第 27 卷 第 4 期
2001 年 7 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol.27,No.4
July, 2001

Automatic Design of Takagi-Sugeno Fuzzy Controllers by a New DNA-Based Evolutionary Algorithm[1)]

DING Yong-Sheng REN Li-Hong SHAO Shi-Huang

(*Deparment of Automation, College of Information Sciences and Technology, Dong Hua University, Shanghai* 200051)

(E-mail: ysding@dhu.edu.cn, shshao@dhu.edu.cn)

Abstract In this paper, we propose a new approach to DNA-based evolutionary algorithm (DNA-EA) to design automatically a class of Takagi-Sugeno (TS) fuzzy controllers. The fuzzy controllers employ TS fuzzy rules with linear consequent, continuous input fuzzy sets, Zadeh fuzzy logic AND operation, and the widely-used centroid defuzzier. The fuzzy controllers are proved to be nonlinear PI controllers with variable gains. The fuzzy rules are automatically discovered, and the design parameters in the input fuzzy sets and the linear rule consequent are optimized simultaneously by the DNA-EA. The DNA-EA uses the DNA encoding method stemmed from the structure of the biological DNA to encode the design parameters of the fuzzy controllers. The gene transfer operation and bacterial mutation operation inspired by a microbial evolution phenomenon are introduced into the DNA-EA. Moreover, frameshift mutation operations based on the DNA genetic operations are also used in the DNA-EA. Our encoding method is suitable for complex knowledge representation, and is easy for the genetic operations at gene level to be introduced into the DNA-EA. The length of the chromosome is variable and it is easy to insert and delete parts of the chromosome. As a demonstration, we show how to implement the new method to design automatically a TS fuzzy controller in the control of a nonlinear system. The fuzzy controller can be automatically constructed by the DNA-EA. Computer simulation results indicate that the new method is effective and the designed fuzzy controller is satisfactory.

Key words DNA encoding method, gene transfer operation, frameshift mutation, evolutionary algorithms, TS fuzzy control.

1 Introduction

Fuzzy systems have successfully been used for a variety of applications, especially in control applications and modeling applications[1,2]. However, sometimes, expert knowledge cannot be easily described in linguistic languages. In these cases, the alternative methods are to acquire automatically the fuzzy rules and determine the parameters of a fuzzy system. Many approaches, such as, artificial neural networks (e.g., [3, 4]) and evolutionary algorithms (EAs) (including genetic algorithms (GAs))(e.g.,[5∼ 9]), have been proposed to develop fuzzy systems by automatically determining the design parameters in the input fuzzy sets and the rule consequent. For examples, Karr and Gentry applied a GA to tune the fuzzy sets of a fuzzy controller that controlled a pH control process and a cart-pole balancing system[5]. Shi et al. discussed how to evolve the shapes and types of the fuzzy sets and fuzzy rules using an EA[8].

Although the GA approach provides a way to possibly obtain global optimization solution, it has some limitations. In a GA, the search for an optimal solution is achieved through the manipulation of a population of string structures known as chromosomes. Each chromosome is a simple coding of a potential solution to the problem to be solved. With successive generations of population through reproduction and recombination operators, such as crossover and mutation,

[1)] Supported by National Natural Science Foundation of P.R.China(69874038, 60004006) & Shanghai Shuguang Project(99SG20).
Received 15 August, 2000; in revised form 21 November, 2000
收稿日期 2000-08-15 收修改稿日期 2000-11-21

the overall quality of the population assessed by the fitness function can be improved. In general, the bit string (0-1's) encoding is the most common method adopted by GA researchers because of its simplicity and tractability. However, when the size of the population grows up and the encoding of the chromosomes is very large, the expense of the evolutionary computation would become almost impractical. The rather long strings after encoding will increase the complexity of the problem. Also, GAs are not effective for searching the solution space locally due to crossover-based-search, and the diversity of the population sometimes decreases rapidly. Moreover, the GAs can neither represent the diverse genetic information by only using 0-1's encoding, nor can better imitate the regulation action of genes to the genetic processes. As such, some biological operations at the gene level cannot be effectively adopted in the existing GAs. For the applications of GAs, a well-chosen chromosome format can enhance the understanding of the problem formulation and is also flexible for practical implementation.

Deoxyribonucleic acid (DNA) is the major genetic material for life, and encodes plentiful genetic information. A natural question we wonder is how to utilize the genetic mechanisms of the biological DNA to develop a new DNA-based computation model for optimization problems? Recently, extensive studies have been conducted to explore the possibility of using DNA as computing hardware (e.g.,[10∼ 13]). Moreover, research has been directed to the soft computing aspect of the DNA computing, that is, the integration of DNA computing with intelligent technologies[14], e.g., evolutionary computation[15∼17] and neural networks[18,19]. Among them, the integration of DNA computing with evolutionary computation is a major topic and the current interests are on two aspects: to explore the relationship between evolutionary computation and DNA-based computation[14], and to apply GAs to search for good DNA encoding methods[15]. View from the mechanism of evolutionarily inspired approaches, it can be seen that they seem suitable to be implemented by DNA. As such, in order to overcome the limitations of the GAs, a few of GAs based on the mechanism of the biological DNA, such as double stranded DNA[17] and DNA encoding method [16,20,21], have been developed. Our group used a DNA encoding method and developed a DNA-based GA (DNA-GA) to find the fuzzy control rule sets in a Mamdani fuzzy system[20]. Yoshikawa et al. combined the DNA encoding method with the pseudo-bacterial genetic algorithm (PBGA)[22], and developed a DNA PBGA. Besides the algorithms based on DNA models, Chen et al. proposed the laboratory implementation of the DNA-GA for some simple problems, such as the MAX 1s, the royal road and the cold war problems[23].

The genetic information is a great source from which to develop variations on the basic GA scheme. During the microbial evolutionary process, bacterial can transfer DNA to recipient cells through bacterial recombination at the bacterial genetic level. Genes can be transferred from a single bacterium to others[24]. Inspired by this phenomenon, we could develop a gene transfer operation and a bacterial mutation operation, which directly transfer a gene strand from a cell to other cells.

In this paper, we investigate how to design automatically a class of Takagi-Sugeno (TS) fuzzy controllers by employing a new DNA-based evolutionary algorithm (DNA-EA). In Section 2, the configuration of the fuzzy controllers is defined. In Section 3, we show how to employ the DNA-EA to discover the effective fuzzy rules of the TS fuzzy controllers. We first give an introduction of the biological background about DNA-EA, and present an artificial DNA model based on the biological structure of the DNA. Then, we develop a new DNA-EA, which incorporates features of DNA and bacterial evolution into the EA. Some important procedures of the DNA-EA to design the TS fuzzy controllers, such as DNA encoding method and DNA-based genetic operators, are discussed. Moreover, we show how to employ the DNA-EA to design automatically the TS fuzzy controllers by adding or deleting the number of fuzzy rules and tuning the design parameters in the input fuzzy sets and the rule consequent. In Section 4, we provide an example to demonstrate the efficiency and effectiveness of DNA-EA in the design of the TS fuzzy controllers. In Section 5, we discuss some advantages of the DNA-EA.

2 Configuration of a Class of Takagi-Sugeno Fuzzy Controllers

The TS fuzzy controllers studied in this paper employ two input variables and one output variable. The input variables are error and rate change of error (rate, for short) of system output with respect to output setpoint/reference. They are denoted as follows:

$$e(nT) = SP(nT) - y(nT),$$
$$r(nT) = (e(nT) - e(nT - T))/T,$$

where n is a positive integer, T is sampling period, $SP(nT)$ is the setpoint/reference, and $y(nT)$ is system output. The two input variables are both fuzzified by N input fuzzy sets. The generalized membership function is used for a fuzzy set, and is designated as $\mu_i(e)$ (or $\mu_i(r)$), where $i = 1, \cdots, N$. The mathematical representation for the membership functions is:

$$\mu_i(x) = GM(\alpha_i, \beta_i, \gamma_i) = e^{-|\alpha_i x + \beta_i|^{\gamma_i}}, x = e \text{ or } r, \tag{1}$$

where α_i, β_i, and γ_i are three design parameters in the membership functions. We call (1) the generalized membership functions, because with different values of the parameters α_i, β_i, and γ_i, (1) can approximate various membership functions, such as widely-used triangular, trapezoidal and gaussian membership functions[20]. That is to say, the definitions of the input fuzzy sets are very general and contain almost all of the fuzzy sets employed in fuzzy systems[1,2].

NTS fuzzy control rules with linear consequent are used. The form of the TS fuzzy rules is as follows:

$$\begin{array}{ll} \text{IF} & e(n) = GM(\alpha_{ei}, \beta_{ei}, r_{ei}), \\ \text{AND} & r(n) = GM(\alpha_{ri}, \beta_{ri}, r_{ri}), \\ \text{THEN} & \Delta u(nT) = p_i e(nT) + q_i r(nT), \end{array} \tag{2}$$

where $\Delta u(nT)$ is the incremental output contribution of this rule to the fuzzy controller output, and p_i and $q_i (i = 1, \cdots, N)$ are design parameters in the rule consequent. For N fuzzy rules, there are $2N$ design parameters in the rule consequent. The parameter values are chosen by the fuzzy system developer. Zadeh fuzzy logic AND operation is employed to evaluate the ANDs in the fuzzy rules, and the combined membership for the rule consequent is

$$\mu_i(\Delta u) = \min(\mu_i(e), \mu_i(r)).$$

The centroid defuzzifer is employed for defuzzification[5]:

$$\Delta u(nT) = \frac{\sum_{i=1}^{N} \mu_i(\Delta u) \cdot (p_i e(nT) + q_i r(nT))}{\sum_{i=1}^{N} \mu_i(\Delta u)}. \tag{3}$$

The new output of the fuzzy controller at $nT + T$ is

$$u(nT + T) = u(nT) + \Delta u(nT).$$

From (3), we have:

$$\Delta u(nT) = K_I(e, r)e(nT) + K_P(e, r)r(nT), \tag{4}$$

where

$$K_P(e, r) = \frac{\sum_{i=1}^{N} \mu_i(\Delta u) \cdot q_i}{\sum_{i=1}^{N} \mu_i(\Delta u)}, \qquad K_I(e, r) = \frac{\sum_{i=1}^{N} \mu_i(\Delta u) \cdot p_i}{\sum_{i=1}^{N} \mu_i(\Delta u)}.$$

Recall that the linear PI controller in incremental form is:

$$\Delta u(nT) = \bar{K}_i e(nT) + \bar{K}_p r(nT), \tag{5}$$

where $\bar{K}_p$ and $\bar{K}_i$ are proportional-gain and integral-gain, respectively. Comparing (4) with (5), one sees that the fuzzy controllers are actually nonlinear PI controllers with variable proportional-gain,$K_P(e,r)$, and variable integral-gain, $K_I(e,r)$, which are determined by $e(nT)$, $r(nT)$ and the design parameters, α_i, β_i, and γ_i in the input fuzzy sets, and p_i and q_i, in the rule consequent.

The above gain relationship between the TS fuzzy controllers and the linear PI controller can be used to achieve reasonable initial ranges of the design parameters for the fuzzy controllers, as will be shown in detail later.

3. Automatic Design of TS Fuzzy Controllers via the DNA-EA

In this section, we develop a new approach to the DNA-EA and their use for the automatic design of the TS fuzzy controllers defined in Section II. The DNA-EA uses a new DNA encoding method inspired from the biological DNA. We also discuss how to use the DNA encoding method replacing the bit string encoding method in the GAs to represent the fuzzy rules of the TS fuzzy controller. In the DNA-EA, we employ the gene transfer operation replacing the crossover operation, and the bacterial mutation and the frameshift mutation operations replacing the point mutation operation in the GAs. By using the DNA-EA, the effective fuzzy rule sets of the TS fuzzy controllers can be discovered and the design parameters in the input fuzzy sets and the rule consequent are optimized simultaneously.

A) New DNA Encoding Method to Fuzzy Rules

As we known, the basic elements of biological DNA are nucleotides. Due to their different chemical structure, nucleotides can be classified as four bases: Adenine (A), Guanine (G), Cytosine (C) and Thymine (T). A triplet code of nucleotide bases specifies the codon, which in turn contains a specific anticodon on transfer RNA (tRNA) and assists subsequent transmission of genetic information in the formation of a specific amino acid. A chromosome consists of combinations of the above four bases and can represent different genes. Although there are 64 possible triplet codes, only 20 amino acids are interpreted by codons.The corresponding relationship between codons and amino acids is shown in Table 1. It should be noticed that same amino acid may be encoded by different codons in the DNA[24].

Table 1 Translation from the codons in the DNA chromosome into the amino acids, then into the parameter valuse of the TS fuzzy controllers

First Base	Second Base				Third Base
	T	C	A	G	
T	Phe (1)	Ser(3)	Tyr(4)	Cys(5)	T
	Phe (1)	Ser (3)	Tyr (4)	Cys(5)	C
	Leu (2)	Ser (3)	Stop(0)	Stop(0)	A
	Leu (2)	Ser(3)	Stop (0)	Trp (6)	G
C	Leu (2)	Pro (7)	His (8)	Arg(10)	T
	Leu (2)	Pro (7)	His (8)	Arg(10)	C
	Leu (2)	Pro (7)	Gln (9)	Arg(10)	A
	Leu (2)	Pro (7)	Gln (9)	Arg(10)	G
A	Ile (11)	Thr (13)	Asn (14)	Ser (3)	T
	Ile (11)	Thr (13)	Asn (14)	Ser (3)	C
	Met (12)	Thr (13)	Lys (15)	Arg (10)	A
	Met (12)	Thr (13)	Lys (15)	Arg (10)	G
G	Val (16)	Ala (17)	Asp (18)	Gly (20)	T
	Val (16)	Ala (17)	Asp (18)	Gly (20)	C
	Val (16)	Ala (17)	Glu (19)	Gly (20)	A
	Val (16)	Ala (17)	Glu (19)	Gly (20)	G

From the biological DNA structures, we can exploit an artificial DNA computation model for some practical problems. A single strand of DNA can be likened to a string consisting of a combination of four different symbols, A, G, C, T. Mathematically, this means we have a four-letter alphabet $\sum\{A, G, C, T\}$ to encode information, which is more than enough, considering that an electronic computer needs only two digits, 0 and 1, for the same purpose. In the artificial DNA model, the design parameters of a problem to be

solved are encoded by four-letter alphabet $\sum\{A, G, C, T\}$ to form a chromosome.Based on the DNA model, we can introduce features of the biological DNA into the EA and develop the new DNA-EA.

In our new DNA encoding method for the TS fuzzy controllers, the chromosome with $\sum\{A, G, C, T\}$ codes determines the parameter values in the input fuzzy sets and the rule consequent. Specifically, a triplet code of nucleotide bases specifies the codon, and transmits genetic information in the formation of a specific amino acid. Each amino acid is interpreted as the values of design parameters in the input fuzzy sets and the rule consequent. The corresponding relationship between a DNA chromosome and the TS fuzzy rule sets is described in Fig. 1.

In the figure, the DNA chromosome is made of n TS fuzzy rules, and its length is variable according to the number of the fuzzy rules. A part of DNA codes (totally 24 bases) is corresponding to a fuzzy rule, and can be translated into the design parameters in the input fuzzy sets and the rule consequent. The meaning of each amino acid is determined by its position in the sequence of amino acids corresponding to a chromosome. In Fig.1, we also provide the translation process from a part of DNA codes to the design parameters in the rule # k. Based on Fig.1 and Table 1, reading from the top of a chromosome, it can be translated into the design parameters to form a fuzzy rule set.

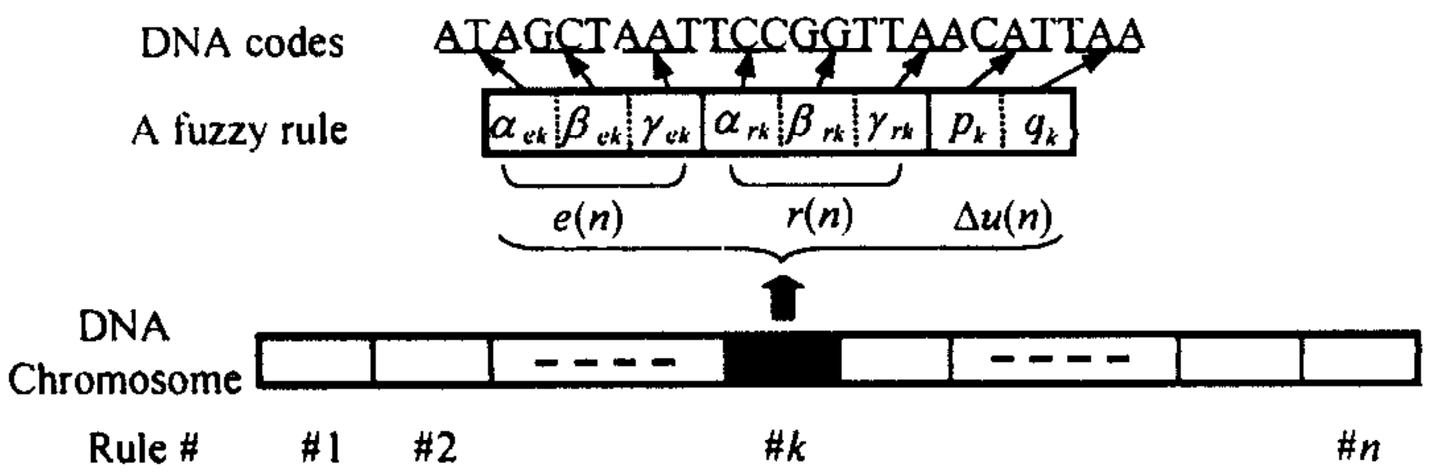

Fig.1 The DNA decoding method of a DNA chromosome corresponding to a group of fuzzy rule sets

It should be noted that the DNA encoding method to the TS fuzzy rule sets supplies a high degree of freedom for the DNA-EA, which can simultaneously define the variables to be used in the rules, the rules themselves, the parameters in the input fuzzy sets and the rule consequent, and the number of the rules in the rule base. However, the encoding method has some drawbacks due to the lack of uniformity of membership functions. Every rule has a different set of membership functions, and consequently, there is no bonding between the membership functions for a variable. The purpose we use such DNA encoding method is that the genetic operations in the DNA-EA can be implemented on the DNA chromosome. And the adaptive addition or deletion of a fuzzy rule can be easily implemented.

Based on the DNA encoding method, we following develop the genetic operators in the DNA-EA.

B) Gene Transfer Operation

The gene transfer operation is done as follows:

1) Sort the population and divide it in two halves. The half with higher fitness is called the superior half, and the other half is called the inferior half;
2) Choose randomly one DNA chromosome from the superior half, named source DNA chromosome, and another DNA chromosome from the inferior half, named destination DNA chromosome;
3) According to a given criterion, choose a good part from the source DNA chromosome and transfer it to the destination DNA chromosome. A good part can be a fuzzy rule or a group of rules with a high degree of activation value;
4) Repeat 1), 2), and 3) for M times in one generation, where M is the number of infections per generation.

The process for the gene transfer operation is shown in Fig.2. The gene transfer operation is expected to rapidly spread the good parts (corresponding to the good fuzzy rules)

of the DNA chromosome with superior fitness to the DNA chromosome with inferior fitness. By doing so, the overall search process should be proceeded more efficient, since the operators actuate more frequently on better rules, leading to the rapid construction of fuzzy systems that fulfill the requirements.

C) Mutation Operation

In DNA-EA, two mutation methods are employed. One is the bacterial mutation, and the other is frameshift mutation.

1) Bacterial mutation

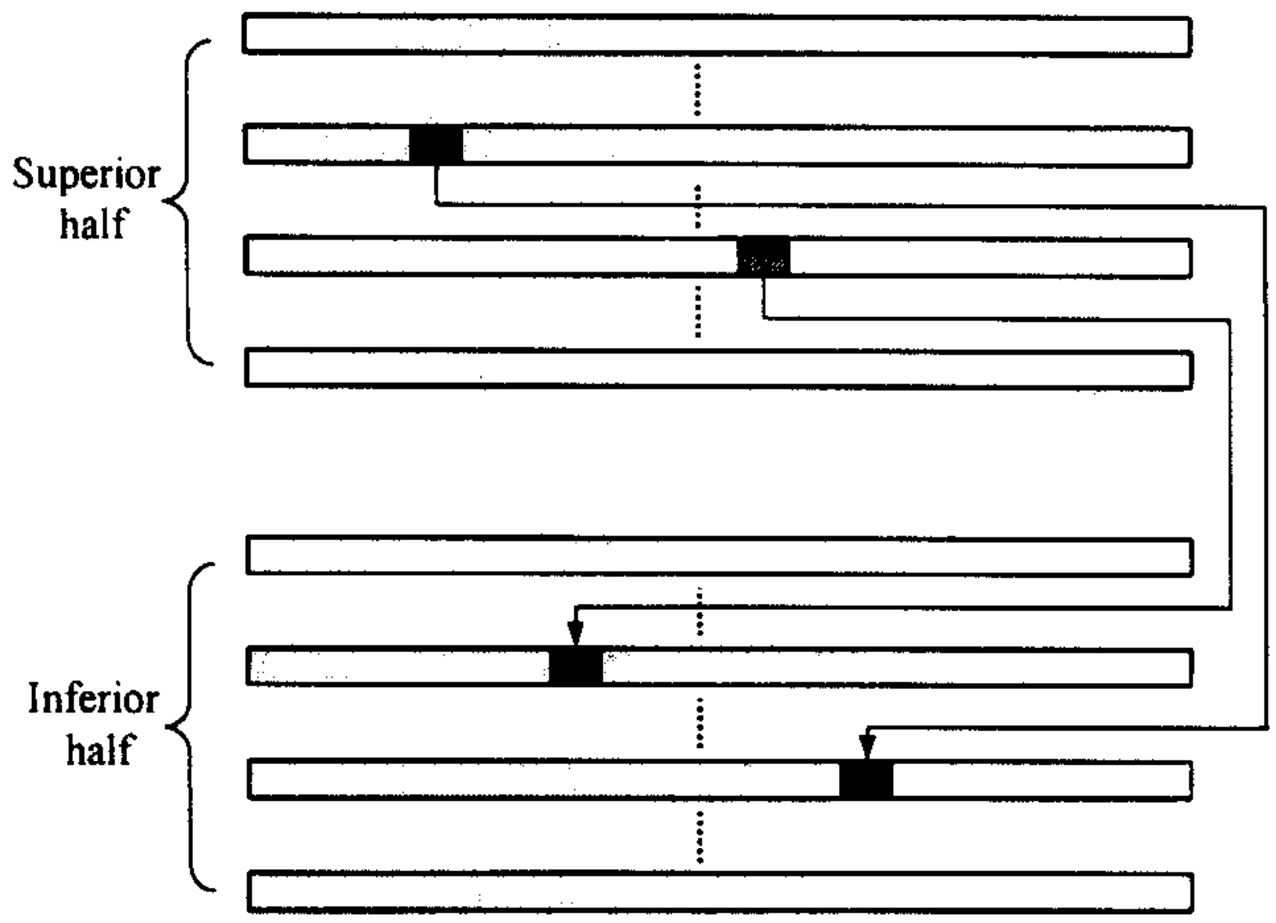

Fig.2 The gene transfer operation

Suppose there are p parts in a DNA chromosome. A part is corresponding to a fuzzy rule. The best DNA chromosome is chosen from the m DNA chromosomes. The ith part of the selected DNA chromosome is randomly chosen and transferred to corresponding part of the rest $m-1$ DNA chromosome. The bacterial mutation operation is always applied to all the m chromosomes in the population.

2) Frameshift mutation

In the biological DNA chromosome, there are two frameshift mutations. One is deletion mutation, in which one or more base-pairs are lost. Deletion mutation is due to enzyme operation. The other is insertion mutation, in which one or more base-pairs are inserted into the sequence. Insertion mutation is due to virus operation. Accordingly, we have developed two frameshift mutations of ours: deletion and insertion. The two frameshift mutations are selectable according to the evolution process of the DNA-EA. Fig.3 shows an example of deletion operation where the bases between the start codon TAG and end codon TAT are lost, as such a fuzzy rule is deleted. Fig.4 shows an example of insertion operation where a base sequence moves into the chromosome, and a fuzzy rule is added into the fuzzy system. That is to say, the deletion and the insert mutation operations could be used to add or delete some bases in the DNA chromosome. The operated bases can be one or several fuzzy rules. If the contributions of two fuzzy rules to the fuzzy controller are near, one fuzzy rule can be deleted by using the deletion operation. If the control performance of the fuzzy system cannot be improved any more by using the current fuzzy rule sets, we should consider to add a (or several) fuzzy rule(s) by using the insertion operation. As such, the fuzzy rules could be automatically added or deleted to obtain the more proper fuzzy rule sets.

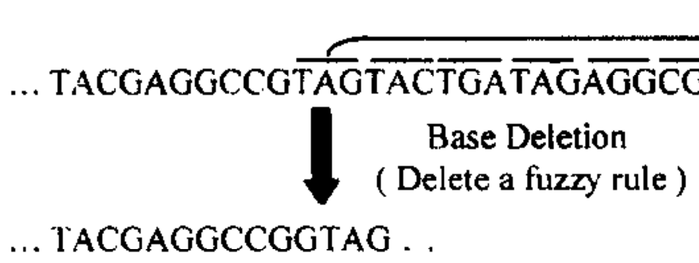

Fig.3 An example of frameshift mutation: Deletion operation.

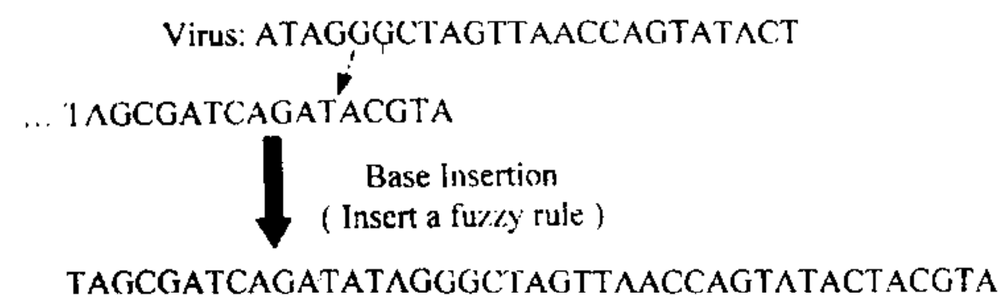

Fig.4 Another example of frameshift mutation: Insertion operation

D) Performance Evaluation

Using the new DNA encoding method, we can translate the codons into the amino acids based on the corresponding relationship between the codons and the amino acids shown in Table 1. Then, the amino acids are translated into the design parameters of the TS fuzzy controllers. The translation process in Table 1 imitates the translation process from DNA to protein. Also, it is a basic framework for translating the codons into the amino acids, and then into the design

parameters. The range of the design parameters can be adjusted with respective to [0, 20] according to different design problems. For a particular application, one may transfer a value in the range of [0, 20], into the proper range of the design parameters. After the translation, the fuzzy controller with these design parameters can be used and the fitness function can be computed. The choice of the fitness function or the performance index is dependent on the types of responses that are desired for the particular system.Since the central objective of fuzzy control is to minimize the error between the actual system response and the setpoint/reference, the fitness function we adopt in this paper is chosen as follows:

$$f_{fit} = C - \sum_{k=1}^{n}(e^2(k) + r^2(k)),$$

where C is a constant.

When we use the DNA-EA to design automatically the TS fuzzy controllers, we first employ a small number of (e.g., two) input fuzzy sets to fuzzify $e(nT)$ and $r(nT)$, initially. Then, we adopt the gene trasfer operation and bacterial mutation operation in the DNA-EA to evolve the fuzzy systems. After evolution of a certain generations, e.g. 10 generations, if the contributions of two fuzzy rules to the fuzzy controller are near, one fuzzy rule is deleted by using the deletion operation. If the control performance of the fuzzy system cannot be improved any more by using the current fuzzy rule sets, we consider to add a (or several) fuzzy rule(s) by using the insertion operation. As such, the fuzzy rules could be automatically added or deleted to obtain the more proper fuzzy rule sets. After the convergence of DNA-EA, the structure of the fuzzy controller can be built, and the design parameters in the input fuzzy sets and rule consequent can be obtained.

4 Simulation Study

Due to their nonlinear properties, fuzzy controllers are known to be capable of regulating nonlinear systems. However, it is rather difficult to construct the fuzzy rules and tune the design parameters in the input fuzzy sets and the rule consequent, because there are many design parameters.

In order to examine the effectiveness of the DNA-EA, we now employ the DNA-EA to design automatically the TS fuzzy controller and use it to control a nonlinear system. The DNA-EA is used to build the fuzzy controller and to optimize the design parameters in the input fuzzy sets and the rule consequent simultaneously. The nonlinear systems is

$$y(k) = 0.8y(k-1) - 0.6y(k-2) + 0.4u(k-1) + 0.12u^2(k-1) + 0.2u(k-2) + 0.06u^2(k-2). \quad (6)$$

Initially, the TS fuzzy controller uses two generalized membership function-type input fuzzy sets to fuzzify $e(nT)$ and $r(nT)$, respectively, which means the TS fuzzy controller only employs two fuzzy rules at the beginning. The number of population, m, is 50. The number of infections per generation, M, is 20.

Before computer simulations, we find a proper initial range for each design parameter in the rule consequent according to the gain relationships between the TS fuzzy controller and linear PI controller derived in Section 2. First, we design a PI controller by using the trial-and-error tuning method. The good gains of the PI controller are: $\bar{K}_p = 0.1$, and $\bar{K}_i = 0.1$. Then, according to the gain relationships between the TS fuzzy controller and the linear PI controller, we can achieve reasonable initial value ranges of the design parameters for the TS fuzzy controller. We roughly choose the proper ranges of the design parameters in the rule consequent based on the values of $\bar{K}_p$ and $\bar{K}_i$, where $[p_i^{\min}, p_i^{\max}] = [0, 0.2]$ and $[q_i^{\min}, q_i^{\max}] = [0, 0.2]$. Furthermore, the transforming relationship between the parameter values [0, 20] and $[p_i^{\min}, p_i^{\max}]$ (or $[q_i^{\min}, q_i^{\max}]$) can be established.

During the computer simulation, the gene transfer operation and the bacterial mutation operation are employed in the DNA-EA at each generation. At the mean time, after the

evolution of a certain generations, e.g. 10 generations, when the satisfactory simulation results cannot be achieved by using the current fuzzy rule sets, we consider to add or delete one or more fuzzy rules by using the insertion or the deletion operation. Computer simulations show that after convergence of the DNA-EA, we can always find a group of parameter values for the input fuzzy sets and the rule consequent that obtain satisfactory control performance. One typical example of the control performance of the TS fuzzy controller designed by the DNA-EA is shown in Fig.5. The final number of fuzzy rules is five, and the values of the design parameters of the TS fuzzy controller are shown in Table 2.

Table 2　The values of the design parameters of the TS fuzzy controller designed by the DNA-EA in the control of the nonlinear system (6)

#	$\mu_i(e)$			$\mu_i(r)$			Rule consequent	
	α_{ei}	β_{ei}	γ_{ei}	α_{ei}	β_{ei}	γ_{ei}	p_i	q_i
1	0	0	6.00	2.50	-1.20	2.44	0.18	0
2	-4.0	0.30	8.22	-3.00	1.05	3.33	0.008	0.14
3	-0.5	0.75	5.56	-3.00	-0.75	3.33	0.10	0
4	-0.5	0	7.33	4.00	-0.15	5.56	0.14	0.02
5	-1.0	-0.15	2.89	-3.50	-1.20	9.56	0.10	0.12

In order to further examine the suitability and effectiveness of the DNA-EA, another algorithm, the DNA-GA we previously used in[6, 20] are performed in the experiments. The DNA-GA uses DNA encoding method with two-point crossover and point mutation genetic operations. The compared performances of the TS fuzzy controllers optimized by the two algorithms are also given in Fig.5. From Fig.5, we can know that the control performances of the TS fuzzy controllers designed by the two algorithms are comparable in controlling the system (6). However, the fuzzy controller optimized by the DNA-GA employs 25 fuzzy rules, while the one designed automatically by the DNA-EA only uses five rules. The supposition made to explain this result is that the gene transfer operation in the DNA-EA is spreading good rules in the population, and thus inducing the improvement of the overall performance of the population. And, the bacterial mutation is efficient in the optimization of local portions of chromosomes.

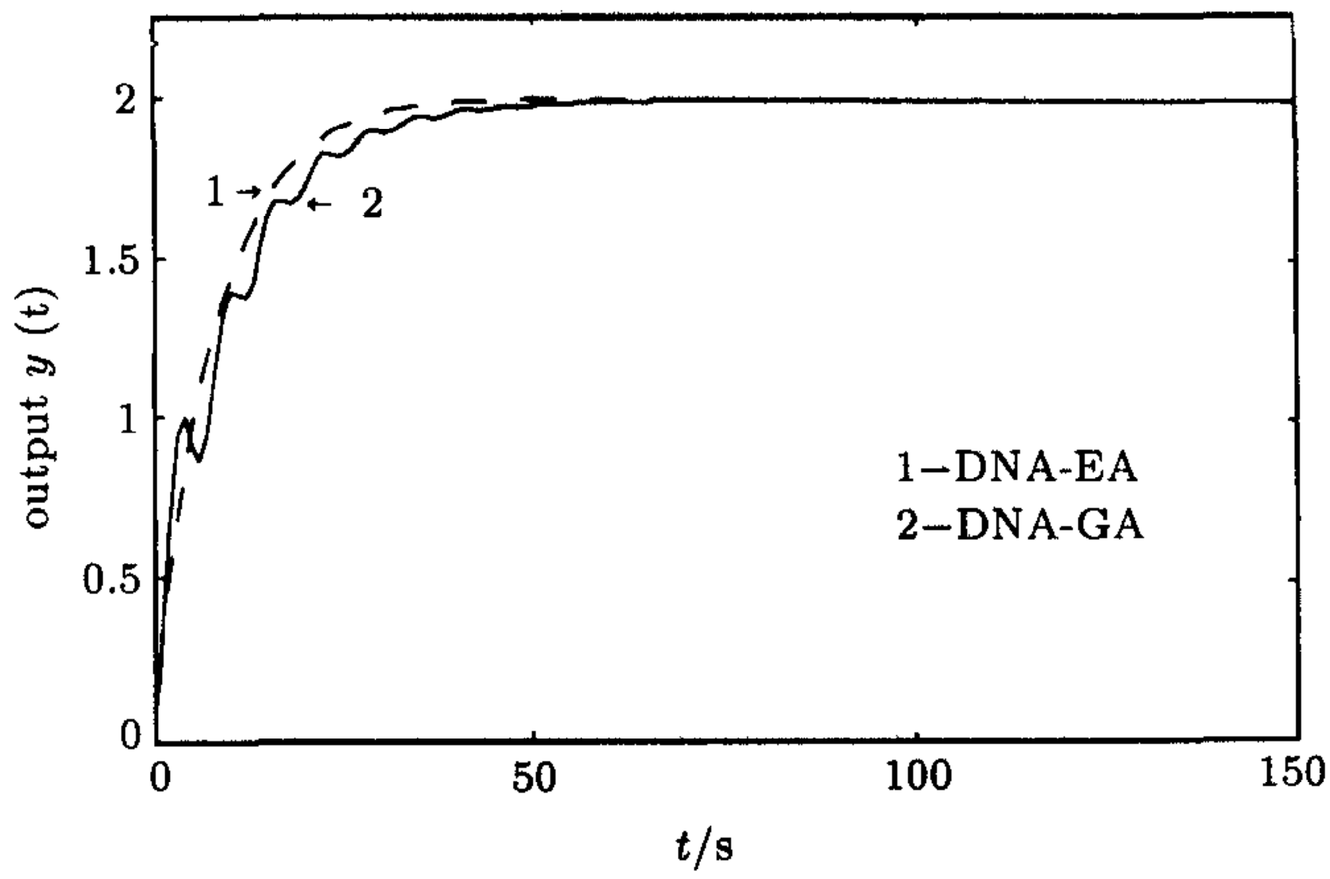

Fig.5　The performance comparison of the TS fuzzy controllers designed by the DNA-EA and the DNA-GA in the control of the nonlinear system (6)

5　Discussions

As we know, there exist some works in literature that both fuzzy rules and tuning parameters are adjusted by the GAs, e.g., [3, 5, 6]. However, the bit string (0-1's) encoding method is often adopted by GA researchers. The configuration of the chromosome is also different from

the one used in the DNA-EA. The genetic operations are also different. In GAs, when many fuzzy rules are used, the length of the individual will be rather long, which will increase the complexity of the problem. In our previous work[20], we have proven that the DNA-GAs are more efficient than the GAs in solving some problems. The DNA-GAs are superior to the GAs in decreasyng the complexity of encoding and searching the solution space effectively. Though it is interesting to compare the approach with other evolutionary computation paradigms, however such comparisons are beyond the scope of this paper. In this paper, we emphasize on possibility of combining the DNA computing with the soft computing. We hope the work in this paper will be a step for future DNA computing with application to the intelligent system. Moreover, since the encoding method and genetic operations are completely different among them, it would be unfair to do such comparison in the simulations.

We should point out that there are several advantages in the DNA-EA:

1) In the DNA-EA, we use the new DNA encoding method that is suitable for complex knowledge representation. That is to say, we have a four-letter alphabet $\sum\{A, G, C, T\}$ to encode information, while in the GAs only need two digits, 0 and 1, for the same purpose. The encoding length of the chromosome can be greatly shortened due to the encoding method of four alphabets.

2) In the DNA-encoded chromosome, we can easily introduce features of the biological DNA into the EAs and develop some new genetic operations. For example, the length of the DNA chromosome is variable and it is easy to insert and delete parts of it by using the new frameshift (i.e., insert and delete) mutation operations. They can be used to add (or delete) a (or several) fuzzy rule(s) to (or from) a fuzzy controller. As such, the structure of the fuzzy controller is automatically built, while in the previous work (e.g., [3, 5, 6]), the structure of the fuzzy controller is first defined, then the design parameters in the input fuzzy sets and rule consequent are designed. The latter should not be the real "automatic design". Similarly, some other genetic operations at the gene level could be introduced into DNA-based algorithms to enrich the DNA-EAs.

3) Inspired by DNA transudation process, we develop and introduce the gene transfer operation into the DNA-EA. The gene transfer operation can spread good rules in the population. The good portions of chromosomes with high fitnesses are directly transferred to the individuals with lower fitnesses, and thus induce the improvement of the overall performance of the population. By doing so, the overall search process should be proceeded more efficient, since the operators actuate more frequently on better rules, leading to the rapid construction of fuzzy systems that fulfill the requirements. Also, bacterial mutation operation inspired by bacterial genetic process is efficient in the optimization of local portions of chromosomes.

4) During the translation process from the DNA chromosome to the design parameters of the fuzzy systems, we employ the translation process of the biological DNA, that is, from 64 triplet codes to 20 amino acids. Different codons are corresponding to the same amino acids could speed up the search process in optimization and could quickly find the expected values of the designed parameters.

5) In the near future, with the development of DNA computer, DNA-based soft computing[14] will have many applications in many scientific and experiment problems. Based on Chen's work[23], we will also possibly fulfill the laboratory implementation of the DNA-EA using the biological technology, which will be useful for the DNA computer or the future DNA intelligent computer.

6 Conclusions

An automatic design method for the TS fuzzy controller is proposed by employing the new DNA-EA. The DNA-EA uses the DNA encoding method and genetic operators inspired from the biological DNA and microbial evolution phenomenon. The DNA-EA can automatically design the TS fuzzy controller and optimize the design parameters in the input fuzzy sets and the rule consequent simultaneously. The computer simulation example is provided to demonstrate the

effectiveness of our new method. The DNA-EA can also be employed to design the other types of fuzzy systems.

Using the DNA encoding method, the genetic operations at gene level can be easily introduced into the DNA-EA. The length of the chromosome is variable and it is easy to insert and delete parts of the chromosome. The next step, some other DNA-based technologies in soft computing, such as DNA-based immune algorithms and DNA-based neural networks should be further advanced. Also, more operations at the gene level should be introduced into the DNA-based algorithms to enrich the evolutionary methods. The DNA-based learning algorithms have potential advantages in many complex practical problems. The work in this paper is expected to have theoretical and practical implications on applications of the fuzzy systems and DNA-based computation.

Acknowledgments The authors thank the suggestions of Dr. Hao Ying at Department of Electrical and Computer Engineering, Wayne State University, Detroit, Michigan, U.S.A.

References

1 Lee C C. Fuzzy logic in control systems: Fuzzy logic controller —— Part I & II. *IEEE Trans. Systems, Man and Cybernetics*, 1990,**20**(2): 404~435

2 Yen J, Langari R, Zadeh L A(Editors). Industrial Applications of Fuzzy Logic and Intelligent Systems. IEEE Press. 1995

3 Berenji H R. Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Trans. Neural Networks*, 1992, **3**: 724~740

4 Jang J R. Self-learning fuzzy controllers based on temporal back propagation. *IEEE Trans. Neural Networks*, 1992, **3**: 714~721

5 Karr C L, Gentry E J. Fuzzy control of pH using genetic algorithms. *IEEE Trans. Fuzzy Systems*, 1993, **1**: 46~53

6 Lim M H, Rahardja S, Gwee B H. A GA paradigm for learning fuzzy rules. *Fuzzy Sets & Systems*, 1996, **82**: 177~186

7 Matsushita S, Furuhashi T, Sutsui H T, Uchikawa Y. Efficient search for fuzzy models using genetic algorithm. *Information Sciences*, 1998, **110**: 41~50

8 Shi Y H, Eberhart R, Chen Y B. Implementation of evolutionary fuzzy systems. *IEEE Trans. Fuzzy Systems*, 1999, **7**(2): 109~119

9 Wang C H, Hong T P, Tseng S S. Integrating fuzzy knowledge by genetic algorithms. *IEEE Trans. Evolutionary Computation*, 1998, **2**(4): 138~149

10 Adleman L M. Molecular computation of solutions to combinatorial problems. *Science*, 1994, **266**(5187): 1021~1023

11 Kari L. DNA computing: arrival of biological mathematics. *The Mathematical Intelligence*, 1997,**19**(2): 9~22

12 Lipton R J. DNA solution of hard computational problems. *Science*, 1995, **268**(5210): 542~545

13 Ren L H, Ding Y S, Shao S H. The study of DNA computing: Current state and future directions. *Information and Control* (in Chinese), 1999, **28**(4): 241~248

14 Ren L H, Ding Y S, Shao S H. DNA bio-soft computing and its application to intelligent control systems. *J. Shanghai Jiaotong University (English version)*, 1999, **E-4**(2):95~101

15 Deaton R, Murphy R C, Rose J A, Garzon M, Franceschetti D R, Jr Stevens S E. A DNA based implementation of an evolutionary search for good encodings for DNA computation. In: Proc. 1997 IEEE Int. Conf. Evolutionary Computation, Indianapolis, IN, USA: 1997. 267~271

16 Ding Y S, Ren L H, Shao S H. Fuzzy approximators based on DNA genetic algorithms. In: The Third Asian Control Conference, Shanghai, China: 2000. 4~7

17 Doi H, Furusawa M. Evolution is promoted by asymmetrical mutations in DNA replication-genetic algorithm with double-stranded DNA. *Fujitsu Scientific and Technical Journal*, 1996, **32**(2): 248~255

18 Farhat N H, Hernandez E D M. Logistic networks with DNA-like encoding and interactions. In: Proc. Int. Workshop on Artificial Neural Networks, Malaga-Torremolinos, Spain: 1995. 215~222

19 Russo M F, Huff A C, Heckler C E, Evans A C. An improved DNA encoding scheme for neural network modeling. In: Neural Network Society Annual Meeting, San Diego, CA, USA: 1994. I/354~359

20 Ren L H. Artificial DNA Computation Models-Based Intelligent Systems and Their Applications: [Ph.D. Dissertation], 2000
21 Yoshikawa T, Furuhashi T, Uchikawa Y. DNA coding method and a mechanism of development for acquisition of fuzzy control rules. In: Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, New Orleans, LA, USA: 1996. **3**: 2194~2200
22 Yoshikawa T, Furuhashi T, Uchikawa Y. The effects of combination of DNA coding method with Pseudo-Bacterial GA. In: Proc. 1997 IEEE Int. Conf. Evolutionary Computation, Indianapolis, IN, USA: 1997. 285~290
23 Chen J H, Antipov E, Lemieux B, Cedeno W, Wood D H. DNA computing implementing genetic algorithms. DIMACS Workshop on Evolution as Computation. Princeton University, 1999
24 Yan L F, Zhang Y L (Editors). Molecular biology. The Press of China Agriculture University, 1997

采用新的 DNA 进化算法自动设计 Takagi-Sugeno 模糊控制器

丁永生 任立红 邵世煌

(东华大学信息科学与技术学院自动化系 上海 200051)
(E-mail: ysding@dhu.edu.cn, shshao@dhu.edu.cn)

摘 要 提出一种新颖的基于 DNA 的进化算法 (DNA-EA) 来自动设计一类 Takagi-Sugeno(TS) 模糊控制器. TS 模糊控制器采用带有线性规则后项的 TS 模糊规则, 连续输入模糊集, Zadeh 模糊逻辑和常用的重心反模糊器. TS 模糊控制器被证明是带有可变增益的非线性 PI 控制器. DNA-EA 被用于自动获取 TS 模糊规则, 并同时优化模糊规则前项和后项中的设计参数. DNA-EA 采用由生物 DNA 结构启发得到的 DNA 编码方法来编码模糊控制器的设计参数. 在 DNA-EA 中, 引入了受微生物进化现象启发的基因转移和细菌变异操作. 另外, 也引入了基于 DNA 遗传操作的框构变异操作. DNA 编码方法非常适合于复杂知识的表达, 基于基因水平的遗传操作也很容易引入到 DNA-EA 中. 染色体的长度是可变的, 且可插入或删除部分碱基序列. 作为示例, 给出了采用 DNA-EA 来自动设计 TS 模糊控制器用于控制一类非线性系统的方法. DNA-EA 能自动地构造模糊控制器. 计算机仿真结果表明, DNA-EA 是有效的, 且优化得到的模糊控制器是满意的.

关键词 DNA 编码方法, 基因转移操作, 框构变异, 进化算法, TS 模糊控制.

丁永生 博士, 东华大学 (原中国纺织大学) 副教授, 1996~1998 年作为访问学者在美国得克萨斯大学医学院工作. 研究方向为模糊控制、智能控制、 DNA 计算、人工免疫系统、软计算、网络智能自动化和非线性系统控制.

任立红 博士, 东华大学副教授. 研究方向为 DNA 计算、软计算和智能控制.

邵世煌 1960 年毕业于东南大学自动控制系, 现为东华大学校长、教授、博士生导师. 研究方向为模糊控制、智能控制、非线性系统控制、智能 CAD 、 DNA 计算和 Internet 智能自动化.