# A Study on DNA Coding Method and Knowledge Discovery

Tomohiro YOSHIKAWA

# A Study on DNA Coding Method and Knowledge Discovery



Tomohiro YOSHIKAWA

# A Study on DNA Coding Method and Knowledge Discovery

（DNAコーディング法と知識発見に関する研究）

1997

Department of Information Electronics

Nagoya University

Tomohiro YOSHIKAWA

# Contents


**1 Introduction** **1**

1.1 Genetic Algorithm . . . . . 1

1.1.1 Outline . . . . . 1

1.1.2 Advantages of Genetic Algorithm . . . . . 3

1.1.3 Modifications for GA . . . . . 4

1.2 Discovery of Fuzzy Rules . . . . . 6

1.2.1 Fuzzy Inference . . . . . 6

1.2.2 Application of GA to Discovery of Fuzzy Rules . . . . . 8

1.3 Research Goals . . . . . 9

1.4 Composition of the Thesis . . . . . 11

**2 DNA Coding Method** **12**

2.1 Introduction . . . . . 12

2.2 DNA Chromosome . . . . . 13

2.3 Genetic Operations . . . . . 18

2.4 Virus and Enzyme Operations . . . . . 20

2.5 Features of the DNA Coding Method . . . . . 23

2.6 Conclusions . . . . . 25

**3 Combination of the DNA Coding Method and Pseudo-Bacterial GA** **26**
3.1 Introduction . . . . . 26
3.2 Transfer of Bacterial DNA . . . . . 28
3.3 Conventional PBGA . . . . . 32
3.4 Combination of the DNA Coding Method and the PBGA . . . . . 35
3.5 Conclusions . . . . . 37

**4 Application of the DNA Coding Method** **38**
4.1 Introduction . . . . . 38
4.2 Problem Formulation for Knowledge Discovery . . . . . 39
4.3 Representation of Rules . . . . . 42
4.4 Genetic Operations . . . . . 47
4.4.1 Payoffs for Fitness Value . . . . . 47
4.4.2 Virus and Enzyme Operations . . . . . 48
4.4.3 Bacterial Operation . . . . . 49
4.5 Conclusions . . . . . 50

**5 Effectiveness of the DNA Coding Method** **52**
5.1 Introduction . . . . . 52
5.2 Virus and Enzyme Operations . . . . . 53
5.3 Redundancy and Overlapping of Genes . . . . . 59
5.4 Examples of the Robot Behavior . . . . . 68
5.5 Bacterial Operation . . . . . 74
5.6 Conclusions . . . . . 81

**6 Summary** **82**
6.1 Summary of the Thesis . . . . . 82
6.2 Future Outlook of the Study . . . . . 85

**References** **87**

# Chapter 1

# Introduction

## 1.1 Genetic Algorithm

### 1.1.1 Outline

Genetic Algorithm (GA) [1-4] is a model for machine learning whose mechanism is derived from a metaphor of evolutionary process in nature. This algorithm is implemented in a machine by creating a population of individuals represented by a set of binary or character strings. This representation is analogous to the four-base chromosomes in biological DNA. Through selection, reproduction, crossover, and mutation like the biological evolution, the set of strings evolves to fulfill the tasks required of the machine.

Coding is the most important step in applying GA. There can be many ways to encode a solution of the problem, such as binary encoding, gray scale encoding, and real value encoding. There is no way which always outperforms others, since the coding way is problem dependent. Most of the application problems have a definite structure of the solution. However, there are some problems in which the structure of the solution is indefinite (e.g., the complex knowledge representation).

In this case, GA itself is expected to find out the structure of the solution by devising a coding way.

Reproduction is a process in which individual strings are copied according to their objective functions. This objective function is called fitness function. In practice, the fitness function is a measure of profit, utility, or goodness of individuals under a task environment. Copying of strings according to their fitness values means that individuals with higher fitness have higher probabilities to contribute to the next generation. Various ways have been reported to implement the reproduction. One of them is to create a biased roulette wheel where, for each current individual in the population, a roulette wheel slot is sized in proportion to its fitness. A simple spin of the roulette wheel selects an individual of the current population to yield a new offspring. In this way, more highly fit individuals have more offsprings in the succeeding generation.

In the process of crossover, a pair of chromosomes exchange chunks of genetic information. Crossover is carried out in two steps: the newly reproduced strings in the mating pool are mated at random. Then, each pair of strings undergoes crossover as follows:

An integer position $k$ along the string is selected at random in the domain of [1, $l$ - 1], where $l$ is the string length. Two new strings are created by swapping all the characters between position $k + 1$ and $l$.

Mutation, in simple GA, is occasional random alteration of the value at a certain string position. For example, in a binary coding, this simply means to change the value from 1 to 0, or vice versa. This operation is necessary because, even though the reproduction and crossover operations effectively search better combinations of strings, occasionally they lose some potentially useful strings and

they probably cause premature convergence. In artificial genetic systems, the mutation operator works against such misses of strings and local minima.

### 1.1.2 Advantages of Genetic Algorithm

Genetic Algorithm has been applied to a number of different applications [5-14]. Multidimensional optimization problem is a typical application of GA. Conventional search methods studied for the multidimensional optimization problems are calculus-based, enumerative, and random searches.

Calculus-based search is local in scope, the optimum it seeks is the best in a neighborhood of the current point, and it is not necessarily the global optimum. Moreover, the calculus-based methods assume the existence of derivatives of objective functions while many practical problems have difficulties in obtaining differentiable functions to optimize.

Enumerative schemes have been considered in many shapes and sizes of a objective function. The idea is fairly straightforward: looking at objective function value at every point in searching space. Although it is attractive because of the simplicity, it is lacking in efficiency. Many practical problems are simply too large to search the optimum in the search space in a practical time.

Random search has achieved increasing popularity as researches have shown the shortcomings of the above two methods. Yet, this algorithm is short of efficiency. In the later search of the long run, it is not expected to do better than enumerative schemes.

GA is different from these conventional optimization and search procedures in several aspects. GA searches from a population of points, not a single point. In

many optimization methods, a search point moves gingerly from a point in the search space to the next, using some transition rules to determine the next point. This point-to-point method is dangerous because it is probably trapped into a local minimum located in the multimodal (many-peaked) search space. By contrast, GA searches many points simultaneously, climbing a lot of peaks in parallel; thus, the probability of finding a false peak is smaller than that of the point-to-point method.

GA uses payoff information (fitness function), not derivatives of objective functions or other auxiliary knowledge. To perform an effective search for better structures, they only require payoff values (objective function values) associated with individual strings.

Conventional optimization methods can solve only problems which have predictable structures of solutions. GA, on the contrary, can find out the structure of the solution by contriving the way of coding. These characteristics make GA a more useful method than many other search schemes.

### 1.1.3 Modifications for GA

In general, GA is said that the ability of global search is high but that of local search is not. To solve this problem, modifications for GA have been studied.

#### Hybrid GA

One of the effective method to enhance the capability of the local search is hybrid GA [15-18]. The hybrid GA is the method combining GA with heuristic algorithms. In this method, GA is used for global searches and heuristic algorithms are used for local searches. M. Malek et al. proposed a combination of plural search

algorithms [16] for the improvement of precision and efficiency. D. Powell et al. achieved a fast search of solutions for design optimization by a combination of GA with an expert system [7] [17]. H. Kitano proposed a new weight training method of neural networks, which uses GA in early generations and Back Propagation (BP) in later generations [18]. He showed that the GA-BP method is faster than the simple BP learning.

**VEGA**

Schemata theory is well-known as a fundamental theory of GA [2]. Abundance of various types of effective schemata in the population enables the efficient search for the solution. A proportional selection and crossover of GA often cause a premature local convergence because an individual with the higher fitness value is selected many times. The proportional selection prevails not only effective schemata but also ineffective schemata in the population. N. Kubota et al. have proposed Virus-Evolutionary Genetic Algorithm (VEGA) [19-22] in order to transfer only effective schemata. This method is based on the virus theory of evolution, which is based on the view that virus transduction is a key mechanism for transporting segments of DNA across species [23].

**PBGA**

T. Furuhashi et al. have proposed an efficient method for local searches using a new GA technique. They have presented Pseudo-Bacterial GA (PBGA) [24-26] which is simple and very efficient in improving local portions of chromosomes. They have introduced mechanisms of genetic recombination in bacterial genetics [27-30] into GA. The PBGA is very efficient in finding fuzzy control rules.

## 1.2 Discovery of Fuzzy Rules

### 1.2.1 Fuzzy Inference

Fuzzy inference is based on fuzzy rules described in IF-THEN rules [31-34]. The process of the fuzzy inference used in this thesis is described in this subsection.

The fuzzy rule is generally expressed as:

$$R^i: \text{ IF } x_1 \text{ is } A_{i1} \text{ AND } x_2 \text{ is } A_{i2} \text{ AND } \ldots x_m \text{ is } A_{im} \text{ THEN } y = B_i \quad (i = 1, 2, \ldots, n)$$

where $R^i$ is the $i$ -th fuzzy rule and $A_{ij}$, $B_i$ are fuzzy variables. $n$ is the number of fuzzy rules. The following method is used to obtain the crisp output:

Fig.1.1 depicts an example of the inference method with three inputs $x_1$ - $x_3$, and three fuzzy rules $R^1$ - $R^3$. The triangular shapes in the figure are examples of membership functions. The truth value for each rule is derived as the minimum of the grades of memberships in the antecedent, and the height of the membership function in the consequent is reduced to the truth value. The inferred value $y^*$ is obtained by summing up the triangular membership functions in the consequent and by calculating the center of gravity of the summed functions.

Since the minimum operation is used for deriving the truth value of the fuzzy rule, the whole grades of the memberships $A_{ij}$ ($j = 1, 2, \ldots, m$) in the antecedent should have certain values so that the fuzzy rule is activated, i.e. the truth value is greater than zero.

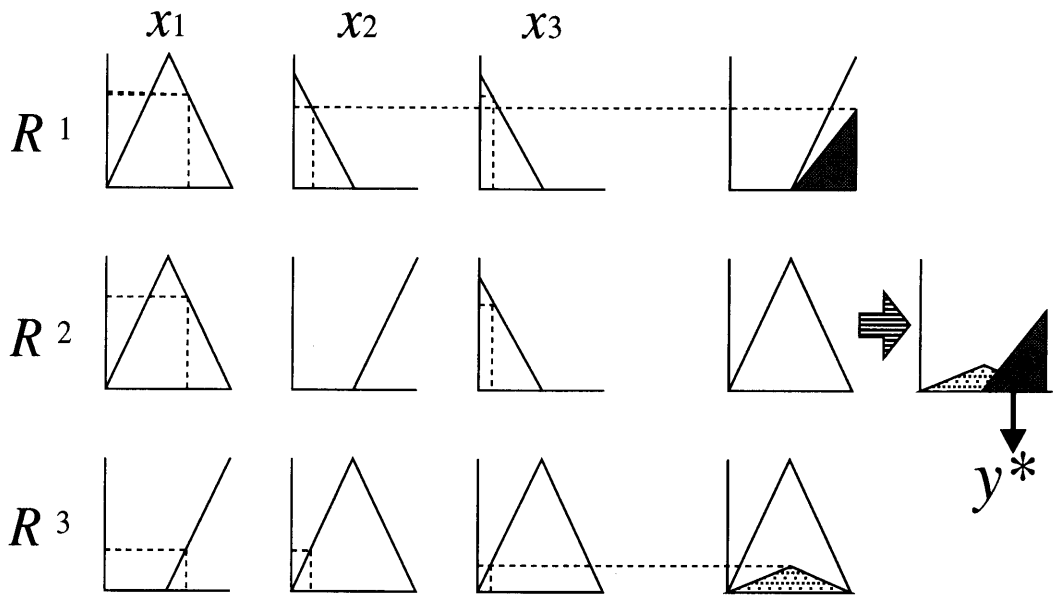



Fig.1.1 An Example of Fuzzy Inference

### 1.2.2 Application of GA to Discovery of Fuzzy Rules

Fuzzy rules can describe human knowledge using IF-THEN rules. Fuzzy controls, described in linguistic IF-THEN rules, have been widely used in industry for their high degree of performance in human-computer interactions. However, if the knowledge is not known in advance, it can not be represented by fuzzy rules. Therefore, GA has been applied to discovery of fuzzy control rules. C. L. Karr [35] [36] has proposed an application of GA to the design of fuzzy logic controllers, and his work was a pioneering effort in the application of GA to fuzzy controls. M. Valenzuela-Rendon [37] has proposed a fuzzy classifier system (FCS) by introducing fuzzy logic into the classifier system [38] and by applying the FCS to approximate a nonlinear function. T. Furuhashi, K. Nakaoka et al. [39-41] have studied applications of the FCS to knowledge finding for fuzzy controls. M. A. Lee and H. Takagi [42] have devised another interesting approach to the fusion of the fuzzy logic and GA. They have presented a method to control the parameters of GA, i.e. mutation rate, crossover rate, etc., by fuzzy logic. Selection of input variables for fuzzy control rules was addressed by T. Hashiyama et al. [25] using the efficient Pseudo-Bacterial GA (PBGA) [24] [26].

## 1.3 Research Goals

The above conventional methods encode the fuzzy rules directly using the position and width of membership functions or linguistic labels of membership functions, or real values of input/output parameters. This thesis presents a new coding method for GA based on biological DNA and a mechanism of development from the artificial DNA. This thesis calls this new coding method the "DNA coding method". The DNA coding method and the mechanism of development from the artificial DNA are suitable for knowledge representation. This method uses DNA itself and a way of development from DNA to a set of fuzzy rules. This thesis calls this chromosome the "DNA chromosome". Each fuzzy rule is translated from a part of the DNA chromosome, and the DNA chromosome has many redundant parts just as biological DNA. This method allows overlapped representation of genes, and each overlapping gene plays an important role. It has no constraint on crossover points, and the same type of mutation can be applied to every locus. The length of the proposed DNA chromosome is variable and it is easy to insert and delete parts of chromosomes. This thesis also presents a discovery of effective fuzzy control rules using the DNA coding method. This thesis presents a simple method of representing fuzzy rules developed from this DNA chromosome.

The fuzzy rules are discovered through controls of mobile robots which play chasing and avoiding. Selection of input variables and tuning of membership functions are done by this method. This thesis shows the effectiveness of the redundancy and overlapping of genes realized by this DNA coding method. The redundancy and overlapping of genes work well so that genes survive far beyond the life time of individuals. Effective fuzzy control rules of robots are discovered by

this method.

The VEGA described in subsection 1.1.3 uses some of the characteristics of the biological virus, however the process of the virus infection is purely an artificial one and special devices for each problem should be needed because of the constraint of this coding method. For example, the biological virus inserts its own DNA into the DNA host. However, the VEGA is applied to a Traveling Salesman Problem (TSP) as a typical optimization problem, and the infection of the VEGA transfers a strand of strings to other chromosomes. This operation makes duplicated cities in the recipient chromosomes. Therefore, other loci of the chromosome should also be modified by this infection. These devised operations are applicable only to this particular TSP problem. This thesis also shows the effectiveness of an application of virus and enzyme operators into the proposed DNA coding method. The enzyme operation is introduced in this thesis to cut the chromosome short. This is because the virus operation just makes the length of the chromosome longer. These virus and enzyme operations on the DNA chromosome can be applied easily with no difficulty to apply.

Though the PBGA applied the mutation operation to all of the parts of a chromosome, the division of the chromosome into the parts needed heuristics. This thesis presents a combination of the DNA coding method and the PBGA. The mutation of the PBGA in this combined method, which is called the "bacterial operation", is applied to meaningful parts of a chromosome. It means that just the activating fuzzy rules are changed. This combination of the DNA coding method with the PBGA accelerates the knowledge discovery process. The simulation results show how the bacterial operation is applied and how to change the genes (fuzzy

rules) by this operation. The change of a gene and the overlapped gene by this operation results in effective movements of the robots.

## 1.4 Composition of the Thesis

The DNA coding method and the combined method with the PBGA are applied to the discovery of fuzzy control rules. This thesis consists of six chapters. The background and the research goals are described in chapter 1. The DNA coding method is presented in chapter 2. The flow of development from the DNA chromosome, the genetic operations including virus and enzyme operations, and the features of the DNA coding method are also described in this chapter. In chapter 3, the biological way of transfer of bacterial DNA, the conventional way of the PBGA and the proposed combination of the DNA coding method and the PBGA are examined. Chapter 4 describes the problem formulation for knowledge discovery, which is a discovery of effective fuzzy control rules using the DNA coding method. The way of development from the DNA chromosome to the fuzzy rules and the concrete way of the genetic operations including virus, enzyme, and bacterial operations are presented. Chapter 5 shows simulation results of the knowledge discovery by chasing and avoiding operations of two mobile robots to show the effectiveness of this proposed method. The effectiveness of redundancy and overlapping of genes realized by the proposed method is also shown. The redundancy and overlapping of genes work well so that genes survive far beyond the life time of individuals. Chapter 6 summarizes this thesis and discusses the future prospects of this method.

# Chapter 2

# DNA Coding Method

## 2.1 Introduction

Genetic Algorithm [1-4] has been widely studied, and it has been applied to the discovery of fuzzy control rules [35] [36] [39] [43]. It is difficult, however, to acquire effective rules when the system becomes very large. Coding method has been one of the main issues to select effective input parameters from various candidates of the large system. Various coding methods have been studied [25] [26] [41] [46]. However, these GAs encode the fuzzy rules directly using the position and width of membership functions, or linguistic labels of membership functions, or real values of input/output parameters. Furthermore, GA usually uses a coding method which has no redundant parts. As a result, the genetic operations have some constraint to apply. For example, the crossover points are limited to the border of rules, the changeable range by mutation varies at each locus, and it is difficult to insert and delete random portions of chromosomes.

This chapter presents a new coding method for GA based on biological DNA and a mechanism of development from the artificial DNA. This thesis calls

this new coding method the "DNA coding method" [47-53]. The DNA coding method and the mechanism of development from the artificial DNA are suitable for knowledge representation. This method uses DNA itself and a way of development from DNA to a set of fuzzy rules. This thesis calls the chromosome used for this method the "DNA chromosome". Each fuzzy rule is translated from a part of the DNA chromosome, and the DNA chromosome has many redundant parts just as biological DNA. This method allows overlapped representation of genes, and each overlapping gene plays an important role. It has no constraint on crossover points and the same type of mutation can be applied to every locus. The length of the proposed DNA chromosome is variable and it is easy to insert and delete any parts of chromosomes.

This chapter describes the flow of development from the DNA chromosome to fuzzy rules, and genetic operations including virus and enzyme operations proposed. The features of the DNA coding method are also described in this chapter.

## 2.2 DNA Chromosome

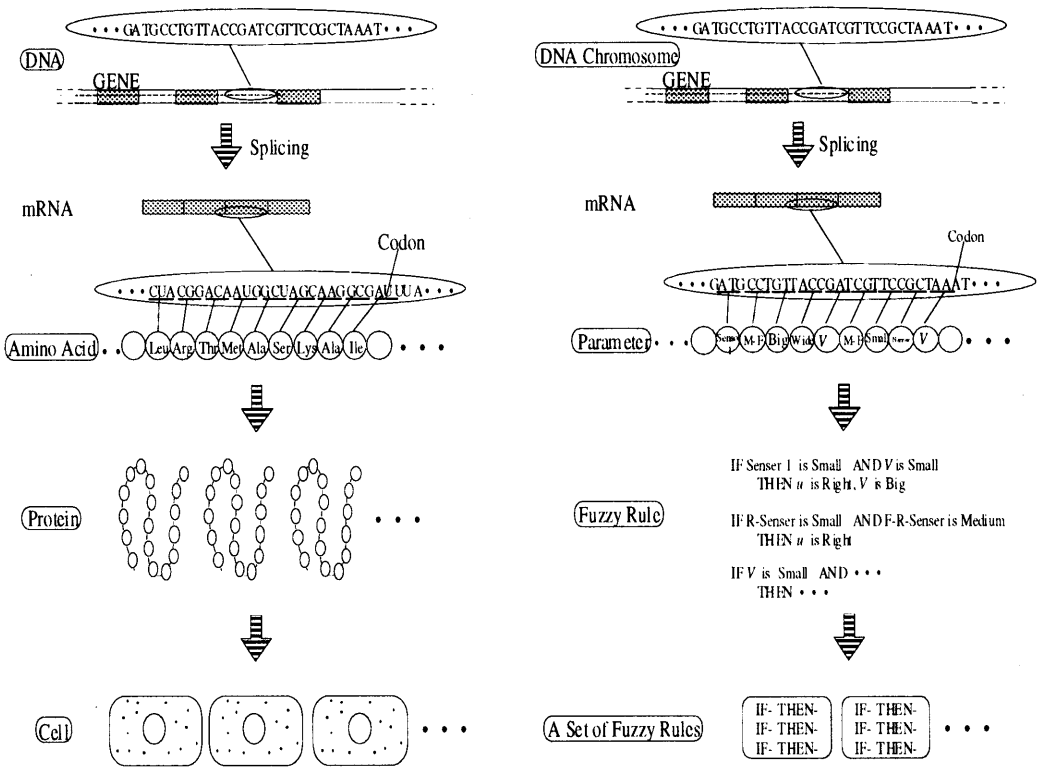
Fig.2.1(a) shows a flow from biological DNA to cells. The biological DNA consists of nucleotides which have four bases, Adenine(A), Guanine(G), Cytosine(C), Thymine(T) [28-30] [54]. Most of these bases in the top figure in Fig.2.1(a) are not used for the synthesis of proteins. A messenger RNA (mRNA), which has many unused parts, is first synthesized from DNA. In the synthesis of RNA, each base is translated into the complementary base i.e. T into A, G into C

and so on. Moreover in RNA the base U is used instead of T. Then the unused parts are cut out. This operation is a splicing. After this splicing has occurred, the mRNA is completed. Three successive bases called codons are allocated sequentially in the complete mRNA. These codons are the codes for amino acids. All of the 64 kinds of codons (4 bases$^3$) except end codons (TAA, TAG, and TGA) correspond to 20 kinds of amino acids. The details of translation into amino acids from codons are omitted here. This allocation of amino acids makes proteins, and proteins make up cells.

Fig.2.1(b) shows the DNA coding method and the flow of development to a set of fuzzy rules. This figure shows a correspondence of this method to the biological development. GA usually used a coding method specifically devised for each problem and it had no redundant parts. This conventional coding method could be regarded as a coding into the complete mRNA. The new coding method uses DNA itself and a way of development from DNA to a set of fuzzy rules. A chromosome consists of combinations of four bases, A, G, C, T. The chromosome has many redundant parts, and after the splicing, the mRNA is completed. In this artificial RNA synthesis, each base is translated into the same base. Moreover in RNA T is also used instead of U of the biological RNA. The codons in Fig.2.1(b) also correspond to amino acids. Unlike the biological amino acid, each artificial amino acid has several meanings, and the meaning of a gene is determined by the combination of the amino acids. An amino acid can be translated to be an input variable, a form of a membership function, and so on. A sequence of amino acids makes a fuzzy rule. One gene corresponds to one fuzzy rule. The DNA chromosome having several genes makes up a set of fuzzy rules for controlling a mobile robot.

Fig.2.2 shows an example of the DNA chromosome and its translation mechanism. In this figure a gene starts from the start codon ATG, and ends at the end codon TAG, and codons in the gene are translated into amino acids: Tyr, Thr, •••. Each amino acid has its own role for the problem.

The biological ϕX174 phage, bacteriophage, has 11 genes in DNA. In this phage the starting point is shifted and some genes are overlapping on other genes, and each overlapping gene plays an important role [28] [30]. By the new mechanism of development from DNA, the starting point can be also shifted from a base to another and some genes overlapping on other genes can be translated. Each overlapping gene plays an important role. Fig.2.3 shows this overlapped representation. In this figure, GENE5 in addition to GENE3 and GENE4 can be read from the DNA chromosome. This DNA chromosome has redundancy and also compresses information by the overlapping of genes.




(a)Biological Development (b)Artificial Development

Fig.2.1 Flows of Development from the DNA Chromosome

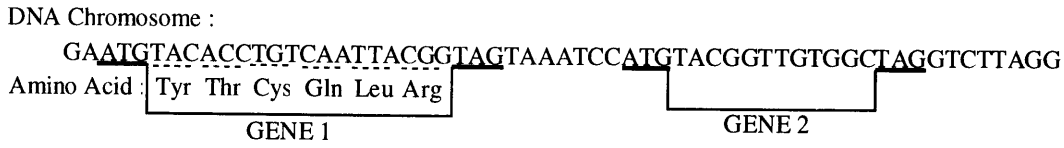



Fig.2.2 An Example of the DNA Chromosome and its Translation Mechanism

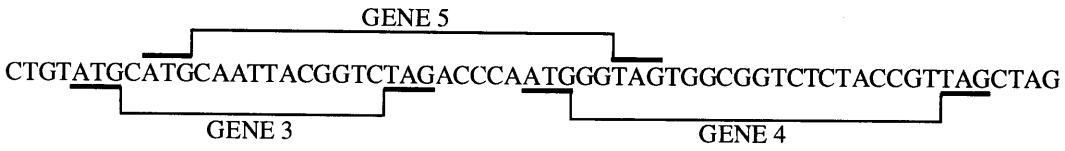
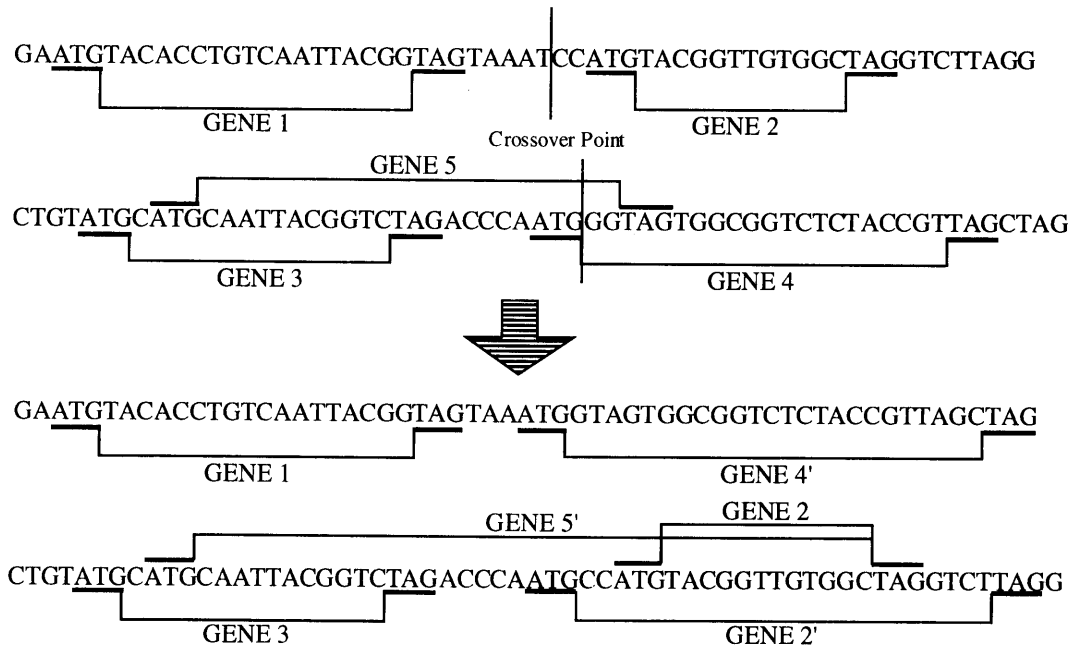



Fig.2.3 Overlapping of Genes

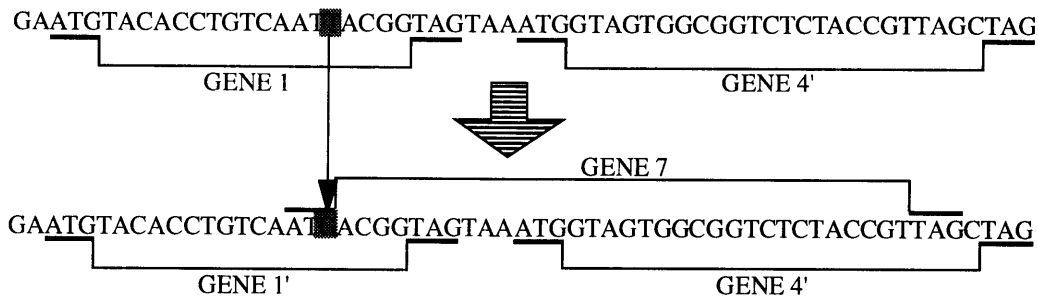
## 2.3 Genetic Operations

Fig.2.4 shows examples of crossover and mutation. Fig.2.4(a) is an example of one point crossover. Right hand sides of the crossover points are exchanged and new GENE2', 4', 5' are generated.

Fig.2.4(b) shows an example of mutation. One base indicated in the figure is changed from T to G. As a result, GENE1 is changed to GENE1'. The start codon ATG is newly generated and new GENE7 is read. The mutation can be done equally to every locus by changing each base. The changes of bases in the redundant parts by this mutation are accumulated. Once these parts are used, this accumulation of change is expected to make the phenotype drastically change just in the same way as the neutral mutation in biology [55].




(a)Crossover




(b)Mutation

Fig.2.4 Crossover and Mutation

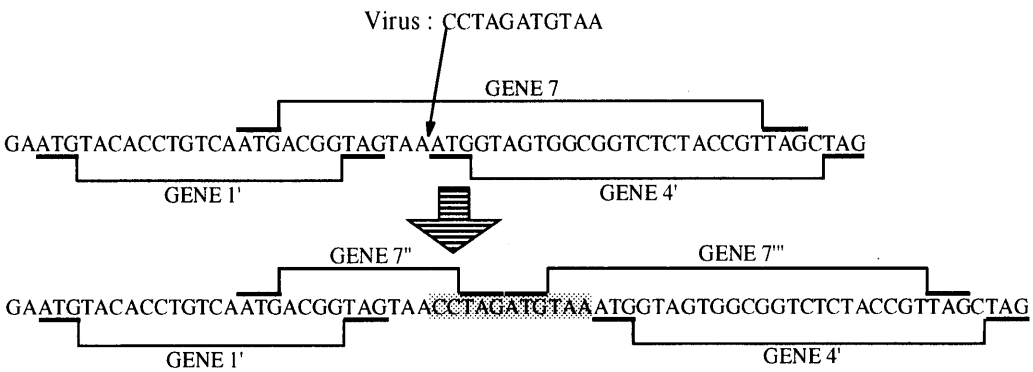
## 2.4 Virus and Enzyme Operations

N. Kubota et al. have proposed Virus-Evolutionary Genetic Algorithm (VEGA) [19-22]. This method is based on virus theory of evolution, which is based on the view that virus transduction is a key mechanism for transporting segments of DNA across species [23]. This method uses some of the characteristics of the biological virus, however the process of the virus can not be completely realized and special operations are required to use this method because of the constraint of the coding. For example, the VEGA was applied to a Traveling Salesman Problem (TSP) which was a typical conventional optimization problem [56]. The infection by the VEGA transfers a strand of strings to other chromosomes. This operation makes duplicated cities in the recipient chromosomes. Therefore, other loci of the chromosome should also be changed by this infection.

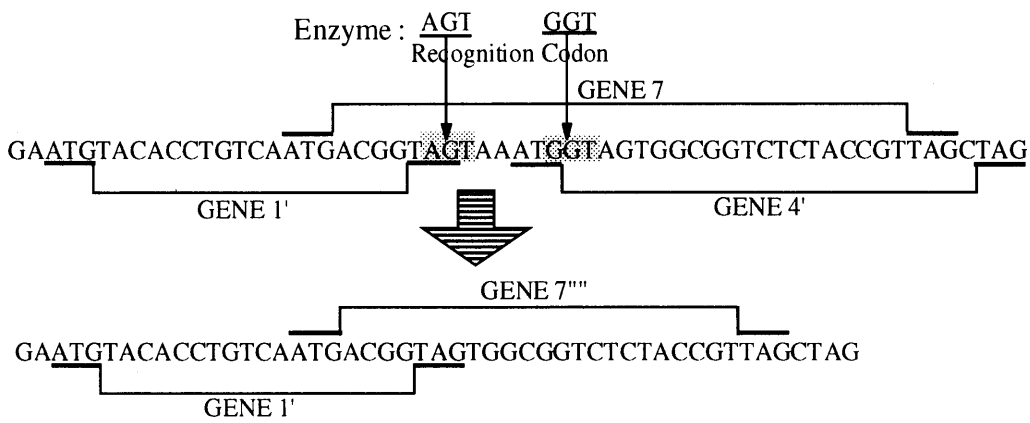
It is easy to insert and delete parts of chromosomes represented by the DNA coding method. The process of the virus infection can simply be realized by the application of virus operator. Furthermore, since the length of the chromosome is made longer and longer by the virus operation, the author proposes a new operation which is called enzyme operation.

Fig.2.5 shows examples of the virus and enzyme operations. The biological function of virus and enzyme is not known completely. The artificial virus and enzyme are applied in the following way: Fig.2.5(a) shows a virus operation. A part of sequence of DNA from other DNA chromosome moves into the chromosome. As a result, the two genes of GENE7", GENE7"' are generated from GENE7. Fig.2.5(b) shows an enzyme operation. The enzyme distinguishes two amino acids, and splices the part between the two. In this figure the part between

the codon AGT (Ser) and the codon GGT (Gly) is cut out, and GENE4' is deleted and GENE7"" is produced from GENE7.




(a)Virus




(b)Enzyme

Fig.2.5 Virus and Enzyme Operations

## 2.5 Features of the DNA Coding Method

The DNA coding method has the following features:

(a)Flexible representation of knowledge is realizable.

This artificial chromosome consists of four bases and three continuous bases called codons correspond to amino acids. Each amino acid has several meanings, and each parameter of fuzzy rules i.e. an input/output variable, a form of a membership function, and so on is determined by the combination of the amino acids. The correspondence between amino acids and the parameters of fuzzy rules can be determined flexibly, and the DNA chromosome is translated into a set of fuzzy rules flexibly. Therefore, the flexible knowledge representation given by fuzzy rules is realizable.

(b)The coding is redundant and overlapped.

The DNA coding method uses DNA itself and the DNA chromosome has many redundant parts just the same as biological DNA. Usually these redundant parts are not used for synthesis of fuzzy rules. Potential and inactive genes are probably accumulated in these redundant strings. The chromosome can adapt itself to varying environment by having these redundant parts. The neutral theory of molecular evolution in biology is also expected to be observed in the simulation using this method. This method allows overlapped representation of genes, and each overlapping gene plays an important role. The DNA chromosome has compressed information by the overlapping of genes. The overlap of genes means inter-dependence of plural genes, and a change of a gene causes another change of overlapping gene. The details of effectiveness of redundancy and overlapping of

genes are shown in chapter 5.

(c)The length of the chromosome is variable.

The DNA chromosome generates fuzzy rules by translating the genes from start codons in the chromosome. The number of genes in a chromosome is not fixed. The length of the DNA chromosome is changeable. In addition to this flexible representation of fuzzy rules, it is easy to insert and delete parts of chromosomes. As a result, the genetic operations including virus and enzyme operations can also be applied easily with no difficulty to apply.

(d)No constraint on crossover points is imposed.

The conventional coding has constraint on crossover points because crossover operations cause overlaps of the same variable in one rule. So conventional GA limits the crossover points to the border of rules or the same points in a rule in both chromosomes. This proposed method has no constraint on crossover points even if any type of crossover is used.

## 2.6 Conclusions

This chapter presented a new coding method, DNA coding method, and a mechanism of development from the artificial DNA. This DNA coding method and the mechanism of development from the artificial DNA are suitable for knowledge representation. Selection of input variables and tuning of membership functions are done. The DNA chromosome has redundancy and it can be translated flexibly.

The flow of development from the DNA chromosome was shown with the correspondence to the biological development. The new coding method uses DNA itself and a way of development from the artificial DNA to a set of fuzzy rules. The DNA chromosome has many redundant parts, and allows overlapped representation of genes. This DNA chromosome has compressed information as well as redundant ones.

Examples of the genetic operations were also shown in this chapter. The length of the chromosome is variable, and it had no constraint on genetic operations. The virus and enzyme operations were proposed. These operations also have no difficulty to apply because it is easy to insert and delete parts of chromosomes. The proposed coding method makes these genetic operations easy.

# Chapter 3

# Combination of the DNA Coding Method and Pseudo-Bacterial GA

## 3.1 Introduction

Fuzzy controls, described in linguistic IF-THEN rules, has been widely used in industry for its high degree of performance in human-computer interactions. Demand for fuzzy inference systems which can describe complex, multi-input/output systems is growing. The difficulties associated with the fuzzy inference include the discovery of fuzzy rules and the tuning of membership functions. To address these problems, genetic algorithm (GA) [1-4], one of the basic models of evolution and an effective tool for constructing evolvable/adaptive complex systems, has been applied [35] [36] [39] [43] [44].

T. Furuhashi et al. have proposed an efficient method for finding fuzzy rules using a new GA technique. They have presented Pseudo-Bacterial GA (PBGA) [24-26] which is simple and very efficient in improving local portions of chromosomes. They have introduced mechanisms of genetic recombination in

bacterial genetics [27-30] into GA. In this study, the PBGA is applied to find fuzzy rules for an obstacle avoidance problem involving a mobile robot. The simulation result shows that the PBGA is very efficient in finding control rules with limited information from the payoffs provided by the task environment.

This chapter presents a combination of the DNA coding method and the PBGA [57] [58]. The PBGA proposed in [24-26] applied the mutation operation to all of the parts of a chromosome, and the division of the chromosome into the parts needed heuristics. In the DNA chromosome, a gene corresponds to a fuzzy rule. Accordingly, the PBGA mutation applied only to the activating fuzzy rules is expected to be effective in improving local portions of chromosomes. This combination of the DNA coding method with the PBGA accelerates the knowledge discovery process.

This chapter describes the biological way of transfer of bacterial DNA. Bacteria can transfer its own DNA from male cells to female cells. By this transfer, the characteristics of one bacterium can be spread among the entire bacteria population. This chapter describes the conventional way of the PBGA, and the combination of the DNA coding method and the PBGA.

## 3.2 Transfer of Bacterial DNA

Bacterial genetics provides interesting mechanisms for genetic recombination [27-30]. Bacteria can transfer DNA to recipient cells through mating. Male and female cells are distinguished by the presence of a distinct supernumerary sex chromosome called F (fertility) factor [28]. When it is present as a discrete body, the cells are male ($F^+$) and capable of transferring genes into female cells. In its absence, the cells are female ($F^-$) and act as recipients for gene transfer from male cells. The F factor is a circular double-helical DNA molecule present in one copy per cell. Male cells possess F sex pili on its surface to attach to female cells. Transfer of most bacterial genes from male to female cells only occurs when, through a crossing-over process, the $F^+$ factor becomes integrated into the cell's single main chromosome. Such Hfr (high frequency of recombination) cells remain male, possess F sex pili, and fuse with $F^-$ cells as frequently as $F^+$ cells do. When an Hfr cell joins to an $F^-$ cell, conjugation-induced replication of F DNA begins, and because the leading edge of the $F^+$ factor is now attached to the main chromosome, transfer of the main chromosome follows. Fig.3.1 shows the biological way of transfer of bacterial DNA from an Hfr (male) cell to a female cell. Hfr cells, which have F factor, transfer strands of chromosomes into female cells as explained above. Then the female cells acquire characteristics of the Hfr cells and F factor, and finally change into Hfr cells. By these means, the characteristics of one bacterium can be spread among the entire bacteria population. Fig.3.2 shows the DNA transfer process of bacteria. When bacteria are reproduced, some bacteria are mutated. The bacteria which can adapt the environment transfer its own DNA to female cells with F factor. Then the female cells become the Hfr cells. By this

transfer, more adaptable bacteria can spread their DNA strands among the population.

Bacteriophages carry a copy of the host gene across and incorporate it into the chromosome of the infected cell. This process is called transduction. By the transduction, it is also possible to spread the characteristics of a single bacterium among other bacteria. These genetic recombination have led to a mechanism of microbial evolution [27]. Mutated genes can be transferred from a single bacterium to others and effect rapid evolution.

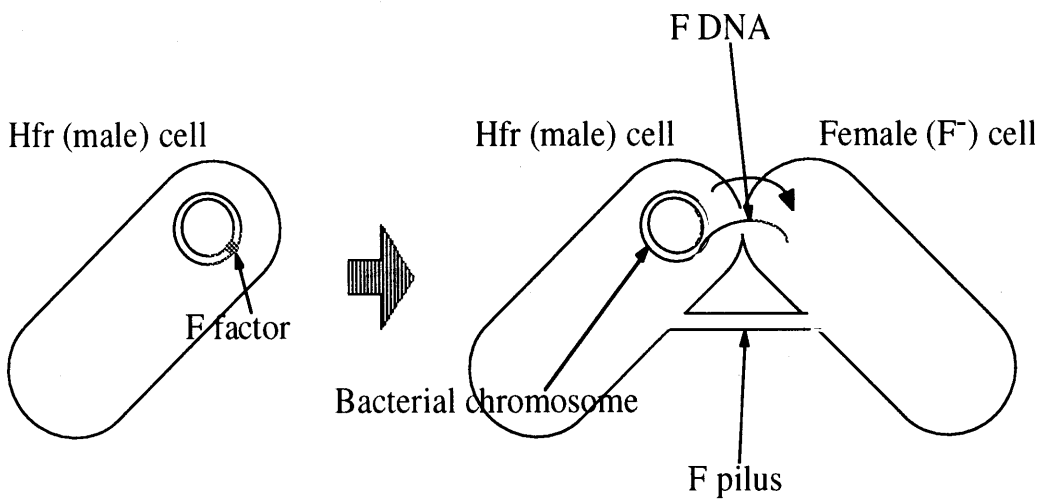



Fig.3.1 Transfer of Bacterial DNA from an Hfr Cell to a Female Cell

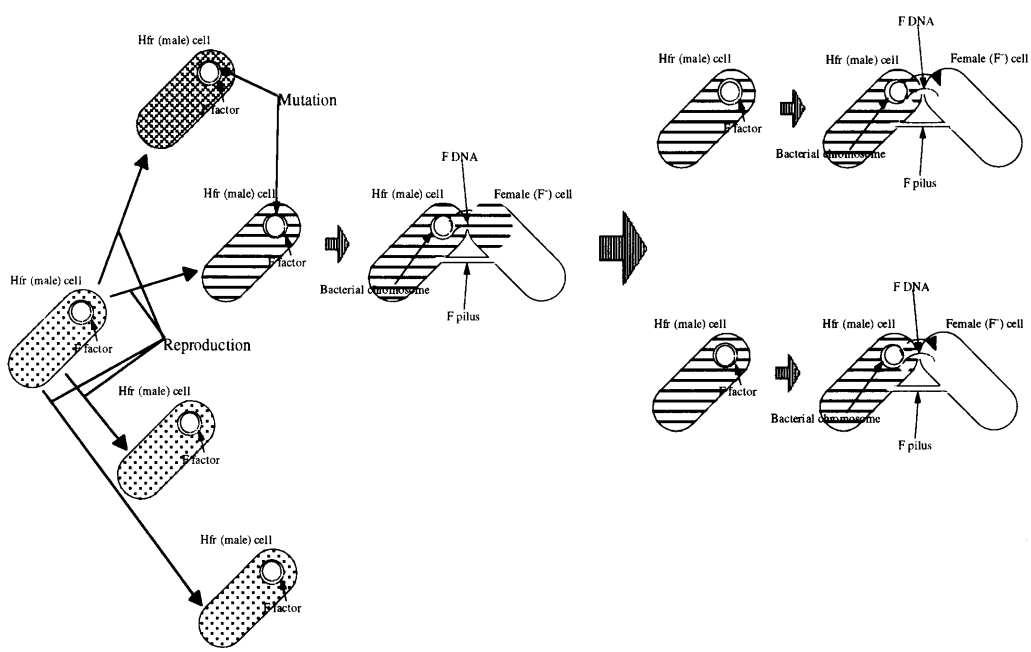



Fig.3.2 DNA Transference Process of Bacteria

## 3.3 Conventional PBGA

T. Furuhashi et al. have introduced the mechanisms of the bacterial genetics described in 3.2 into GA. Multiple bacteria are reproduced and some genes of each chromosome are mutated and tested. The best genes are chosen and transferred to other bacteria. The PBGA streamlines this mechanism in the extreme.

Fig.3.3 shows the basic flow of the conventional PBGA.

(1) Generation of the initial population

$n_{chr}$ individuals are generated randomly. Each chromosome is evaluated.

(2) Genetic operators

The following genetic operations are applied to individuals and new populations of chromosomes are generated:

(i) Mutation and Selection of genes

Suppose there are $n_p$ parts in a chromosome. One chromosome "1" is chosen and this is reproduced to $m$ clones. Part $i$ ($i$ is randomly decided) of $m$ - 1 clones are mutated. Each chromosome is evaluated. The elite among $m$ chromosomes is selected and the rest are deleted. The above process - reproduction, mutation, evaluation, and selection - is repeated. The mutation is applied to a randomly chosen part, excluding previously chosen parts, and a new chromosome "1'" is finally obtained. This genetic operation is applied to all the $n_{chr}$ chromosomes one by one.

The above process can be interpreted as follows:

A bacterium is reproduced to $m$ clones and the same parts of their chromosomes are mutated. The elite bacterium is selected and transferred to other $m$ - 1 bacteria, and

inferior parts are replaced with elite one. Then other parts of the chromosomes are mutated, evaluated, selected, and transferred. All the elite parts are aggregated to a chromosome of one bacterium. Bacteria evolve in this way.

(ii) Selection and reproduction of chromosomes

The selection and reproduction steps are applied to the chromosomes. Each chromosome has its fitness evaluated through the above operations. Those with lower fitness values are deleted. Some chromosomes randomly chosen from the remaining chromosomes are reproduced. In fig.3.3, chromosome $k'$ is deleted, since its fitness value is the least. The chromosomes $i'$, $j'$ are selected and reproduced.

(iii) Crossover

The crossover operation is applied to the newly generated chromosomes and the offsprings $i''$, $j''$ are generated and evaluated. This is an operation of the conventional GA and is also efficient for improving the chromosomes.

(3) Stopping condition

If a stopping condition is not satisfied, go back to (2). If satisfied, stop. The stopping condition is, for example, a certain value or a number of generations.

The above genetic algorithm is efficient in the local improvement of chromosomes, since the evolution is carried out on the level of chromosomal parts.

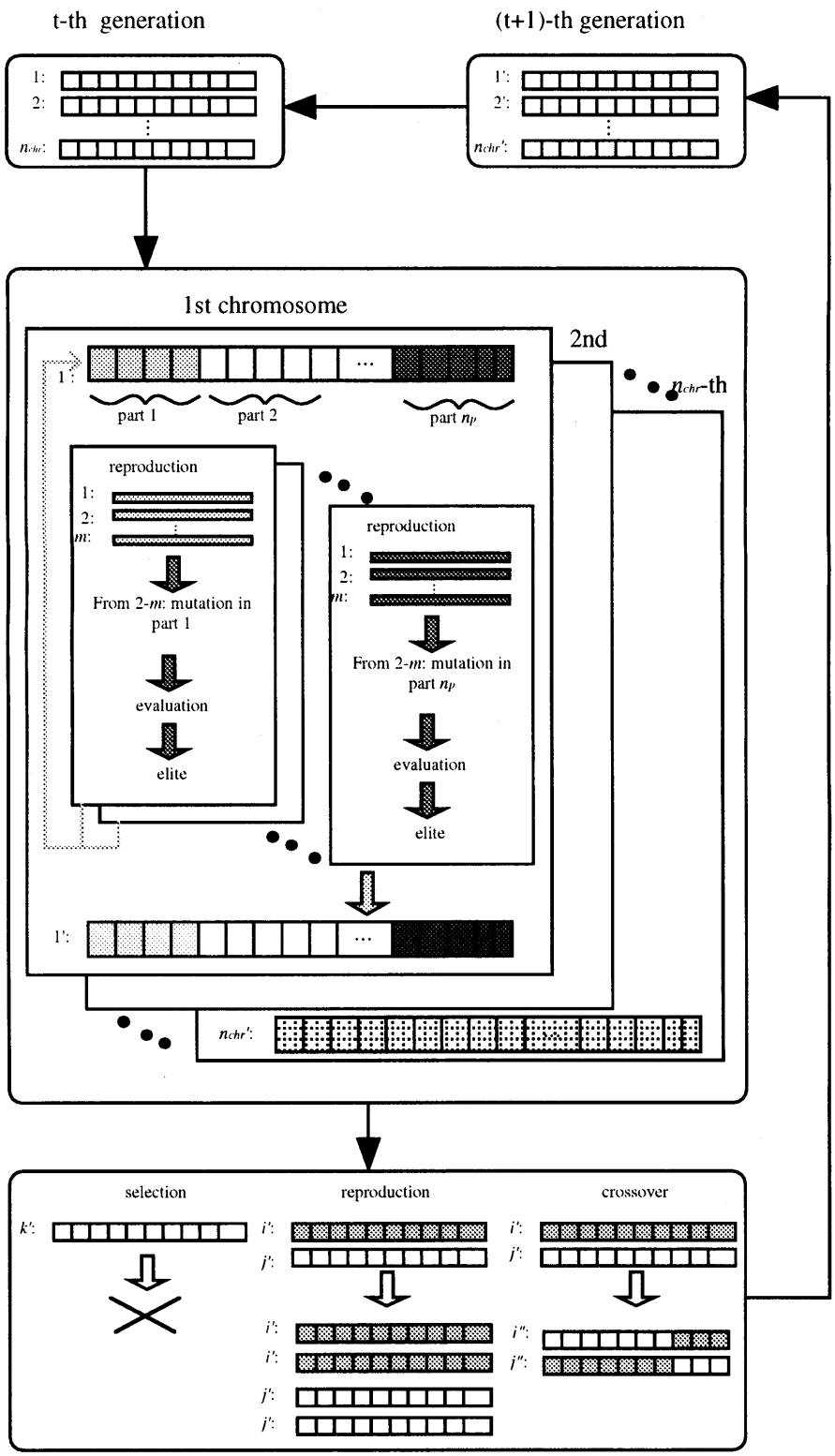



Fig.3.3 The Flow of the Conventional PBGA

## 3.4 Combination of the DNA Coding Method and the PBGA

The author has combined the DNA coding method with the PBGA. The PBGA applies the mutation to all the separated parts of a chromosome. However, any effective way of separation is not shown in [24-26]. The author applies the mutation of the PBGA to meaningful parts. This combined method applies the mutation of the PBGA to genes each of which correspond to a fuzzy rule. The mutation of the PBGA is applied not to all parts or genes in a chromosome but only to $n$ genes. This combination of the DNA coding method with the PBGA accelerates the knowledge discovery process.

From one chromosome, multiple bacteria are reproduced, and the same gene in each new born bacterium is mutated. The best gene among the bacteria is chosen and transferred to other bacteria. In this thesis this operation is called "bacterial operation". Fig.3.4 shows an example of the bacterial operation of this combined method. In this figure, GENE1 is chosen and reproduced to four clones. The clones except one are mutated. The elite among them is selected and the rest are deleted. This operation is applied to randomly selected genes of the chromosome, and to the whole population one by one. After this operation, the conventional selection, reproduction, crossover, and mutation operations are applied.

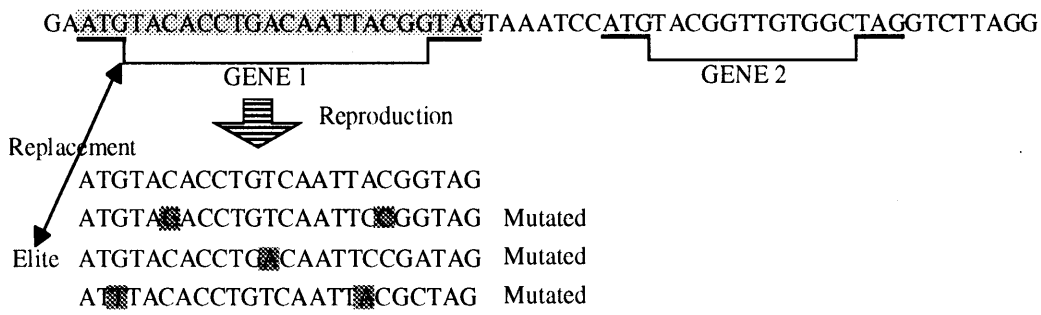



Fig.3.4 An Example of the Bacterial Operation

## 3.5 Conclusions

This chapter described Pseudo-Bacterial GA (PBGA). The PBGA is simple and very efficient in improving local portions of chromosomes. The biological way of transfer of bacterial DNA was shown. Bacteria can transfer their own DNA from male cells to female cells through transfer of F factor. New bacteria whose parts of DNA are mutated are tested in the environment, and the bacteria which can adapt themselves to the environment can survive. By these process, the characteristics of more adaptable bacteria can be spread among the entire bacteria population.

The author has combined the DNA coding method with this PBGA. This combination was also shown in this chapter. The conventional PBGA has applied the mutation to all of the separated parts of a chromosome. However, any effective way of separation is not shown. The author applied the mutation of the PBGA to genes each of which corresponded to a fuzzy rule. Not all the parts of chromosome are the subject to this operation. This combination of the DNA coding method with the PBGA is expected to accelerate the knowledge discovery process.

# Chapter 4

# Application of the DNA Coding Method

## 4.1 Introduction

This chapter, first, describes the problem formulation for knowledge discovery, which is a discovery of effective fuzzy control rules using the DNA coding method. These fuzzy rules are used to control mobile robots which play chasing and avoiding in the area surrounded by walls. The robot which reaches or avoids the other robot receives more payoffs from the environment. Considering these payoffs as fitness values, the genetic operations are applied to the chromosomes, and the fuzzy rules are evolved. Selection of input variables and tuning of membership functions are done.

This chapter, second, describes the way of development from the DNA chromosome to the fuzzy rules. The correspondence between amino acids and the parameters of fuzzy rules is given. Like the biological process, 64 kinds of codons correspond to 20 kinds of amino acids. One gene which starts from a start codon

corresponds to one fuzzy rule. Each fuzzy rule is translated from the DNA chromosome according to the translation rule.

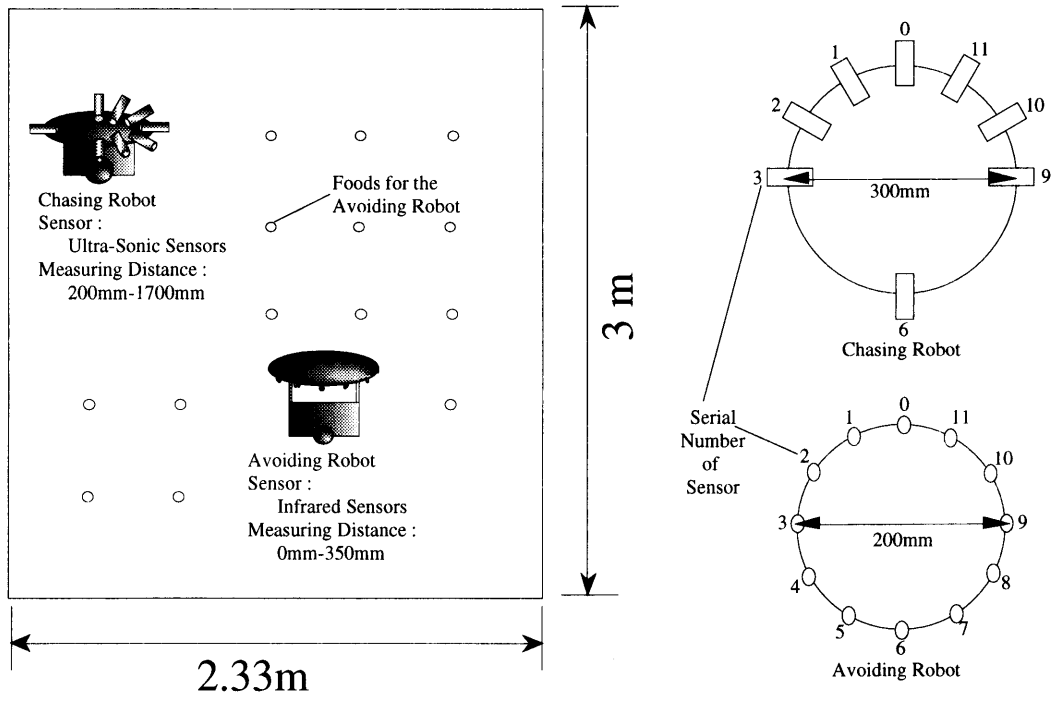
The concrete way of genetic operations including virus, enzyme, and bacterial operations is also described. The payoffs for fitness values of the chromosome are given from the environment according as the movements of each robot having a set of fuzzy rules translated from the DNA chromosome are effective or not. Using these payoffs of each chromosome, selection and reproduction are done. The virus and enzyme operations are carried out by inserting and deleting a part of a chromosome.

## 4.2 Problem Formulation for Knowledge Discovery

The effects of the new method are demonstrated through the evolution of fuzzy rules. Fig.4.1 shows the simulation conditions of the problem and the construction of two robots. Two different types of mobile robots play chasing and avoiding in the area of 2.33m wide and 3m long surrounded by walls. There are several food in the area for the avoiding robot, however the avoiding robot cannot recognize the food in the area. The radius of the chasing robot is 150mm, and that of the avoiding robot is 100mm. The chasing robot has eight ultra-sonic sensors (seven in the front, and one in the rear). These sensors can measure the distances between obstacles and themselves in the range of 200mm to 1700mm. The measured time is 0.1 seconds per a sensor. So it needs about one second to measure all directions. Therefore the chasing robot must acquire rules to catch the other robot

avoiding crashes into the walls using the measured values one second before. The avoiding robot has twelve infrared sensors to see around. These sensors can measure limited distance less than 350mm. Therefore the avoiding robot cannot recognize the approaching robot until the enemy comes near to it. The avoiding robot must find fuzzy rules not to be caught by moving quickly. The serial number of the front sensor of each robot is No.0. That of the next sensor in the counterclockwise side is No.1. Each robot has a DNA chromosome containing a set of fuzzy rules. The fuzzy rules steer and accelerate/decelerate the robot to chase/avoid the other robot and stay away from the walls.

The robot which reaches or avoids the other robot and stays away from the walls receives more payoffs from the environment. Considering these payoffs as fitness values, the genetic operations are applied to the DNA chromosomes of the robots, and the fuzzy rules are evolved.




(a)Simulation Conditions (b)Top of View of Robots

Fig.4.1 The Simulation Conditions and the Construction of Two Robots

## 4.3 Representation of Rules

The candidates for the input variables of each robot are the detected values of sensors $D$ and the velocity of a robot $V$, and those for the output variables are the steering angle $u$, and the velocity $V$. The DNA chromosome has a set of fuzzy rules which are represented by IF-THEN rules. The chromosome determines combinations of input/output variables and membership functions of each fuzzy rule. The central position $x_c$ and the width $\sigma$ of the membership functions are also encoded into the chromosome. The detail way of fuzzy inference in this method is shown in chapter 1.

In the biological DNA, a gene starts from the start codon ATG, and ends at the end codon TAA, TAG, or TGA. In this thesis, a gene also starts from the start codon ATG which corresponds to IF. The end codon is not definitely determined. A gene consists of the codons between IF codon and some related codons succeeding to THEN codon. When there is no rule in a chromosome, the robot having that chromosome moves only straight forward. Even if the lengths of all the chromosomes in the population are too short to have a rule, the robots in the population will get rules by the genetic operations i.e. crossover, mutation, and so on. Reading from the top of the DNA chromosome, the translation to a fuzzy rule starts upon finding the start codon ATG. As described in chapter 2, the overlaps of genes are allowed in the DNA chromosome. After reading a fuzzy rule, re-reading is restarted from the second base of the IF-codon and a new IF codon is sought. Fig.4.2 shows the flow of translation from the DNA chromosome into a fuzzy rule, and Table 4.1 shows the correspondence between amino acids and the parameters defined in this section. Like the biological process, 64 kinds of codons correspond

to 20 kinds of amino acids. The meaning of each amino acid is determined by its position in the sequence of amino acids. For example, (1) When Phe is in the next position of the start codon ATG, its meaning is that the input variable is the sensor. (2) When Tyr follows Phe, the meanings of Tyr becomes that the number of sensors is 2. (3) In this case, succeeding two amino acids determine the sensors to be used. In the same way, the meaning of amino acids in the sequence are determined by the translation rules in Table 1. Fig.4.3 shows an example of the DNA chromosome and genes (fuzzy rules). In this figure, the bases are read from the head of the DNA chromosome, and if the start codon ATG is found, a fuzzy rule starts from this part. In this example, the next codon GCT is Alanine, and Alanine here means that the input variable is sensor. The sequence of Alanine, Serine, Leucine means that this sensor is No.0 sensor (input variable is $D_0$). The next part of Glycine, Cysteine determines the form of membership functions, the central position $x_c$ and the width $\sigma$, for the input $D_0$. The next codon GCC makes also Alanine, and this Alanine here means AND of the fuzzy rule. Like this, each amino acid has several meanings, and one meaning is selected based on the position in the gene. In this example, GENE2 in addition to GENE1 can be read from the DNA chromosome. The amino acids for virus and enzyme operations are defined in subsection 4.4.2.

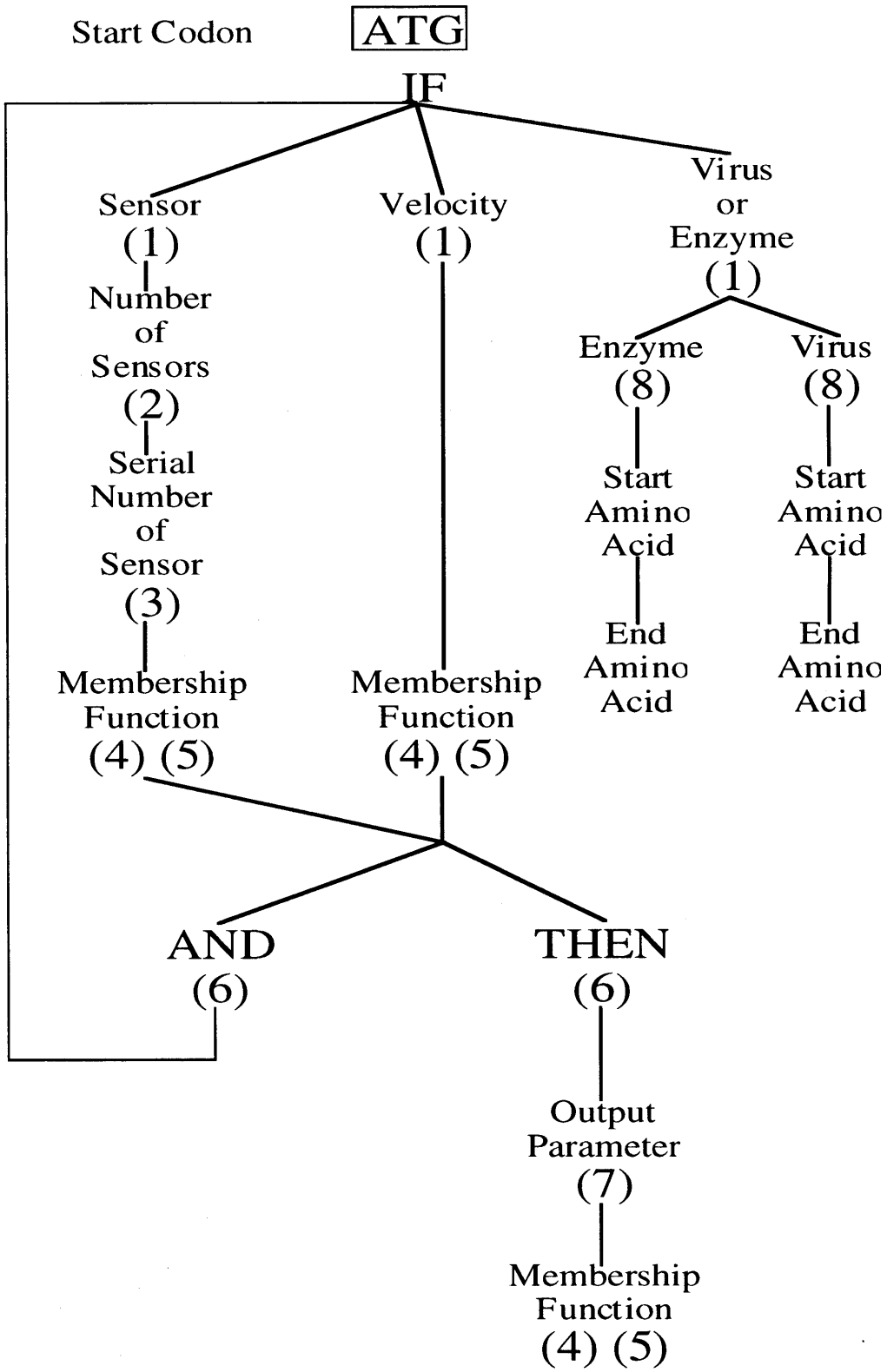



Fig.4.2 The Flow of Translation from the DNA Chromosome into a Fuzzy Rule

Table 4.1 The Correspondence between Amino Acids and the Parameters

| (1)-(8) \ Amino Acid | *Phe* | *Leu* | *Ile* | *Val* | *Ser* | *Pro* | *Thr* | *Ala* | *Tyr* | *His* | *Gln* | *Asn* | *Lys* | *Asp* | *Glu* | *Cys* | *Trp* | *Arg* | *Gly* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (1) Input Parameter | Sensor | | | | | | | | | | | | | | | | | *1 | *2 |
| (2) Number of Sensors | 1 | | | | | | | | 2 | | | | | | 3 | | | | 4 |
| (3) Serial Number of Sensor | 0 | | 3 | | 9 | | 2 | 10 | 1 | | 11 | | 4 | 8 | 5 | 7 | 6 | | |
| (4) Central Position of Membership Fuction $x_c$ | Left ◄ | | | | | | | | | | | | | | | | | | ► Right |
| (5) Width of Membership Fuction $\sigma$ | Narrow ◄ | | | | | | | | | | | | | | | | | | ► Wide |
| (6) AND or THEN | AND | | | | | | | | THEN | | | | | | | | | | |
| (7) Output Parameter | Steering Angle $u$ | | | | | | | | Velocity $V$ | | | | | | | | $u$ and $V$ | | |
| (8) Virus or Enzyme | Enzyme | | | | | | | | Virus | | | | | | | | | | |

*1: Velocity *2: Virus or Enzyme

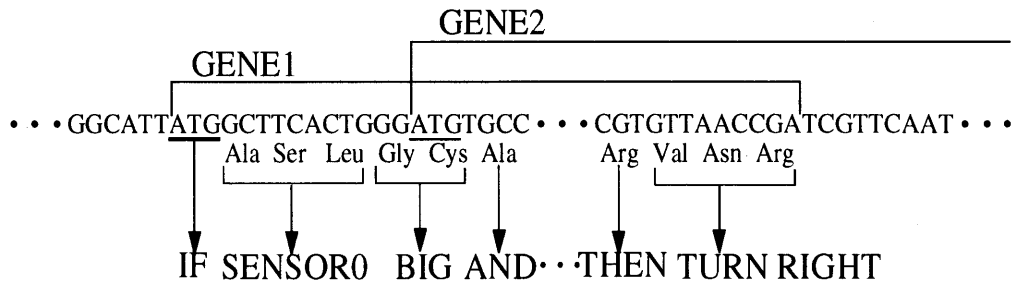



Fig.4.3 An Example of the DNA Chromosome and Genes

## 4.4 Genetic Operations

### 4.4.1 Payoffs for Fitness Values

For the genetic operations, plural robots are needed. There are seven chasing robots and seven avoiding robots in one generation. Each robot, either chasing or avoiding, is tested twice by randomly choosing its opponents from the seven counterparts. Each robot has initial payoffs $E_1$. Each test ends when either of the two confronting robots crashes into the wall, the chasing robot catches the other, or a certain amount of time has passed. The payoffs for fitness values of the chromosomes are given as follows:

If the robot crashes into the wall, the robot loses payoffs $E_w$ ;

If the chasing robot reaches the avoiding robot within a fixed time, the chasing robot receives payoffs $E_c$ , conversely the avoiding robot loses payoffs $E_a$ ;

If the chasing robot does not reach the avoiding robot in a fixed time, the chasing robot loses payoffs $E_m$ ;

If the avoiding robot reaches the foods, the avoiding robot gains payoffs $E_f$ per a food.

After these tests are done, the genetic operations are applied to the chromosomes of the robots by regarding the payoffs of the robots as their fitness values. One chromosome of chasing robot which has the smallest payoffs is deleted. Two chromosomes which are selected from the remaining six robots are reproduced, and one-point crossover shown in section 2.3 is applied to them to

generate a new chromosome for a chasing robot. The chromosome of avoiding robot is newly generated in the same way. There is no constraint on the crossover points as described in chapter 2. The mutation operator is also applied to the newly generated chromosomes. The mutation operation can be done simply by changing the bases. This mutation is done to each base at a rate of $P_m$. The payoffs of all the robots are reset at $E_1$ again. After another simulation for the new generation is done, the genetic operations are applied again. These steps are repeated, and fuzzy rules which control the robots to chase or avoid the other robot and to stay away from the walls are expected to be evolved.

### 4.4.2 Virus and Enzyme Operations

Each chromosome is made to contain virus and enzyme rules. The flow of translation from the DNA chromosome to these virus and enzyme rules in this thesis is also shown in Fig.4.2. The translation rules are defined in Table 1. If one of codons GG* which make the amino acid Gly is found in the DNA chromosome, the next codon decides the virus or enzyme rule. The third and fourth codons determine the start amino acid and the end amino acid of the virus or enzyme, respectively. If more than two rules are in one chromosome, the first virus or enzyme rule will be used. The virus is copied from other randomly selected DNA chromosome. A virus starts from the start amino acid and ends at the end amino acid determined by the virus rule and moves into the chromosome. The codon ATG must be included in between the start amino acid and the end amino acid. By this virus operation, at least one more rule is added into the DNA chromosome.

An enzyme cuts the part of the chromosome between the start amino acid

and the end amino acid. The codon ATG must also be included in this part. If the payoffs of the robot becomes less in the next generation compared with that in the previous generation, the changed part by the virus or enzyme operation restored to the previous part. The virus or enzyme operation onto the part will be prohibited for the future generations. The chromosome on which the virus or enzyme operation was once applied has immunity of that virus or enzyme to prevent the same virus or enzyme from changing the chromosome again.

### 4.4.3 Bacterial Operation

This section describes the combination of the DNA coding method and the PBGA. The bacterial operation is applied to each chromosome one after another. In chapter 3, Fig.3.4 shows the case where the bacterial operation is applied to one gene and the number of the clones are four. In this thesis, the number of the genes and clones are defined by the parameters $n$, $m$ and this operation is applied as follows:

From a chromosome, one gene is randomly chosen and reproduced to $m$ clones. To the $m$ - 1 clones, the mutation is applied at the rate of $Pm$ per a base. However that means possibility of disappearance of an impediment or useless rule. Each clone is recombined with the rest of the chromosome and evaluated. The evaluation of each chromosome having one clone is done in the same way as in section 4.3.

Each of $m$ clones is tested one by one. The elite among $m$ clones is selected and the rest are deleted. The above process: reproduction, mutation, evaluation and selection is repeated to randomly selected $n$ genes of the chromosome. After this bacterial operation is done to all of the chromosomes, the conventional genetic

operations described in section 4.3 are applied to the population of chromosomes by regarding the payoffs of the robots as their fitness values.

## 4.5 Conclusions

This chapter described the problem formulation for discovery of effective fuzzy control rules. These fuzzy rules are used to control mobile robots. The simulation conditions and the construction of two robots which play chasing and avoiding were shown. The robot whose movement with a set of fuzzy rules translated from the DNA chromosome is effective receives more payoffs from the environment. Considering these payoffs as fitness values, the genetic operations are applied to the chromosomes, and the fuzzy rules are evolved. Selection of input variables and tuning of membership functions are done.

This chapter also described the way of development from the DNA chromosome to the fuzzy rules. The flow of translation from the DNA chromosome and the correspondence between amino acids and the parameters were defined. One gene which starts from a start codon corresponds to one fuzzy rule. Each fuzzy rule is translated from the DNA chromosome according to the translation rule defined in this chapter.

The way of genetic operations including virus, enzyme, and bacterial operations was also described in this chapter. The virus and enzyme operations are done in the simple way by inserting and deleting a part of a chromosome. The bacterial operation is carried out through the process: reproduction of genes,

mutation of genes, evaluation and selection of genes. After this bacterial operation is applied to all of the chromosomes, the conventional genetic operations are applied to the population of chromosomes by regarding the payoffs of the robots as their fitness values.

# Chapter 5

# Effectiveness of the DNA Coding Method

## 5.1 Introduction

Simulations were done, and the effectiveness of the proposed DNA coding method and the mechanism of development from the artificial DNA was examined. This chapter describes the simulation results. The value of each parameter was as follows:

The length of each initial DNA chromosome was 500. The payoffs $E_l$, $E_w$, $E_c$, $E_a$, $E_m$, $E_f$ were 0, 20, 50, 20, 20, 5, respectively. Though these parameters were determined after several trials, the effects of changing these parameters were little. The probability of the mutation $P_m$ was 0.05. The numbers of genes and clones $n$, $m$ for the bacterial operation were 1, 6, respectively, in this thesis. The chasing robot which could not reach the avoiding robot within 30 seconds lost payoffs $E_m$. The genetic operations were applied to the chromosomes of chasing and avoiding robots for 100 generations alternately. If the genetic operations are applied to both

the robots simultaneously, the environment for each robot changes too rapidly to adapt itself.

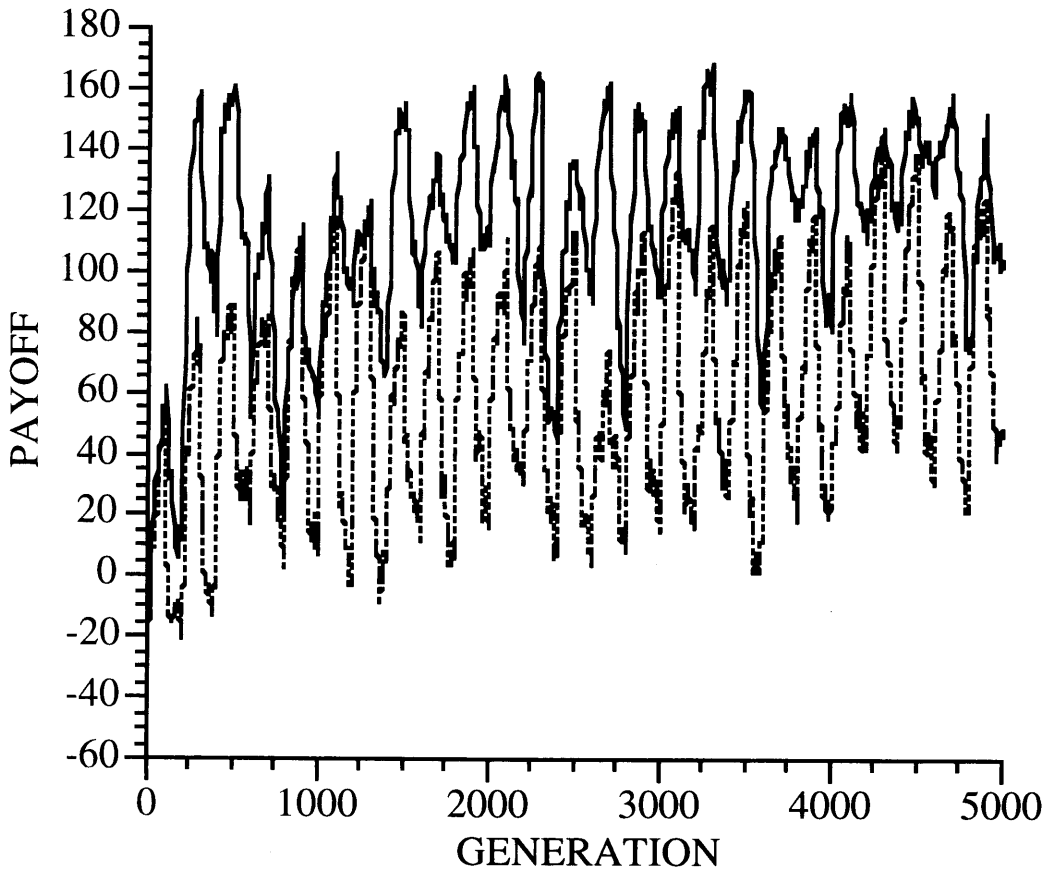
The effectiveness of virus, enzyme, and bacterial operations was also tested under the confrontation of the two mobile robots. The effectiveness of redundancy and overlapping of genes realized by the proposed method is also shown in this chapter. The redundancy and overlapping of genes work well so that genes survive far beyond the life time of individuals. Some improvements of robots' behavior observed in the simulations were examined. A small change of a gene could cause a drastic improvement in the performance of the robots.

## 5.2 Virus and Enzyme Operations

First, the effectiveness of the DNA coding method was examined. Fig.5.1(a) shows the averages of payoffs of 7 chasing robots until 5000th generation and Fig.5.1(b) shows those values obtained during the 3000-5000th generation. The solid line is the case (i) where the chasing robots used the proposed coding method and the avoiding robots used the conventional method in [25]. The dotted line is the case (ii) where the chasing robots used the conventional method and the avoiding robots used the proposed coding method. In both the cases, no virus, enzyme, nor bacterial operations were used. Each is the average in 10 trials. In a trial, the simulation from the initial generation to the 5000th generation was done. Since the genetic operations were applied to the chromosomes of chasing and avoiding robots for 100 generations alternately, the payoffs fluctuated periodically. The parameters of the genetic operations of the conventional method were tuned and

the result in Fig.5.1 was the best among obtained. This figure shows the advantage of the new coding method.

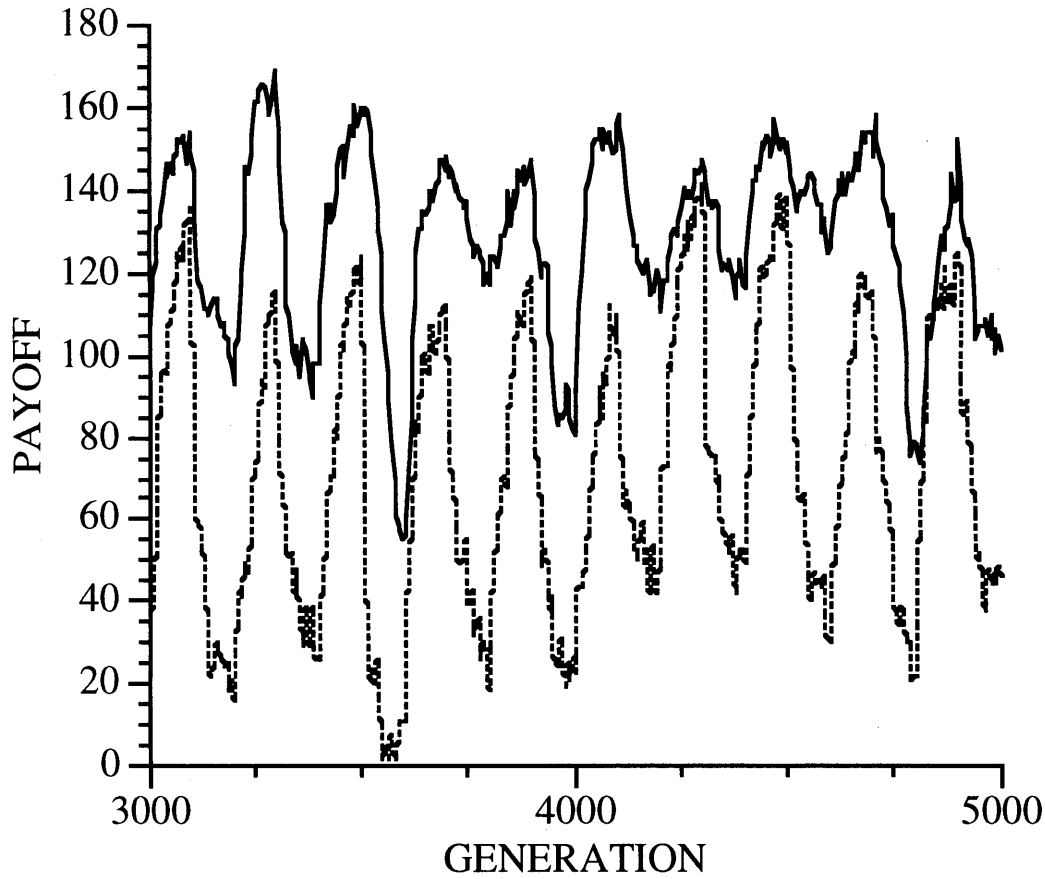
Next, the effectiveness of the virus and enzyme operations was examined. Fig.5.2(a)(b) shows the averages of payoffs of 7 chasing robots in 10 trials. The solid line is the result of the case (iii) where the chasing robots used the virus and enzyme operators and the avoiding robots used no virus nor enzyme operators. The dotted line is that of the case (iv) where the chasing robots did not use the virus nor enzyme operators and the avoiding robots did. In both the cases, no bacterial operation was used. The virus and enzyme operations worked well.




(a)Until 5000th Generation

Fig.5.1 Payoffs of Chasing Robots (10 Trials)

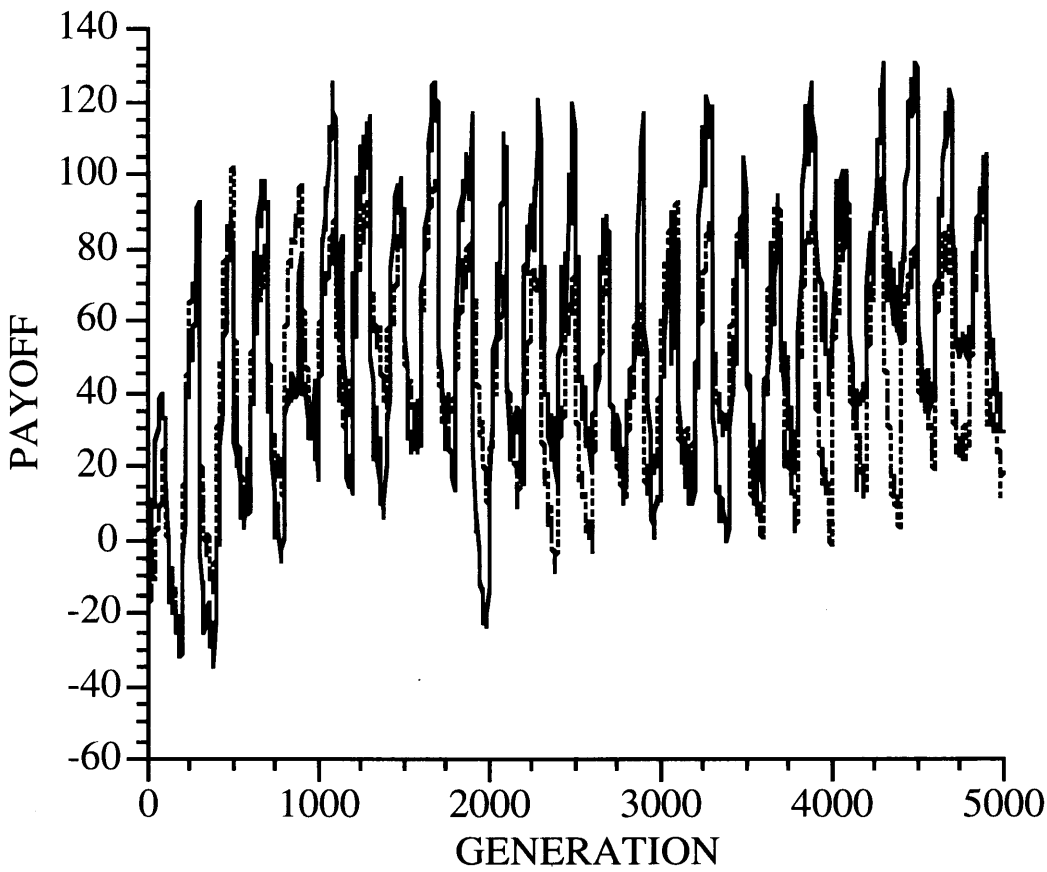
(Effects of the DNA Coding Method)




(b)During the 3000-5000th Generation

Fig.5.1 Payoffs of Chasing Robots (10 Trials)

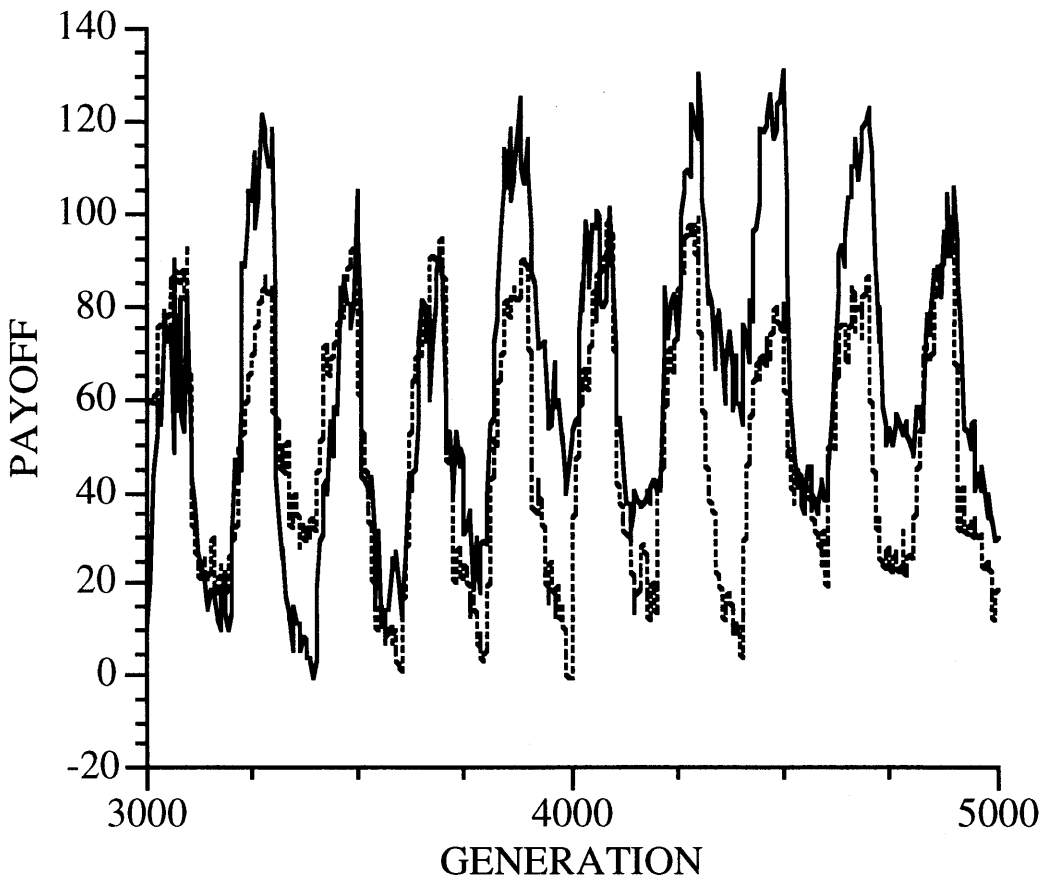
(Effects of the DNA Coding Method)




(a)Until 5000th Generation

Fig.5.2 Payoffs of Chasing Robots (10 Trials)

(Effects of the Virus and Enzyme Operations)




(b)During the 3000-5000th Generation

Fig.5.2 Payoffs of Chasing Robots (10 Trials)

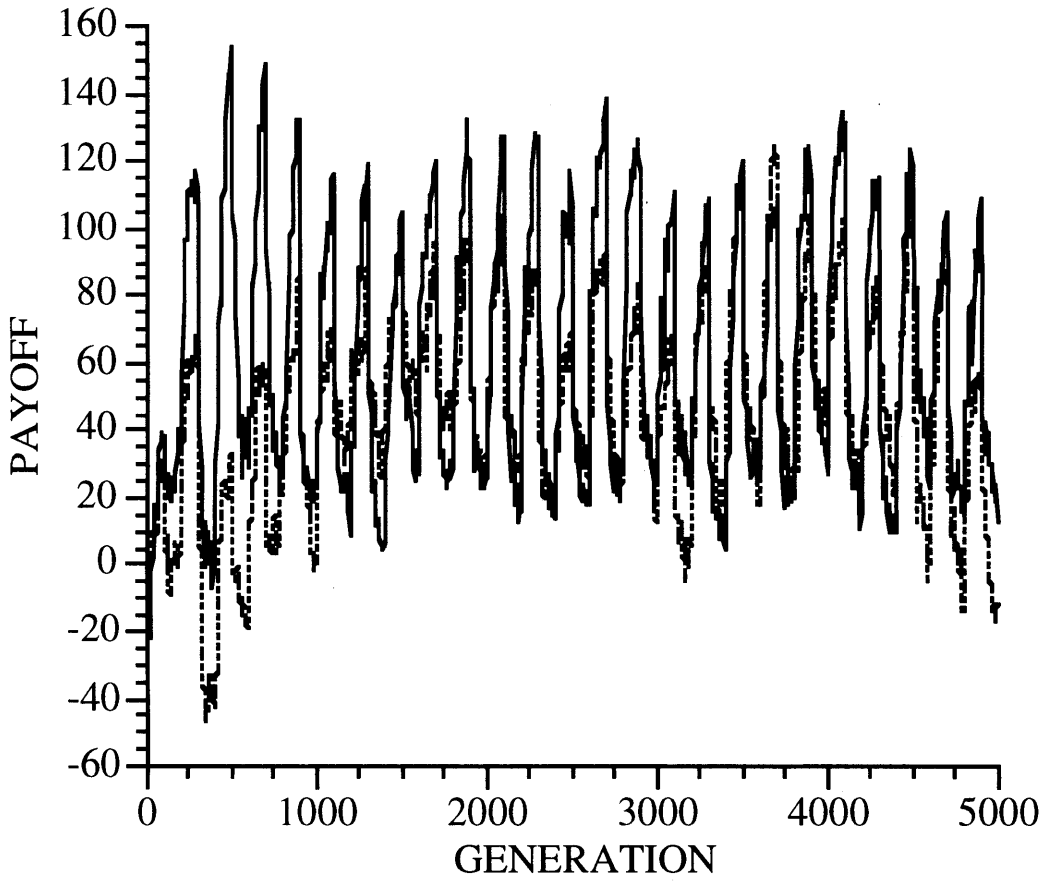
(Effects of the Virus and Enzyme Operations)

## 5.3 Redundancy and Overlapping of Genes

The author examined the effectiveness of the redundancy and overlapping of genes. Fig.5.3(a)(b) shows the averages of payoffs of 7 chasing robots in 10 trials. The solid line is the case (v) where the chasing robots used the proposed coding method with redundancy and overlapping of genes and the avoiding robots did not. The dotted line is the case (vi) under the condition vice versa. In the case of no redundancy and overlapping, no start codon was used. The DNA chromosome was translated from its head. In both the cases, no virus, enzyme, nor bacterial operations were used. This figure shows the effectiveness of the proposed coding method having redundancy and overlapping of genes. The alternating period of the application of the genetic operations to the chromosomes of chasing and avoiding robots was changed from 100 to 200, 300, and 500. The results showed the same effectiveness of the redundancy and overlapping of genes. The effects of the redundancy and overlapping of genes were investigated further. Fig.5.4 shows life time of the genes survived for more than 300 generations in one of the simulations in Fig.5.3. One gene corresponded to a fuzzy control rule which had truth value larger than 0.1 in the chasing or avoiding control. The horizontal axis is the generation. The vertical axis is the type of genes. The higher the density of the line, the more the number of the genes of same type in the population. Fig.5.4(a) shows an example of the case where (A) the genes or chromosomes used the DNA coding method with redundancy and overlapping of genes. Fig.5.4(b) shows another example of the case where (B) no redundancy nor overlapping was used. The genes in the early generations disappeared in a short time, and the genes in the later generations survived for longer generations. Fig.5.5 shows durations of each

chromosome while it was unchanged. Fig.5.5(a) and (b) show the cases (A) and (B), respectively. The horizontal axis is the generation and the vertical axis is the serial number of chromosomes. From Fig.5.5(a), it is known that the durations of chromosomes were shorter than the long lived genes in Fig.5.4(a). Some genes in Fig.5.4(a) survived for thousands of generations. The chromosomes did not live longer than one thousand generations. The disappearance of chromosomes did not mean the disappearance of genes. It is also interesting to note that many genes were activated again after long inactive time. These genes were kept in chromosomes, even though they did not work. Fig.5.4(b) and Fig.5.5(b) show a different story. From the early generations, some genes lived for a very long time. The chromosomes in this case also lived very longer than in the case (A). The deletion of the chromosomes meant the disappearance of the genes. The inredundant coding in this case worked against quick generation of effective sets of genes. The average numbers of the genes which lived for more than 300 generations were 588 in the case (A), 236 in the case (B), respectively. Each was the average in 10 trials. In the case (B), it was difficult to generate new chromosomes which had better combinations of chromosomes and were able to overcome the existing ones. Thus, most of the genes other than those shown in Fig.5.4(b) were short lived. Fig.5.6(a)(b)(c)(d) show examples of genes in a DNA chromosome at the 300th, 1000th, 2500th, and 5000th generations, respectively. The robot having the chromosome received the highest payoffs at the generation. The black parts are the genes, i.e. fuzzy control rules, which had truth values larger than 0.1 in the chasing or avoiding control. In Fig.5.6(a), most of the genes especially the overlapping genes were not used yet. In Fig.5.6(b), though some of the overlapping genes were used, most of the genes were not used. In Fig.5.6(c), the length of the chromosome

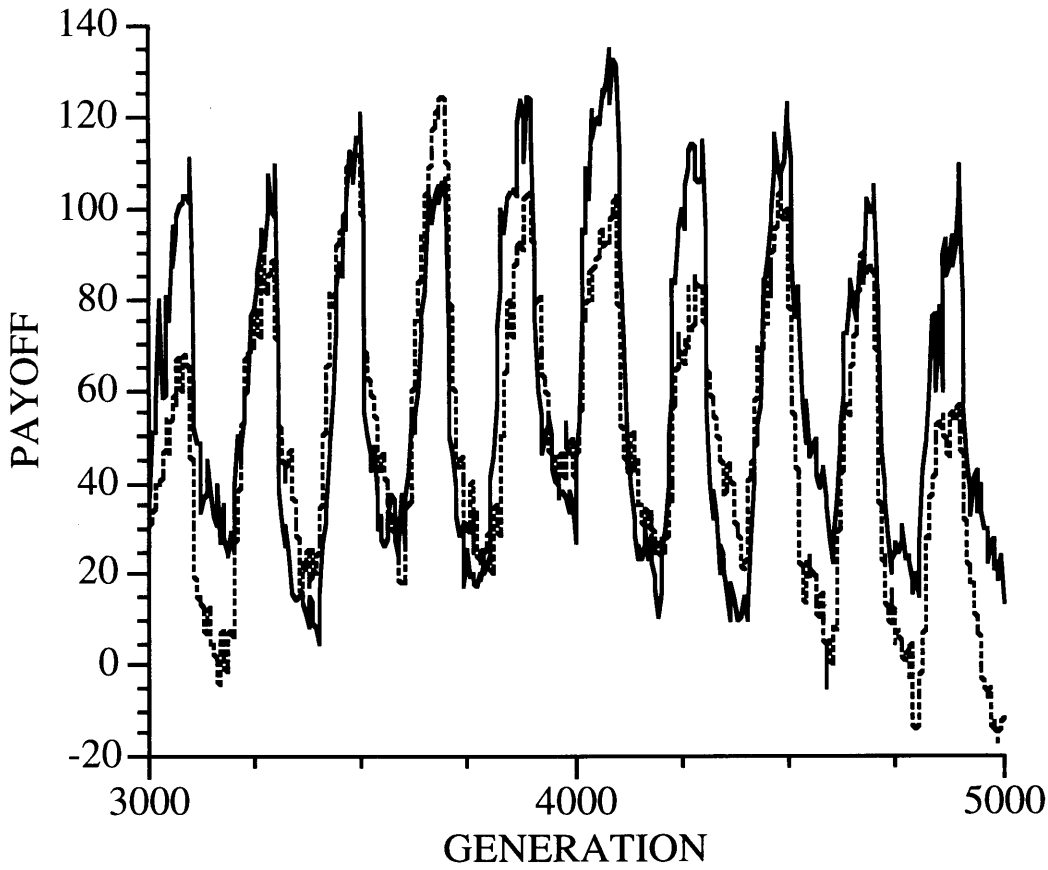
became very long, and many overlapping genes were used. In Fig.5.6(d), useless rules had been cut out, and the length of the chromosome became short. Fig.5.7 shows the averages of (a)the number of the genes, (b)the number of the genes which had truth values larger than 0.1, (c)the number of the overlapped genes in the chromosomes. This is the average in 30 trials. This figure shows the similar results shown in Fig.5.6.




(a)Until 5000th Generation

Fig.5.3 Payoffs of Chasing Robots (10 Trials)

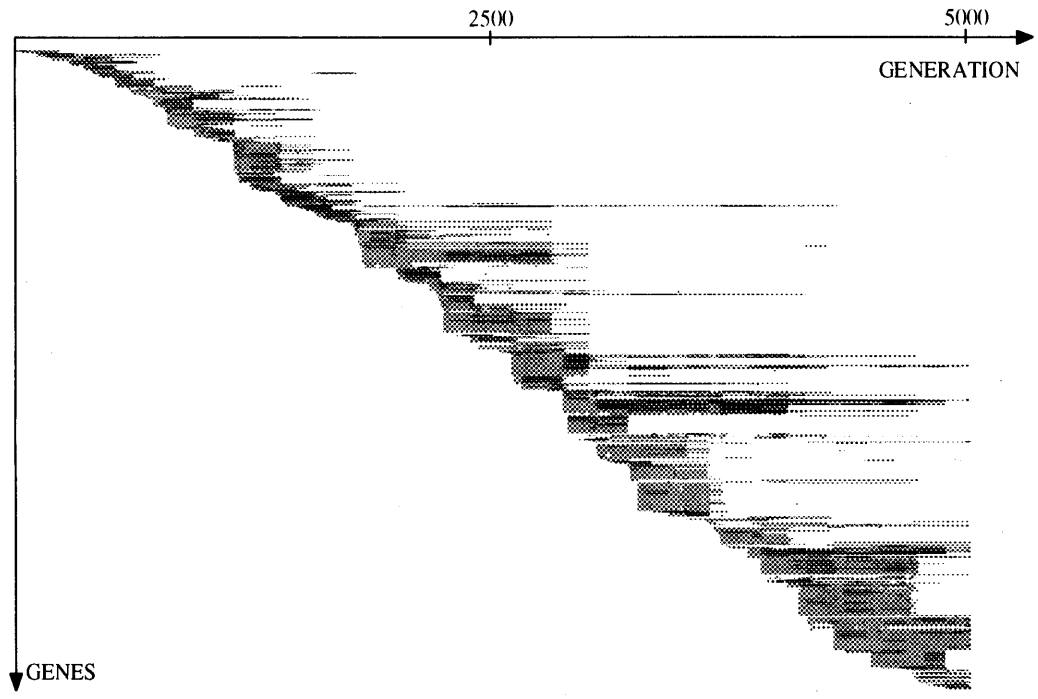
(Effects of Redundancy and Overlapping of Genes)



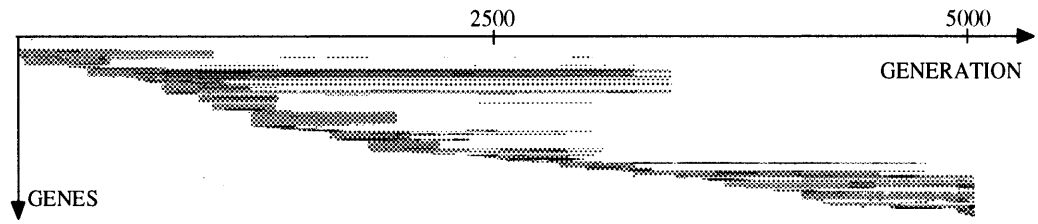

(b)During the 3000-5000th Generation

Fig.5.3 Payoffs of Chasing Robots (10 Trials)

(Effects of Redundancy and Overlapping of Genes)

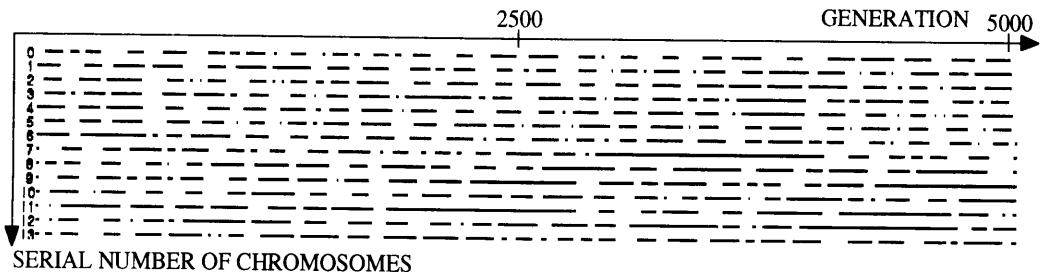



(a)Redundancy and Overlapping

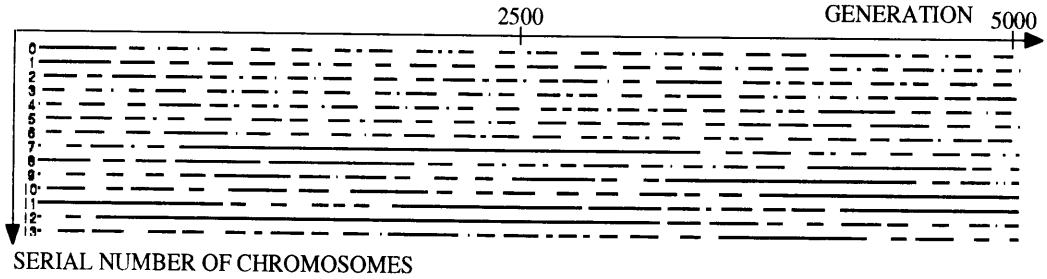



(b)No Redundancy nor Overlapping

Fig.5.4 Life Time of Genes

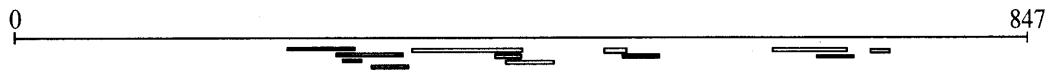



(a)Redundancy and Overlapping

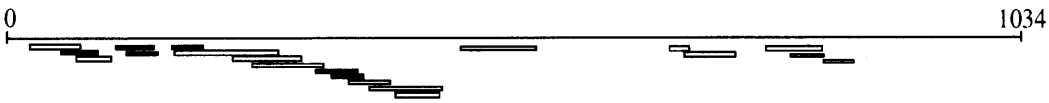



(b)No Redundancy nor Overlapping

Fig.5.5 Durations of Each Chromosome

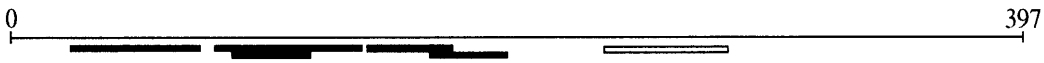



(a)The 300th Generation (The Length of the Chromosome: 847)




(b)The 1000th Generation (The Length of the Chromosome: 1034)


(c)The 2500th Generation (The Length of the Chromosome: 4415)




(d)The 5000th Generation (The Length of the Chromosome: 397)

Fig.5.6 Examples of Genes in a DNA Chromosome

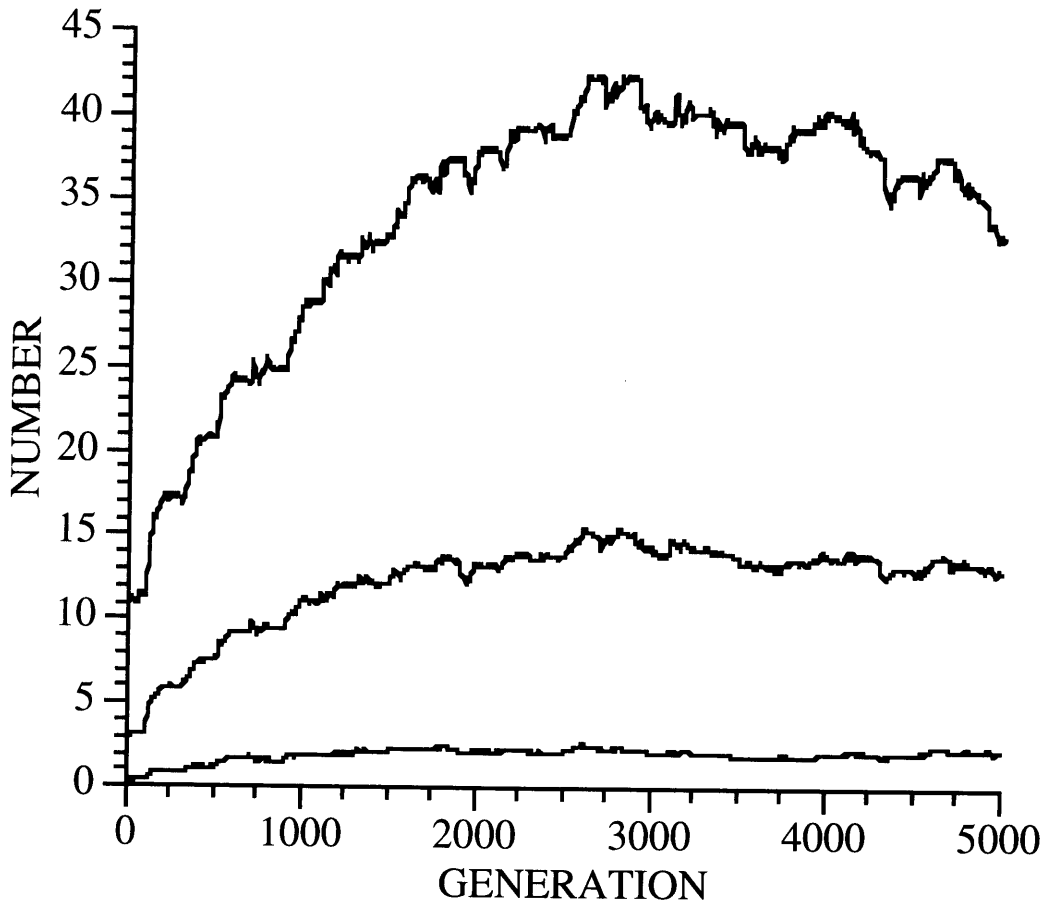



Fig.5.7 The Number of Genes in the Chromosomes (30 Trials)

## 5.4 Examples of the Robot Behavior

An example of evolution of control rules of chasing and avoiding robots is shown in Fig.5.8. Fig.5.8(a) shows the initial position and direction of the chasing robot and the avoiding robot. Fig.5.8(b), (c), (d), (e) and (f) shows an example of each movement at the 300th, 400th, 500th, 2300th, and 5000th generation, respectively. In Fig.5.8(b), the avoiding robot turned right at first and went straight. The chasing robot only went straight and reached the avoiding robot. In Fig.5.8(c), the avoiding robot went straight and the chasing robot crashed into the wall. In Fig.5.8(d), the chasing robot turned right at first, went straight, and reached the avoiding robot. In Fig.5.8(e), each movement was became a little complex. The chasing robot moved around in the upper left area, and the avoiding robot drew a rhombus. For the avoiding robot, the rhombic movement was able to get much food, but it was dangerous to be caught by the chasing robot. Fig.5.8(f) was the case where the avoiding robot acquired very effective rules. The avoiding robot was running along the walls. The chasing robot could measure only the distance using the ultra-sonic sensors. Therefore, the avoiding robot had, so to speak, protective coloring. The avoiding robot could avoid the chasing robot for 30 seconds. The followings are the examples of the acquired rules of the avoiding robot. These were the rules mainly used to run along the walls. The membership functions determined by the central position $x_c$ and the width $\sigma$ in the DNA chromosome are labeled for clarity.

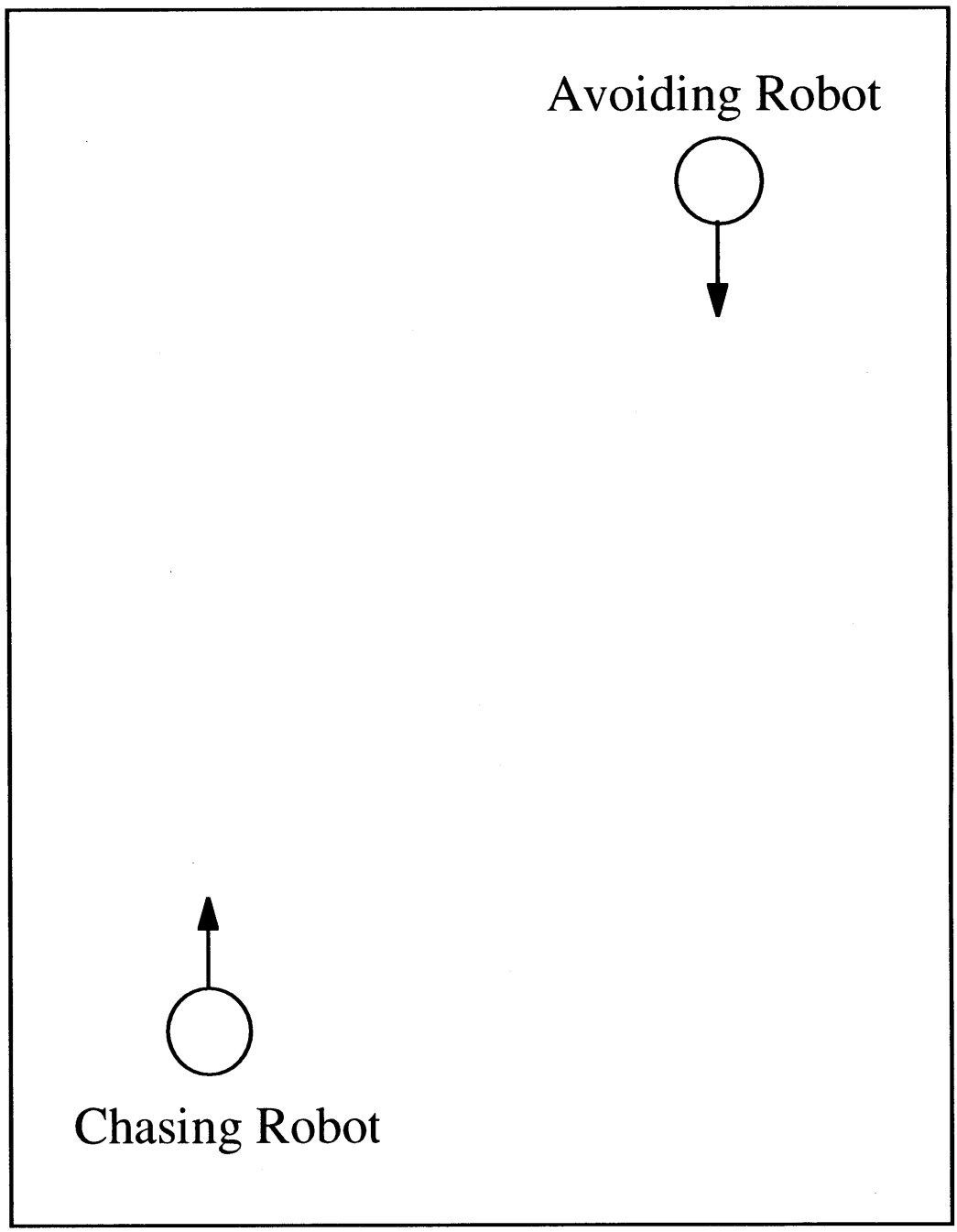
IF $D_0$ is Medium AND $D_2$ is Medium AND $D_1$ is Far AND $D_{11}$ is Medium
THEN $u$ is Very Small Left, $V$ is Medium Small

IF $D_1$ is Medium AND $D_0$ is Medium AND $D_{11}$ is Medium
THEN $u$ is ZERO, $V$ is Medium Big

IF $D_3$ is Near AND $D_0$ is Far THEN $u$ is Small Right

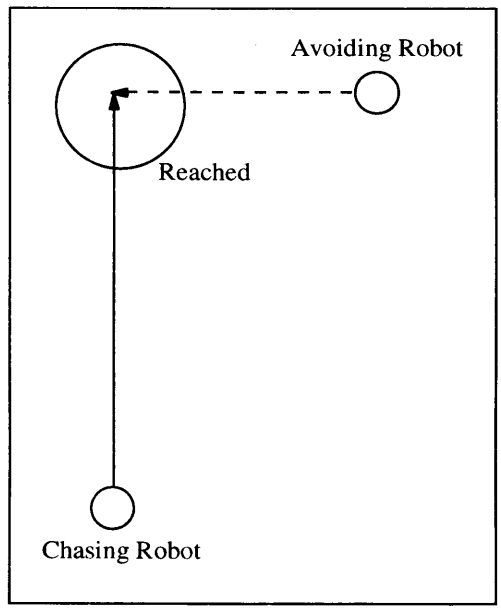
IF $D_0$ is Near THEN $u$ is Medium Right

These rules are used to go along the walls on the left-hand side. In this case, the avoiding robot acquired very effective rules. In other cases, the chasing robot acquired effective rules. For example, the chasing robots ran along the walls looking for the avoiding robots.

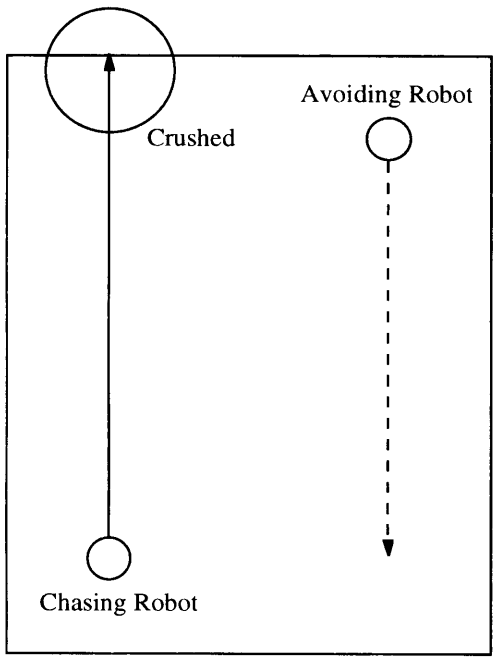


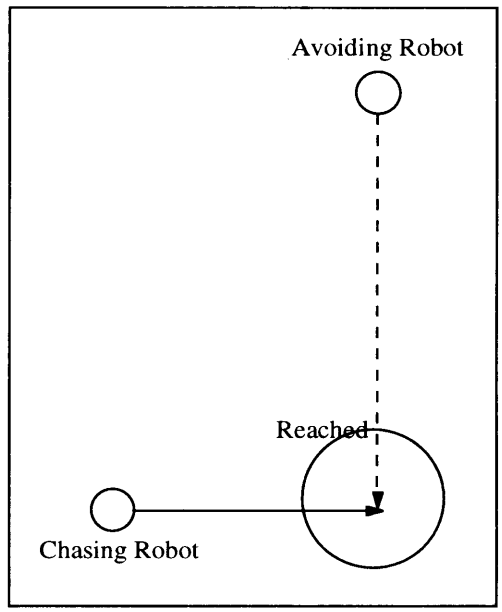

(a)The Initial Position and Direction

Fig.5.8 Movements of Each Robot



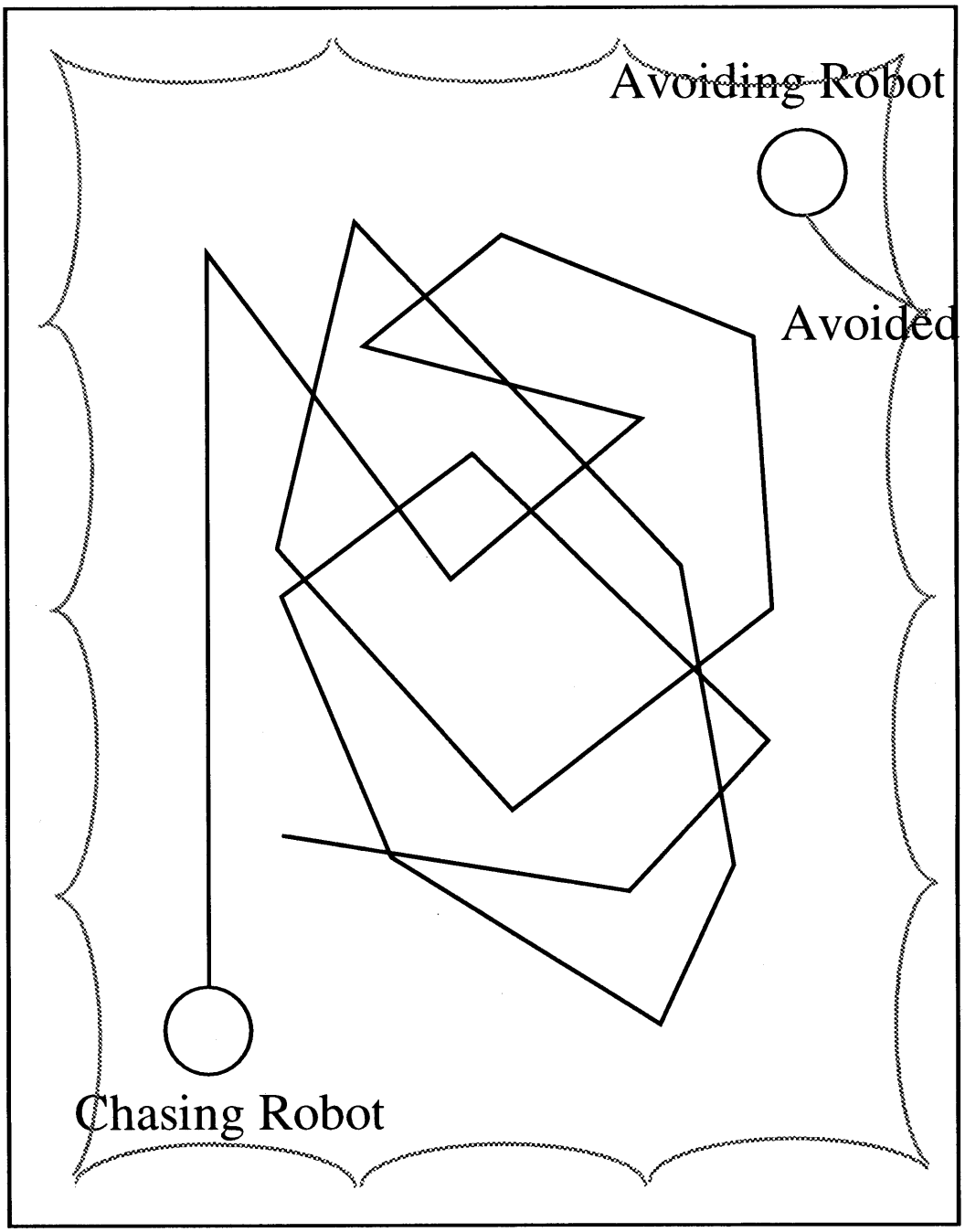

(b)The 300th Generation



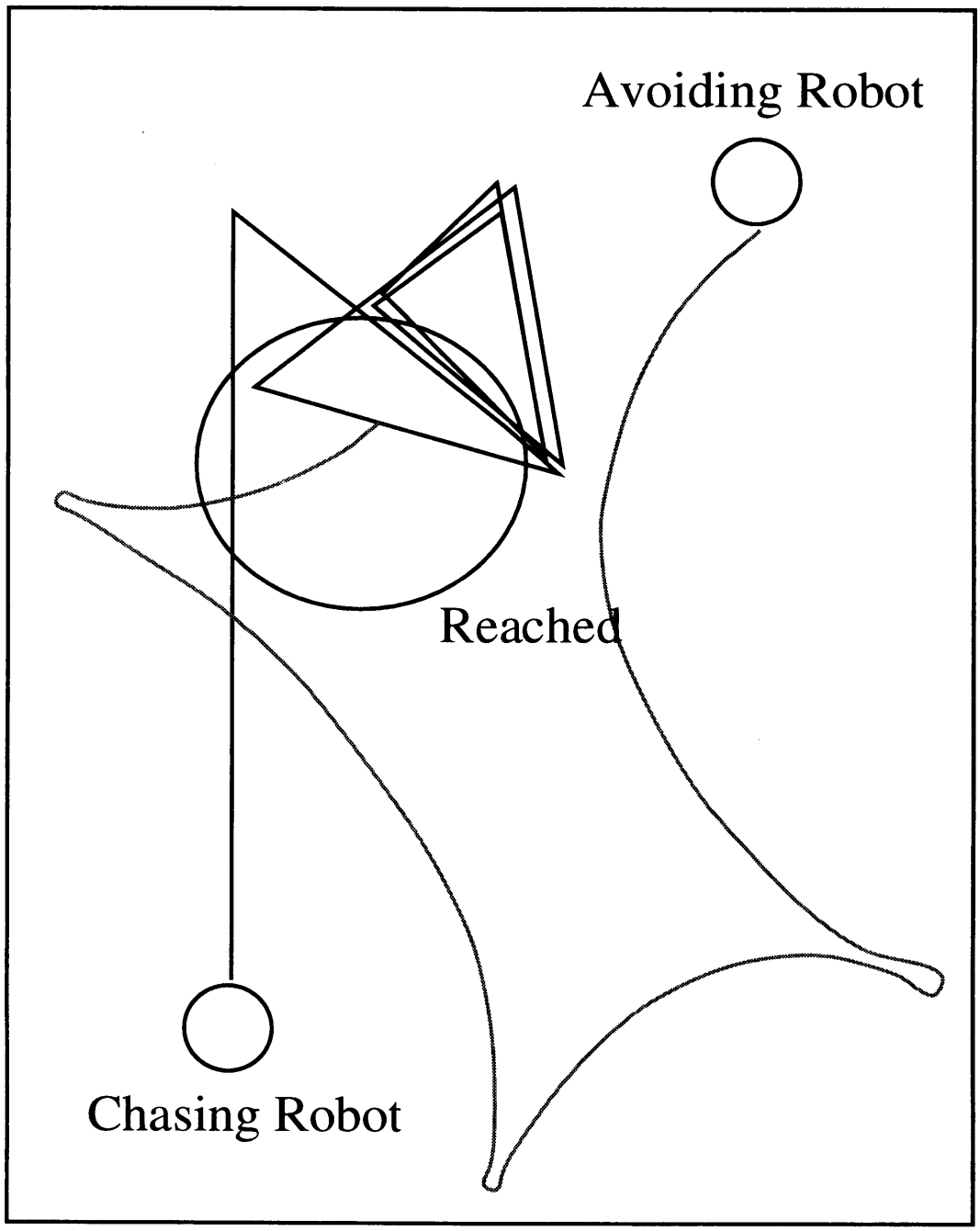

(c)The 400th Generation (d)The 500th Generation

Fig.5.8 Movements of Each Robot




(e)The 2300th Generation

Fig.5.8 Movements of Each Robot




(f)The 5000th Generation

Fig.5.8 Movements of Each Robot

## 5.5 Bacterial Operation

Lastly, the author examined the effectiveness of the bacterial operation. Fig.5.9(a) shows the averages of payoffs of 7 chasing robots until 6000th generation and Fig.5.9(b) shows those during the 3000-6000th generation. The solid line was the case (vii) where the chasing robots used the DNA coding method and the bacterial operation and the avoiding robots used only the DNA coding method. The dotted line was the case (viii) under the condition vice versa. The alternating period of the application of the genetic operations to the chromosomes of chasing and avoiding robots was 120. To make the evolutional opportunity equal, one generation of the bacterial operation was converted to $n \times m = 6$ generations in Fig.5.9. The bacterial operation also worked well. The parameters ($n$, $m$) were changed from (1, 6) to (2, 3), (1, 4), (2, 2), (1, 8), and (2, 4). The results showed the same effectiveness of the bacterial operation.

This thesis studied the effects of changes of genes by the bacterial operation. Fig.5.10(a)(b) and Fig.5.11(a)(b) show examples of the changes in genes and movements of robots at (a)the 500th and (b)the 2000th generations, respectively. In Fig.5.10(a), the bacterial operation was applied to GENE2 and three bases were changed by this operation. As a result, the amino acids were changed from Thr, Gly, and Ser to Lys, Asp, and Thr, respectively. The overlapping GENE1 on GENE2 was also changed by this operation. Two fuzzy rules were changed as follows:

IF $D_0$ is Medium AND $D_{10}$ is Very Near THEN $u$ is Medium Right, $V$ is Small

-->»IF $D_0$ is Medium THEN $V$ is Very Small

IF $D_2$ and $D_1$ is Medium AND $D_6$ is Near AND $D_{11}$ is Near THEN $u$ is Left

-->»IF $D_6$ is Near AND $D_{11}$ is Near THEN $u$ is Left

Before these changes, the chasing robot was turned right by the first rule and crashed into the wall at the point A in the Fig.5.11(a). The second rule did not fire before the change. By the change, the first rule had no steering command$u$ in the consequent and the second rule fired. As a result, the chasing robot turned left at the point A and avoided crashing into the wall. In GENE1, since the change of amino acid from Thr to Lys changed AND into THEN in the fuzzy rule, the length of the first rule became shorter.

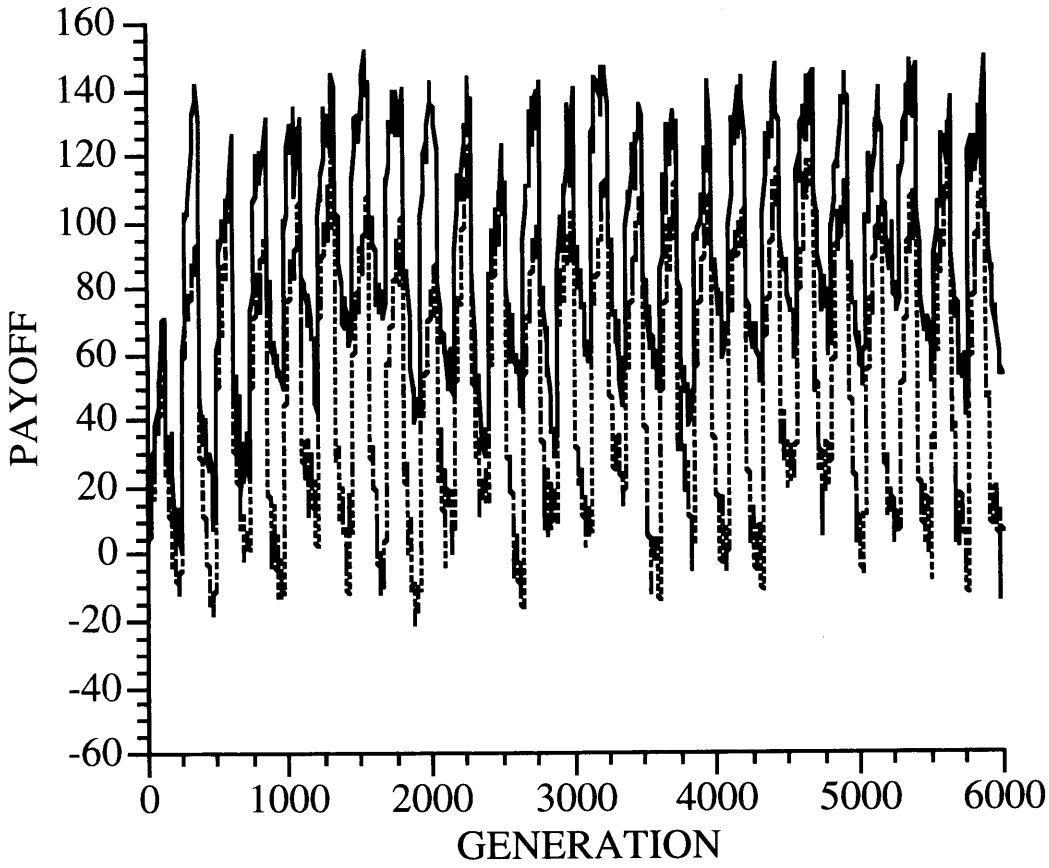
At the 2000th generation, GENE1 and the overlapping GENE2 on GENE1 were also changed as shown in Fig.5.10(b). The fuzzy rules affected by this operation were as follows:

IF $D_3$ is Very Near THEN $V$ is Small

IF $D_0$ is Medium THEN $u$ is Left

The changes were the width of membership function in the consequent part of the first rule and the position of that of the second rule. Though these modifications were a little, the movement of the chasing robot was drastically

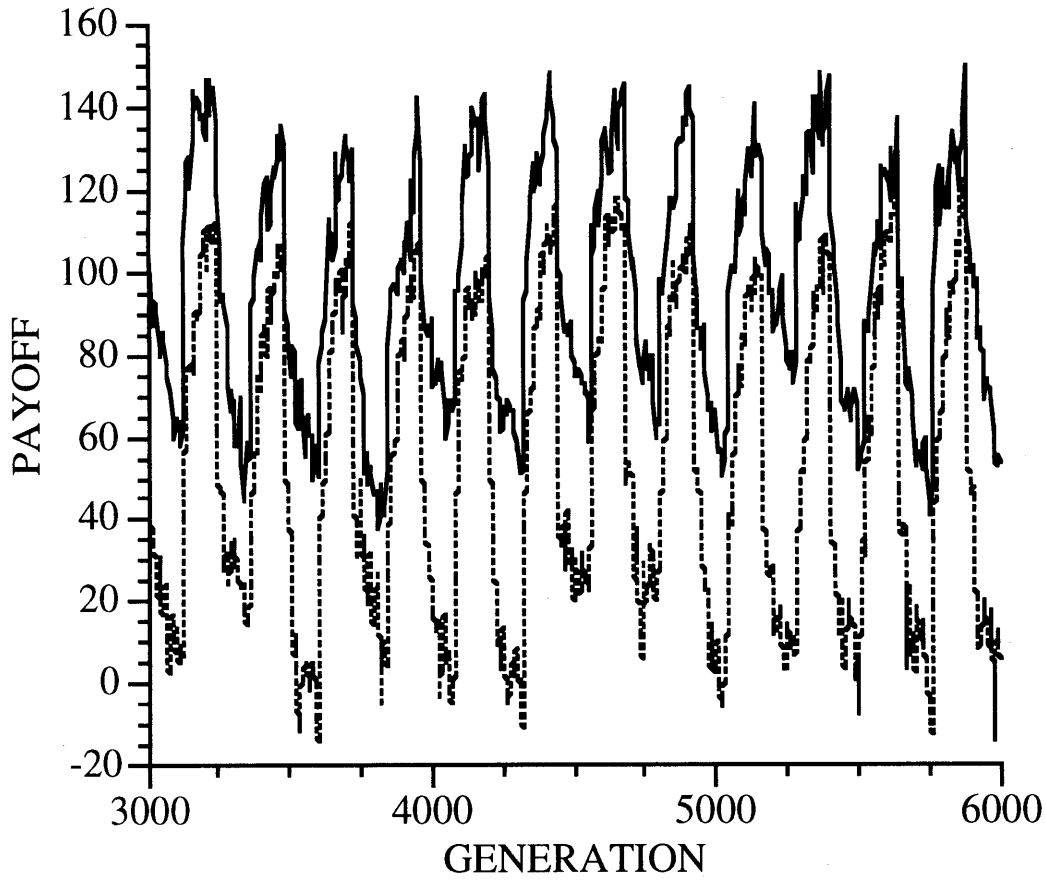
changed and it could reach the avoiding robot. The codon AGG was changed into AGA, but the amino acid corresponding to these codons was not changed. The change of amino acid from Asn to Lys did not change the parameters in that rule. The change from Gly to Arg just changed the membership functions a little.




(a)Until 5000th Generation

Fig.5.9 Payoffs of Chasing Robots (10 Trials)

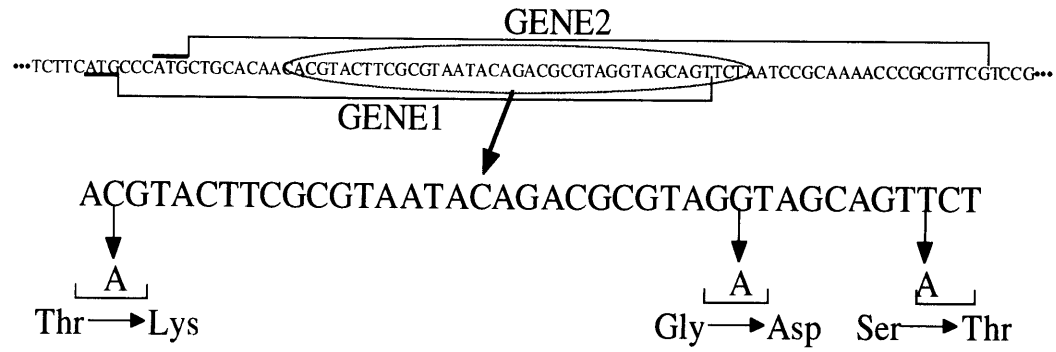
(Effects of the Bacterial Operation)



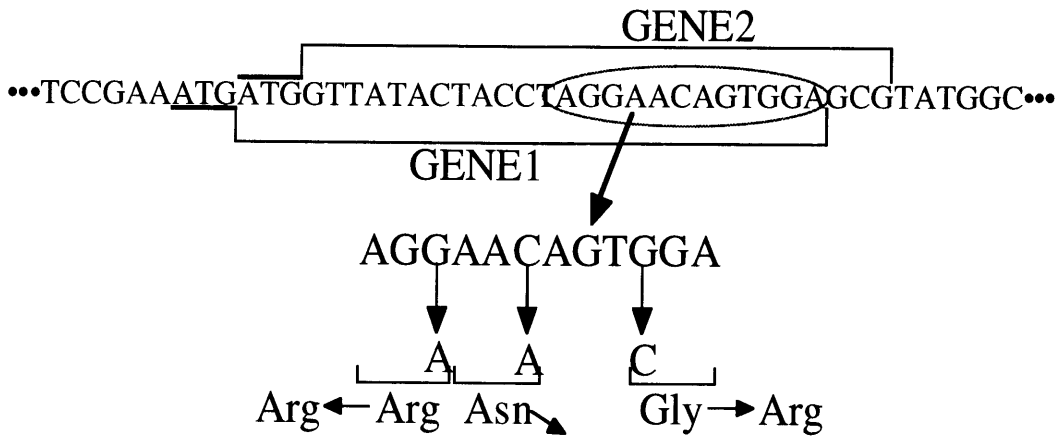

(b)During the 3000-5000th Generation

Fig.5.9 Payoffs of Chasing Robots (10 Trials)

(Effects of the Bacterial Operation)

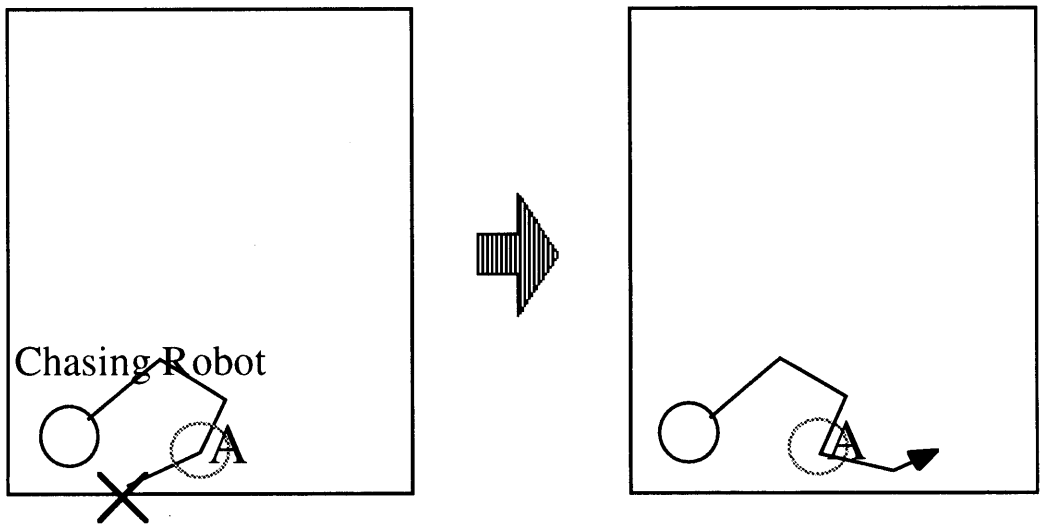



(a)The 500th Generation

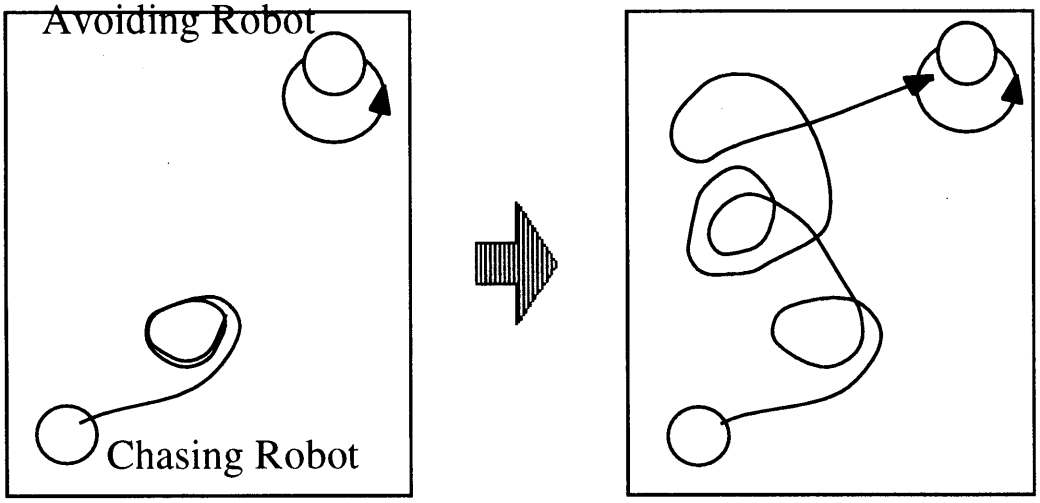



(b)The 2000th Generation

Fig.5.10 Examples of Changes in Genes by the Bacterial Operation




(a)The 500th Generation




(b)The 2000th Generation

Fig.5.11 Examples of Changes of Movements by the Bacterial Operation

## 5.6 Conclusions

Using the DNA coding method and its combined method with the PBGA, simulations were done. The effectiveness of the proposed DNA coding method and the mechanism of development from the artificial DNA was shown.

The effectiveness of this method was shown by comparing its performance with that of the conventional coding method of fuzzy rules. This comparison was done through competition of chasing and avoiding mobile robots. One robot had the proposed method and the other had the conventional method. The evaluation was the payoffs which the chasing or avoiding robots could obtain. The effectiveness of virus, enzyme, and bacterial operations was also examined under the same simulation conditions. The effects of changes of genes (fuzzy rules) by the bacterial operation were also studied in this chapter. The small change of a gene could cause a drastic improvement in the performance of robots. The effectiveness of redundancy and overlapping of genes realized by the proposed method was examined in details. The redundancy and overlapping of genes worked well so that genes survived far beyond the life time of individuals. The DNA coding method having the features of the redundancy and overlapping of genes is suitable for discovery of fuzzy control rules.

# Chapter 6

# Summary

## 6.1 Summary of the Thesis

A new coding method for GA based on the biological DNA, called the DNA coding method, and a mechanism of development from the artificial DNA have been proposed in this thesis. The DNA coding method and the mechanism of development from the artificial DNA are suitable for knowledge representation. One of the features of this method is that the length of the proposed DNA chromosome is variable and it is easy to insert and delete parts of chromosomes. By these features, various operations including virus and enzyme operations can be applied easily with no constraint. Another feature of the proposed coding method is that this method has redundancy and overlapping of genes, and this flexible coding works well so that genes survive far beyond the life time of individuals. This method was combined with the PBGA. This combination of the DNA coding method with the PBGA accelerates the knowledge discovery process. The summarized results obtained in this thesis are as follows:

In chapter 2, the flow of development from the DNA chromosome, the

genetic operations including virus and enzyme operations, and the features of the DNA coding method were described. The new coding method uses the four bases of DNA, and the way of development from DNA to a set of fuzzy rules is a simple analogy of development of biological DNA. The DNA chromosome has many redundant parts, and allows overlapped representation of genes. This DNA chromosome compresses information by the overlapping of genes. Length of the DNA chromosome is variable, and it has no constraint on genetic operations. The virus and enzyme operations also have no difficulty to apply because it is easy to insert and delete strands of strings from the DNA chromosomes.

In chapter 3, the DNA coding method was combined with the PBGA. The biological bacteria can transfer its own DNA from male cells to female cells through transfer of F factor. New bacteria whose parts of DNA are mutated when reproduction has occurred are tested in the environment, and the bacteria which can adapt themselves to the environment best can survive. By these process, the characteristics of more adaptable bacteria can be spread among the entire bacteria population. The PBGA utilizes mechanisms of genetic recombination in bacterial genetics. The PBGA is simple and very efficient in improving local portions of chromosomes. Genes are reproduced and tested, and the elite genes are transferred to the chromosomes. The PBGA can be combined with the DNA coding method easily and accelerates the knowledge discovery process.

Chapter 4 described the problem formulation for knowledge discovery, which was a discovery of effective fuzzy control rules using the DNA coding method. The concrete ways of genetic operations including virus, enzyme, and bacterial operations were also presented. These fuzzy rules are used to control mobile robots. The simulation conditions and the performance of two robots which

play chasing and avoiding were described in this chapter. The robot which performs well with a set of fuzzy rules receives more payoffs from the environment. Considering these payoffs as fitness values, the genetic operations are applied to the chromosomes, and the fuzzy rules are evolved. This chapter defined the flow of translation from the DNA chromosome and the correspondence between amino acids and the parameters. One gene which starts from a start codon corresponds to one fuzzy rule. The virus and enzyme operations are applied in the simple way by inserting or deleting a part of a chromosome. The bacterial operation is also applied to fuzzy rules. After this bacterial operation to all of the chromosomes, the conventional genetic operations are applied to the population of chromosomes by regarding the payoffs of the robots as their fitness values.

Chapter 5 showed the effectiveness of the proposed DNA coding method and the genetic operations described in the previous chapter. Simulations of competitions between the chasing robots and the avoiding robots were done. The effectiveness of this method and the virus, enzyme, and bacterial operations was shown. The performance of this method and these operations showed better result than that of the conventional method and those without these operations. The effects of changes of genes by the bacterial operation were studied in this chapter. The small change of a gene could cause a drastic improvement in the performance of robots. The effectiveness of redundancy and overlapping of genes realized by the proposed method was also shown in this chapter. The redundancy and overlapping of genes worked well so that genes survive far beyond the life time of individuals. The results showed that durations of chromosomes were shorter than long lived genes. The disappearance of chromosomes did not mean the disappearance of genes. Forthermore, many genes were activated again after long inactive time.

These genes were kept in chromosomes, even though they did not work.

## 6.2 Future Outlook of the Study

We, human beings, have DNA. In the real world, various creatures, including us, have evolved through very long history of change of biological DNA. DNA chromosome has amazing mechanisms of adaptability to varying environment. In this thesis the DNA coding method is applied to the discovery of fuzzy control rules. However, there are tremendous possibilities of making good use of its flexibility not only for optimization but also for discovery of knowledge representation, and so forth.

The artificial DNA chromosome in this paper consists of four bases. This number is not necessarily limited to the natural four bases. Analysis on the appropriate number of bases for the artificial DNA should be done.

The way of development from the DNA chromosome to fuzzy rules, and the correspondence between amino acids and the parameters of the fuzzy rules given in this thesis are the first trials of the author. In the future, this development mechanism to fuzzy rules should be reexamined and theoretical paths should be cultivated.

There are various gaps between the simulated robots and the real world robots. It is time consuming to make the robots evolve using the real mobile robots. Combination of simulations and experiments should be studied and some efficient ways to evolve the control rules in real time should be devised.

In future, a framework for evolvable hardware as well as software should

be established and it is the real goal of this research.

# References

## Chapter 1

[1]J.H.Holland, "*Adaptation in Natural and Artificial Systems*", University of Michigan Press (1975)

[2]L.Davis, "*Genetic Algorithms and simulated annealing*", Morgan Kaufmann Publishers (1987)

[3]D.E.Goldberg, "*Genetic Algorithm in Search*", Optimization and Machine Learning, Addison Wesley (1989)

[4]L.Davis(Editor), "*Handbook of Genetic Algorithm*", Van Nostrand Reynold (1989)

[5]D.E.Goldberg, "*Dynamic System Control Using Rule Learning and Genetic Algorithms*", Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI'85), pp.588-592 (1985)

[6]G.A.Cleveland, S.F.Smith, "*Using Genetic Algorithms to Schedule Flow Show Releases*", Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89), pp.160-169 (1989)

[7]D.Powell, M.Skolnick, S.Tong, "*Interdigitation: A Hybrid Technique for Engineering Design Optimization Employing Genetic Algorithms, Expert Systems, and Numerical Optimization*", in Handbook of Genetic Algorithms, ed. L.Davis, Van Nostrand Reinhold (1991)

[8]M.Yamamura, T.Ono, S.Kobayashi, "*Character-Preserving Genetic Algorithms for Traveling Salesman Problem*", Journal of Japanese Society for Artificial Intelligence, Vol.7, No.6, pp.1049-1059 (1992)

[9]H.Fang, P.Ross, D.Corne, "*A Promising Genetic Algorithm Approach to Job-Shop Scheduling, Rescheduling, and Open-Shop Scheduling Problems*", Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA'93), pp.375-382 (1993)

[10]A.Giordana, L.Saitta, F.Zini, "*Learning Disjunctive Concepts by Means of Genetic Algorithms*", Proceedings of the Eleventh International Conference on Machine Learning, pp.96-104 (1994)

[11]K.Morikawa, T.Furuhashi, Y.Uchikawa, "*Evolution of CIM System with Genetic Algorithm*", Proceedings of the First IEEE International Conference on Evolutionary Computation (ICEC'94), pp.746-749 (1994)

[12]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*A Study on Application of Genetic Algorithm to Automatic Placement of Parts on Printed Circuit Boards*", The Transactions of the Institute of Electrical Engineers of Japan (T.IEEE Japan), Vol.114-D, No.4, pp.387-392 (1994)

[13]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*Coding Methods for Automatic Placement of Parts on Printed Circuit Boards*", The Transactions of the Institute of Electrical Engineers of Japan (T.IEEE Japan), Vol.115-D, No.5, pp.642-651 (1995)

[14]X.Yang, T.Furuhashi, K.Oba, Y.Uchikawa, "*Selection of Features for Signature Verification Using the Genetic Algorithm*", Special Issue on Genetic Algorithms and Industrial Engineering, Computer & Industrial Engineering Journal (to be published)

[15]J.Renders, H.Bersini, "*Hybridizing Genetic Algorithms with Hill-Crimbing Methods for Global Optimization: Two Possible Ways*", Proceedings of the First IEEE International Conference on Evolutionary Computation (ICEC'94), Vol.1, pp.312-317 (1994)

[16]M.Malek, M.Guruswamy, et al., "*A Hybrid Algorithm Technique*", Technical Report of Texas University, at Austin, TR-89-06 (1990)

[17]D.Powell, S.Tong, M.Skolnick, "*EnGENEous: Domain Independent Machine Learning for Design Optimization*", Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89) (1989)

[18]H.Kitano, "*Empirical Studies on the Speed of Convergence of Neural Nework Training using Genetic Algorithms*", Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI'90) (1990)

[19]N.Kubota, K.Shimojima, T.Fukuda, "*The Role of Virus Infection in Virus-Evolutionary Genetic Algorithm*", Proceedings of the Third IEEE International Conference on Evolutionary Computation (ICEC'96), pp.182-187 (1996)

[20]K.Shimojima, N.Kubota, T.Fukuda, "*RBF Fuzzy Controller with Virus-Evolutionary Genetic Algorithm*", Proceedings of the 1996 IEEE International Conference on Neural Network (ICNN'96), pp.1040-1043 (1996)

[21]T.Arakawa, K.Shimojima, K.Shimojima, T.Fukuda, "*Virus-Evolutionary Genetic Algorithm for Manufacturing and Robot System*", Proceedings of the 1996 Symposium on Genetic Algorithms and Engineering Design, pp.129-138 (1996)

[22]N.Kubota, K.Shimojima, T.Fukuda, "*Virus-Evolutionary Genetic Algorithm - Coevolution of Planar Grid Model -*", Proceedings of the Fifth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'96), pp.232-238 (1996)

[23]N.Anderson, "*Evolutionary Significant of Virus Infection*", Nature, Vol.227, pp.1346-1347 (1970)

[24]T.Furuhashi, Y.Miyata, K.Nakaoka, Y.Uchikawa, "*A New Approach to Genetic Based Machine Learning and an Efficient Finding of Fuzzy Rules -- Proposal of Nagoya Approach--*", Lecture Notes in Artificial Intelligence, Vol.1011, pp.173-189 (1994)

[25]T.Hashiyama, T.Furuhashi, Y.Uchikawa, "*A Study on Fuzzy Rules for Semi-Active Suspension Controllers with Genetic Algorithm*", Proceedings of the Second IEEE International Conference on Evolutionary Computation (ICEC'95), pp.279-282 (1995)

[26]T.Furuhashi, Y.Miyata, Y.Uchikawa, "*Pseudo-Bacterial Genetic Algorithm and Finding of Fuzzy Rules*", the Second Online Workshop on Evolutionary Computation (WEC2), pp.65-68 (1996)

[27]L.Margulis, D.Sagan, Microcosmos, "*Four Billion Years of Microbial Evolutions*", Summit Boos (1986)

[28]J.D.Watson and others, "*Molecular Biology of the Gene*", The Benjamin/Cummings Publishing Company (1987)

[29]R.Schleif, "*Genetics and Molecular Biology*", The Johns Hopkins University Press (1993)

[30]B.Albers and others, "*Molecular Biology of the Cell*", Garland Publishing (1994)

[31]L.A.Zadeh, "*Fuzzy Sets*", Information and Control, Vol.8, pp.338-353 (1965)

[32]L.A.Zadeh, "*Fuzzy Logic and Approximate Reasoning*", Syntheses, Vol.30, pp.407-428 (1987)

[33]D.Mcneill, P.Freiberger, "*Fuzzy Logic*", Simon & Schuster (1992)

[34]T.Terano, K.Asai, M.Sugeno, "*Fuzzy Systems Theory and Its Applications*", Academic Press (1992)

[35]C.L.Karr, L.Freeman, D.Meredith, "*Improved Fuzzy Process Control of Spacecraft Autonomous Rendezvous Using a Genetic Algorithm*", SPIE Intelligent Control and Adaptive Systems, Vol.1196, pp.274-288 (1989)

[36]C.L.Karr, "*Design of an Adaptive Fuzzy Logic Controller Using a Genetic Algorithm*", Proceedings of the Fourth International Conference on Genetic Algorithms, pp.450-457 (1991)

[37]M.Valenzuela-Rendon, "*The Fuzzy Classifier System: A Classifier System for Continuously Varying Variables*", Proceedings of the Fourth International Conference on Genetic Algorithms, pp.346-353 (1991)

[38]J.H.Holland, J.S.Reitman, "*Cognitive Systems Based on Adaptive Algorithms*", in Pattern Directed Inference Systems, D.A.Waterman, F.HayesRoth (Editors), pp.313-329. Academic Press, New York (1978)

[39]T.Furuhashi, K.Nakaoka, K.Morikawa, Y.Uchikawa, "*Controlling Excessive Fuzziness in a Fuzzy Classifier System*", Proceedings of the Fifth International Conference on Genetic Algorithms, p635 (1993)

[40]T.Furuhashi, K.Nakaoka, K.Morikawa, Y.Uchikawa, "*An Acquisition of Control Knowledge Using Multiple Fuzzy Classifier Systems*", Journal of Japan Society for Fuzzy Theory and Systems, Vol.6, No.3, pp.603-609 (1994)

[41]K.Nakaoka, T.Furuhashi, Y.Uchikawa, "*A Study on Apportionment of Credits of Fuzzy Classifier Systems for Knowledge Acquisition of Large Scale Systems*", Proceedings of the Third International Conference on Fuzzy Systems, pp.1797-1800 (1994)

[42]M.A.Lee, H.Takagi, "*Dynamic Control of Genetic Algorithms Using Fuzzy Logic Techniques*", Proceedings of the Fifth International Conference on Genetic Algorithms, pp.76-83 (1993)

[43]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*A Fuzzy Modeling of Very Large Scale System Using Genetic Algorithm and a Multiple-Representing Method*", Proceedings of the Third IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'94), pp.1895-1898 (1994)

[44]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*A Study on Fuzzy Modeling of Very Large Scale System Using Genetic Algorithm*", Proceedings of the Third International Conference on Fuzzy Logic, Neural Nets and Soft Computing (IIZUKA'94), pp.623-624 (1994)

[45]M.A.Lee, R.Saloman, "*Hybrid Evolutionary Algorithms for Fuzzy System Design*", Proceedings of the Sixth International Fuzzy Systems Association (IFSA'95), Vol.1, pp.269-272 (1995)

[46]F.Hoffman, G.Pfister, "*A New Learning Method for the Design of Hierarchical Fuzzy Controllers Using Messy Genetic Algorithms*", Proceedings of the Sixth International Fuzzy Systems Association (IFSA'95), Vol. 1, pp.249-252 (1995)

# Chapter 2

[47]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*Acquisition of Fuzzy Rules from DNA Coding Method*", Lecture Notes in Artificial Intelligence 1152, pp.71-86 (1996)

[48]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*Emergence of Fuzzy Control Rules from DNA*", Proceedings of the 1995 IEEE/Nagoya University World Wisepersons Workshop (WWW'95) on Fuzzy Logic and Neural Networks/Evolutionary Computation, pp.49-55 (1995)

[49]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*Acquisition of Fuzzy Rules for Constructing Intelligent Systems using Genetic Algorithm based on DNA Coding Method*", Proceedings of International Joint Conference of CFSA/IFIS/ SOFT'95 on Fuzzy Theory and Applications, pp.447-448 (1995)

[50]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*An Emergence of Fuzzy Control Rules for Mobile Robots using DNA Coding Method*", Proceedings of the International Symposium on Artificial Life and Robotics (AROB) (1996)

[51]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*Emergence of Effective Fuzzy Rules for Controlling Mobile Robots using DNA Coding Method*", Proceedings of the Third IEEE International Conference on Evolutionary Computation (ICEC'96), pp.581-586 (1996)

[52]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*DNA Coding Method and a Mechanism of Development for Acquisition of Fuzzy Control Rules*", Proceedings of the Fifth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'96), pp.2194-2200 (1996)

[53]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*Effects of New Mechanism of Development from Artificial DNA and Discovery of Fuzzy Control Rules*", The Fourth International Conference on Fuzzy Logic, Neural Nets and Soft Computing (IIZUKA'96), pp 498-501 (1996)

[54]A.Kornberg, "*DNA Synthesis*", W.H.Freeman and Company (1974)

[55]M.Ridley, "*Evolution*", Blackwell Scientific Publications (1993)

[56]J.Ecker, M.Kupferschmid, "*Introduction to Operations Research*", New York: John Wiley & Sons (1988)

# Chapter 3

[57]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*A Combination of DNA Coding Method with Pseudo-Bacterial GA for Acquisition of Fuzzy Control Rules*", Proceedings of the First Online Workshop on Soft Computing (WSC1), pp.107-112 (1996)

[58]T.Yoshikawa, T.Furuhashi, Y.Uchikawa, "*Knowledge Acquisition of Fuzzy Control Rules using DNA Coding Method and Pseudo-Bacterial GA*", Proceedings of the First Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'96), pp.295-302 (1996)

# Acknowledgments

The author is heartily grateful to many people who contributed to completion of this dissertation in various ways.

He would like to thank Professor Yoshiki Uchikawa, Graduate School of Engineering, Nagoya University, for his reliable guidance, and Professor Shigeru Okuma, Graduate School of Engineering, Nagoya University, for valuable suggestions and comments. He is especially indebted to Associate Professor Takeshi Furuhashi, Graduate School of Engineering, Nagoya University, for his continuous advises and encouragement. He would like to acknowledge Assistant Professor Tetsuji Kodama and Assistant Professor Akio Ishiguro, Graduate School of Engineering, Nagoya University, for their help.

He thanks Dr. Koji Morikawa and Takashi Hasegawa, Graduate School of Engineering, Nagoya University, for their assistance on this study and fruitful discussions.

He thanks the members of the same group: Yujiro Miyata, Masatoshi Kobota, Yoichiro Hattori, Ichiro Takeuchi, Makoto Fujii, for his help and useful

discussions. He also thanks all members of Uchikawa laboratory for their good relationship.

Finally, He would like to thank his family for their help, support, and encouragement.