University, and from 1969 to 1986 he was with Osaka University. Since 1986 he has been with Meiji University.



Akihiro Hosokawa was born on Oct. 1971 in Sapporo JAPAN. He received the B.E. and M.E. degrees in Mechanical Engineering from Meiji University in 1995 and 1997 respectively. Since 1997 he has been with Dainippon Screen Mfg. Co. Ltd. where he works for research and development of semiconductor manufacturing equipments.