*Tạp chí Khoa học và Công nghệ **51** (3) (2013) 267-277*

MARKOV MODEL IN PROVING THE CONVERGENCE OF FUZZY GENETIC ALGORITHM

Tran Manh Tuan[1, *], Le Ba Dung[2]

[1] *School of Information and Telecommunication, Thai Nguyen University*

[2] *Institute of Information Technology, VAST, 18 Hoang Quoc Viet, Cau Giay, Hanoi*

[*] Email: *tuan.cntt.dhtn@gmail.com*

Đến Tòa soạn: 27/4/2012; Chấp nhận đăng: 29/5/2013

ABSTRACT

Genetic Algorithms (GA) was concerned by many authors and researchers from all over the world. There were results in different fields of our lives. But the convergence of GA is an open problems. In this paper, we propose a method using Markov model to prove the convergence of GA. At first, in section 2, we review fundamental concepts in Markov Model, then we present important role of Markov model in GA (section 3). After that, in section 4, we show the weak convergence of GA base on Markov model. In the end, in section 5, we also illustrate these using experiment results.

Keywords: genetic algorithm, fuzzy theory, fuzzy rules, neural network.

1. INTRODUCTION

Learning system based on fuzzy rules with genetic algorithms can lead to effective expression of some problems. Fuzzy logic (FL) gives a simple way for defined conclusion from input data which is unclear, noisy effected by other sources or lost information. FL model is based on users' experiences rather than based on their technique knowledge about system. In fuzzy logic methods, any suitable number of input factors can be processed and we have a huge output, though defining a rule base becomes complex if too much inputs and outputs are selected for simple execution because rules used for determining their correlation have to be defined. This leads to increase in fuzzy rules and complexity but it can increase quantity of control. Many methods are proposed to make fuzzy rule base. Basic idea is to research and to make necessary optimal rules to control input data without verifying the quantity of control. This paper presents a generation of fuzzy rule base by fuzzy clustering technique and use GA for optimizing obtained fuzzy rules.

Firstly, we classify numerical data. The purpose of this step is to get a set of data from a large amount of given data and to have a short representation about behavior of our system. Several different clustering algorithms have been developed. Fuzzy C-means (FCM) is one of these that have widely used. In fact, begin with a certain number of clusters and a prediction for each center of cluster, this algorithm converge to the solution that can be local minimum point. This point is center of cluster in future.

Each cluster presents definite part of system behavior, represented by a IF-THEN rule. Then, each cluster is projected in one dimension of input space. In the input space, each projection makes a priority of one rule. The rules are formed [1]:

R_1: If x_1 is A_1^1 and x_2 is A_2^1 then $y_1 = p_0^1 + p_1^1\, x_1 + p_2^1\, x_2$

.....

R_K: If x_1 is A_1^K and x_2 is A_2^K then $y_k = p_0^K + p_1^K\, x_1 + p_2^K\, x_2$

where as two input variables x_1 and x_2 are process time and priority (or weight) but output variable y_k ($k = 1, 2, \ldots, K$) is select index (or sequent index) of the k^{th}-rule. A_1^k and A_2^k ($k = 1, 2, \ldots, K$) are fuzzy sets of k^{th} –rule obtained by projecting clusters to domains of process time and priority, correspondingly and p_i^k ($i=1,2; k=1, 2, \ldots, K$) is result recurrent parameters.

The next step is to define optimal result parameters by using training data set (x_{1t}, x_{2t}, y_t), $t = 1, 2, \ldots, m$. With this training set, output can be written:

$$y_t = \frac{\sum_{k=1}^{K} w_t^k\, y_t^k}{\sum_{k=1}^{K} w_t^k} = \sum_{k=1}^{K} \beta_t^k\, y_t^k \tag{1}$$

in which: $w_t^k = (A_1^k(x_{1t}) \wedge A_2^k(x_{2t}))$ và $\beta_t^k = w_t^k / \sum_{k=1}^{K} w_t^k$

In matrix form, system output [2]: $[Y] = [X][P]$, where input matrix, defined in each rule, is:

$$X = \begin{bmatrix} \beta_1^1,..,\beta_1^K,\beta_1^1 x_{11},..,\beta_1^K x_{11},\beta_1^1 x_{21},..,\beta_1^K x_{21} \\ \ldots \\ \beta_m^1,..,\beta_m^K,\beta_m^1 x_{1m},..,\beta_m^K x_{1m},\beta_m^1 x_{2m},..,\beta_m^K x_{2m} \end{bmatrix}$$

and output matrix is known as: $Y = [y_1,\ldots,y_m]^T$

Then, P is formed:

$$P = (X^T X) X^T Y = CMS \tag{2}$$

where as $C= (X^T X)$, $M = X^T$, $S = Y$.

2. MARKOV MODEL

Markov model is defined as the development of system in the future only depends on present and not on the past. In [4], we have: C, M, S are random matrix. In which, if M is positive matrix and S is column satisfaction then $CxMxS$ is a positive random matrix. Where:

- $A = (a_{ij})$ is positive (denote $A > 0$) if $a_{ij} > 0$ $(\forall i, j = 1,..,n)$

- Matrix is random if $\sum_{j=1}^{n} a_{ij} = 1 \qquad (\forall i = 1,..,n)$

Moreover, assumption that *P* is reducible random matrix and *C* is primative random matrix size m and *R,T* are non-zero matrixes. Then:

$$P^{\infty} = \lim_{k\to\infty} P^k = \lim_{k\to\infty} \begin{pmatrix} C^k & 0 \\ \sum_{i=0}^{k-1} T^i R C^{k-1} & T^k \end{pmatrix} = \begin{pmatrix} C^{\infty} & 0 \\ R_{\infty} & 0 \end{pmatrix} \tag{3}$$

is stable random ma trix with $P^{\infty} = 1' p^{\infty}$ where $p^{\infty} = p^0 P^{\infty}$ doesn't depends on original distribution and p^{∞} satisfies $p_i^{\infty} > 0$, with all $i \in \{1,...,m\}$ and $p_j^{\infty} = 0$, with $m < j \le n$.

Markov process: we consider a process …depends on time. At the point *t = 0, X* can have a random state in state space *S*. Denote *X(t)* is state of *X* at the time *t*. Thus, corresponding to a point of time *t*, *X(t)* is a random variable describing state of progress. $\{X(t)\}_{t\geq 0}$ is called a random process.

Assumption that, before the point of time *s*, random process has some state, at the point of time *s*, it is at state *i*. We consider the probability for at the point *t, (t > s)*, the process is at state *j*. If this probability only depends on the group of four *(s,i,t,j)* as (4):

$$P[X(t)=j|X(s)=i] = p(s,i,t,j), \quad \forall s,i,t,j \tag{4}$$

then the process's development in the future is only depends on the present but does not depend on the past (non-remember feature). This is Markov feature. A random process X(t) having Markov feature as above is called Markov process. If the state space S consists of a uncountable number of states then Markov process X(t) is called a Markov chain.

We have a finite Markov chain *X* on state space *S = {1,...,T}*. If at *t = n*, *X(n)* get one of these values *1,2,...,T* corresponding to the probabilities $\pi_1^{(n)}, \pi_2^{(n)}, ..., \pi_T^{(n)}$ ($\pi_1^{(n)} + \pi_2^{(n)} + ... + \pi_T^{(n)} = 1$) then vector $(\pi_1^{(n)}, \pi_2^{(n)}, ..., \pi_T^{(n)})$ is called probabilities distribution vector at the time *t = n*. At *t = 0*, we have generation probabilities distribution vector $\Pi^{(0)} = (\pi_1^{(0)}, \pi_2^{(0)}, ..., \pi_T^{(0)})$.

$$\Pi^{(n)} = (\pi_1^{(n)}, \pi_2^{(n)}, ..., \pi_T^{(n)}) \tag{5}$$

and $p_{ij} = p(t,i,t+1,j) = P[X(t+1)=j|X(t)=i]$ ($\forall t \in N$) is state transfer probability from state i to state j after one step, $\forall i,j \in \{1,2,..,T\}$, then matrix P = $[p_{ij}]_{T\times T}$, is called state transfer probability matrix or Markov chain's state transfer probability matrix. After every one step, the process transfers from state i to some state j with probability $p_{ij} \geq 0$, thus we have:

$$\sum_{j=1}^{T} p_{ij} = 1 \tag{6}$$

as defination, state transfer probability matrix P is a random.

3. MARKOV MODEL IN GENETIC ALGORITHM

3.1. The finite population model

The finite population model is an important tool in modeling the process of implementing finite population genetic algorithm (GA). For simple genetic algorithms (GAs), the model is presented through out a function called heuristic function G. This forms out a function, M, describing the combination (between crossover and mutation) and another function, F for the selection. We have $G = M \circ F$ is used for identifying the presentation of binary strings. With a binary string presentation, M can describe crossover and mutation operators based on the crossover and mutation masks, including one point, two point and synchronous crossover. F can key out clear and well-proportioned selection. G has three expressions. Firstly, it is a finite presentations of a population in the simple GA, such as population size is finite. Secondly, if the population in present generation is x, then with every selection an individual for population in next generation, probability of individual $i \in \Omega$ added is $G(x)_i$. Third, $G(x)$ is expected population of the next generation [3].

3.2. Random Heuristic Searching model (RHS)

This model show the way to transfer a Heuristic function as G in simple GA to a finite population model. This model consists of four steps as follow:

- Step1: Select a initial random population X size r.
- Step 2: Select r random independent samples from distribution G(X/r) and set as a new population Y.
- Step 3: Replace X by Y.
- Step 4: Repeat step 2.

When this model is used for simple GA, G is the function of finite population model shown above. The random algorithms such as simulation training and genetic programming can be modeled by using a suitable function G.

RSH model supposes that a GA follows the generations and each individual of next generation is independently selected with present population. Many GAs used commonly, stable state algorithm is an example, are not suitable with this instance. The simple GA is modeled as a Markov chain [2]. The states of Markov chain are populations. There are $N = \binom{n+r-1}{r}$ such populations. Transfer probability of Markov chain can be obtained by using polynomial probability distribution. If X and Y corresponding are populations size r, then transfer probability from X to Y is shown as:

$$P(X,Y) = r! \prod_{j \in \Omega} \frac{\mathcal{G}(X/r)_j^{Y_j}}{(Y_j)!}$$

Above formula can be understood as sample space is a set of ordered strength r strings in Ω. A population can be considered as a equivalent class of ordered string, where two strings are equivalent if one can be rearranged to become another. Probability of any string corresponding with population Y is $\prod \mathcal{G}(X/r)_j^{Y_j}$. The number of these strings is $r!/\prod Y_j!$

We hope a general version of RSH formula using to compute probability of selection a population Y size k from a population X size r by heuristic H. Thus, we define function *R* as follow:

$$R(\mathcal{H}, X, Y) = k! \prod_{j \in \Omega} \frac{\mathcal{H}(X/r)_j^{Y_j}}{Y_j!}$$

4. WEAK CONVERGENCE IN GENETIC ALGORITHM AND ANALYSIS PROCESS BASED ON MARKOV CHAIN

When applying GA for solving complex and large scale real world problems, one of the difficulties is convergence. Weak convergence happens when population in GA get optimal state, that means most of genetic operators can not generate a child populations which more perfect than their parents. Some methods has been proposed to carry out this difficulty including selection procedure restriction method, change the fitness rate…. However, these methods are heuristic. Their effects changes in different problem and their application trends need to be justified for each suitable particular instance.

A trouble in researching about weak convergence is to identify when it happens and describe its range. For example, Srinivas and Patnaik [8] used the difference between average fitness value and maximum one for measure weak convergence, although not measure density of algorithm and change crossover and mutation probabilities reasonably following the measurement. On the other hand, population diversity in quantitative definition is used in many papers for studying this. It easy to recognize that, decreasing population diversity straightly leads to weak convergence. However, there are some efforts in quantitative analysis process about population diversity to use it as a tool preventing weak convergence.

Population diversity rate and its Markov analyzing

In this section, concept "population diversity rate" is proposed and defined as a way following correctly to the concepts of population diversity that not characterized strictly in semantic meaning. Then, we use concept definition for studying the problem about weak convergence of GA by using Markov chain combination with simple GA operators.

Considering GAs with binary presentations length l and a fixed size N, we have following definitions:

Definition 4.1. Let $\vec{X} = (X_1, X_2, ..., X_n) \in S^n$ is a population. Population diversity $\vec{X}$, denoted by $\lambda(\vec{X})$, is defined as a number of elements in vector $\sum_{i=1}^{n} X_i$ that their values not equal to 0 or N. Thus, we have $\beta(\vec{X}) = l - \lambda(\vec{X})$.

If denoting $\vec{X}$ in matrix form, then $\lambda(\vec{X})$ is number of columns in $\vec{X}$ that their elements of column get both 0 and 1 values. Specially, all of individuals in $\vec{X}$ are completely defined with every $\lambda(\vec{X})$ = 0. Otherwise, $\beta(\vec{X})$ is only number of columns consisted of all elements that equal 0 or 1 of $\vec{X}$.

A schema L([1]) is a hyperspace in individual space S and can be presented as follow:

$$\mathbf{L} = \{X = (x_1, \cdots, x_l) \in S;\ x_{i_k} = a_{i_k},\ 1 \le i_k \le l,\ 1 \le k \le K\},$$

where K ($1 \le K \le l$) is order of L, {i_1, …,i_K} is values of identified elements. A schema order K includes 2^{l-K} different individuals.

Definition 4.2. Let $\vec{X} = (X_1, X_2, ..., X_n) \in S^n$ is a population with population diversity rate $\lambda(\vec{X})$ and mature rate $\beta(\vec{X}) = l - \lambda(\vec{X})$. Let i_k, $1 \le k \le \beta(\vec{X})$ is components that all individuals of $\vec{X}$ get the same value. Then, we have $a_{i_k} \in \{0,1\}$, $1 \le k \le \beta(\vec{X})$. We call schema $L(a_{i_1},...,a_{i_{\beta(\vec{X})}})$ is minimum schema containing $\vec{X}$ and denote it by $L(a_{i_1},...,a_{i_{\beta(\vec{X})}};\vec{X})$ or $L(\vec{X})$ for simple (if there is not confusion).

To determine the effect of population diversity, we assume that mutation probability equals 0.

Theorem 4.1. Let $\{\vec{X}(k), k \ge 0\}$ is a population Markov chain with $p_m = 0$ and $\vec{X}(0) = \vec{X}_0$.

a, For every Y ∈ $L(a_{i_1},...,a_{i_{\beta(\vec{X}_0)}};\vec{X}_0)$, exist a mumber n ≥ 0 so that

$$P\{Y \in \vec{X}(n) / \vec{X}(0) = \vec{X}_0\} > 0$$

b, For every Y ∉ $L(a_{i_1},...,a_{i_{\beta(\vec{X}_0)}};\vec{X}_0)$ and every n ≥ 0 we have

$$P\{Y \in \vec{X}(n) / \vec{X}(0) = \vec{X}_0\} = 0$$

Lemma 4.1. Let $\{\vec{X}(k), k \ge 0\}$ is a population Markov chain of the Canonical GA with p_m=0 and B is a set of homogeneous populations, for example B = {(X, X, …, X); X∈S}. Then, for every n ≥ 1, we have

$$P\{\vec{X}(n) \in \mathbf{B} / \vec{X}(0) \in \mathbf{B}\} = 1.$$

Theorem 4.2. Let $\{\vec{X}(k), k \ge 0\}$ is a population Markov chain of a GA with $p_m = 0$ and B is a set of homogeneous populations.

a, $\{\vec{X}(k), k \ge 0\}$ converges to B with probability 1, it means:

$$P\{\lim_{k \to \infty} \vec{X}(k) \in \mathbf{B}\} = 1.$$

b, Diversity of a series of populations is uniform decreasing with probability 1, strictly decreasing with positive probability and converges to zero with probability 1, these mean:

$$P\{\lambda(\vec{X}(k+1)) \le \lambda(\vec{X}(k))\} = 1,\ k \ge 0$$
$$P\{\lambda(\vec{X}(k+1)) < \lambda(\vec{X}(k))\} > 0,\ k \ge 0$$
$$P\{\lim_{k \to \infty} \lambda(\vec{X}(k)) = 0\} = 1$$

Definition 4.3. Given a population $\vec{X} = (X_1, X_2, ..., X_N)$, where $X_i = (x_{i1}, \ldots, x_{il})$, i =1,…,N, for every interger *m*, $1 \le m \le l$, set I_0^m và I_1^m is index sets of all individuals in population $\vec{X}$, corresponding get 0 or 1 at *m*th gene position, it means:

$$I_0^m = \{i \in \{1, 2, \cdots, N\};\ x_{im} = 0\}$$
$$I_1^m = \{i \in \{1, 2, \cdots, N\};\ x_{im} = 1\}$$

and write

$$f_0^m(\vec{X}) = \sum_{i \in I_0^m} f(X_i), \quad f_1^m(\vec{X}) = \sum_{i \in I_1^m} f(X_i).$$

We call:

$$a_m = \frac{f_0^m(\vec{X})}{\sum_{j=1}^{N} f(X_j)}, \quad b_m = 1 - a_m = \frac{f_1^m(\vec{X})}{\sum_{j=1}^{N} f(X_j)},$$

corresponding are fitness rate 0 and 1 of the individuals at m^{th} gene position.

With the results shown in [9], we can see that weak convergence probability at a gene position is independent with crossover probability. Thus, said that crossover probability justifying method prevents weak convergence has not theory basis. Crossover probability justifying can promote the searching rate of minimum schema containing present population.

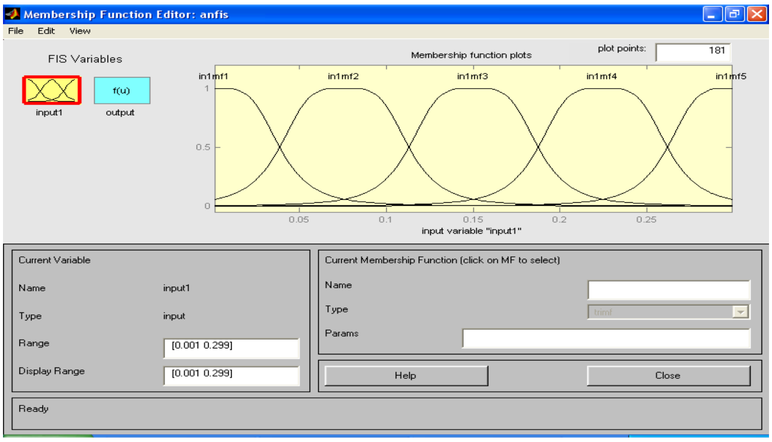
5. EXPERIMENT RESULTS

Global optimal solution for mathematic model.

Mathematic model (7) is used to find global optimal point as the method presented above. As we known, on mathematic form, there are many ways to find the globe optimal solution of (7). Here, using fuzzy genetic algorithm to find the global optimal model or optimal point is meaningful. Looking for the optimal solution is constructed by using fuzzy model (on other words, knowledge model). It's suitable for complex problems, less-information problems and hard to construct a mathematic model satisfying the given requirement. In this example, we have many optimal points. However, we find the global optimal rapidly by using fuzzy genetic algorithm.

$$f(x) = 10 + \left[\frac{1}{\{(x-0.16)^2 + 0.1\}}\right]\sin\left(\frac{1}{x}\right) \tag{7}$$

For searching global optimal for (7), we apply Fuzzy GA by above steps. Fuzzy model constructing process of (7) is taken part by choosing member function in Figure 1. After that, modifying process is carried out to get the best result in with *(x = 0.127 ; f = 19.89)* and member function is presented in figure 2.



plot points: 181

Current Variable

Name: input1

Type: input

Range: [0.001 0.299]

Display Range: [0.001 0.299]

Current Membership Function (click on MF to select)

Name:

Type: trimf

Params:

Ready

Figure 1. Form of member function after searching optimal solution by GA

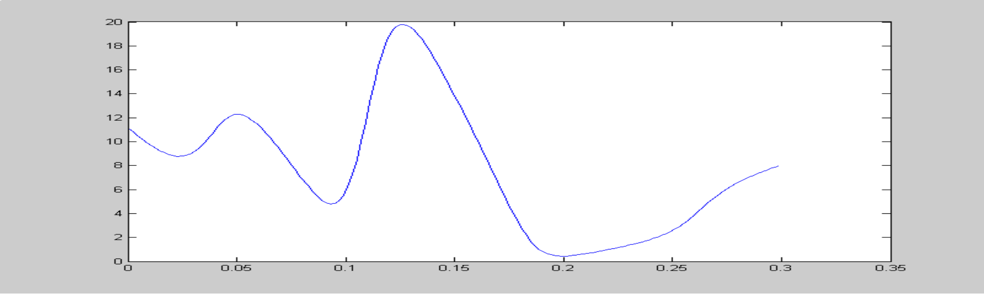


Figure 2. The global optimal solution using fuzzy GA.
x = 0.127, f = 19.89

Searching optimal model for nonlinear kinematic system.

Assumption that, we have a nonlinear kinematic system presented by this equation:

$$y(k) = h(y(k-1), y(k-2)) + u(k) \tag{8}$$

where as

$$h(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)(y(k-1)-0.5)}{1+y^2(k-1)y^2(k-2)} \tag{9}$$

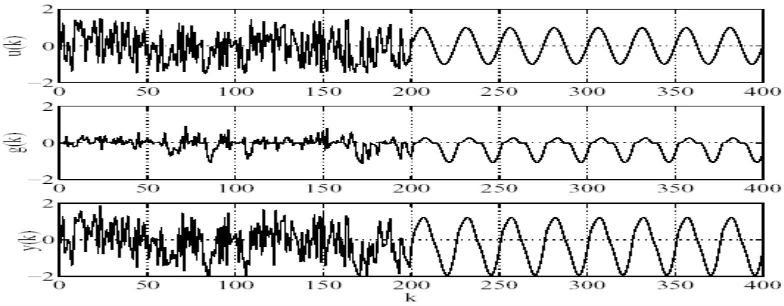


Figure 3. Inputs $u(k)$, $g(k)$ of evolution process.

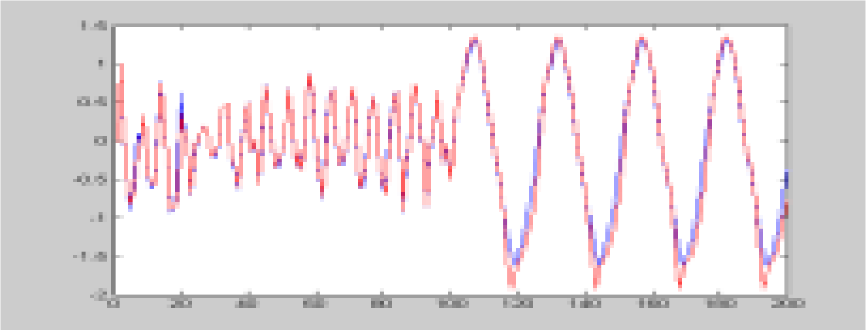


Figure 4. Output $y(k)$ of evolution process

Our purpose is to approximate nonlinear element $g(y(k-1), y(k-2))$ of this process by using evolution model with fuzzy logic in 400 training data points obtained from evolution model. Begin from balance state (0,0), two hundred samples of identifying data are defined with input signal $u(k)$, chosen randomly, has steady distribution in (-1.5, 1.5). In period from 200 to 500, $u(k)$ is formed $u(k) = \sin(2\pi k / 25)$. Then we have output in figure 3. The result is presented in figure 4.

When comparing results obtained by three different approaches, we sumary it by the convergence (MSE) of GA applied for TSK model [4, 5] with linear constant: training data (bold line), testing data (dash line) in figure 4.

In general, experiment result shows that crossover probability P_c chosen approximately 0.6 and mutation probability P_m is small. If mutation happens too often, evolution process will be a random process. Otherwise, if mutation probability is too small, searching process will be a local searching. Hence, we have to choose mutation probability from 0.01 to 0.05.

TSK fuzzy rule system for model constructing and form of fuzzy set.

R_1 If y_{k-1} small and y_{k-2} small then $g = 0.452y_{k-1} + 0.166y_{k-2} + 0.143$

R_2 If y_{k-1} small and y_{k-2} large then $g = -0.419y_{k-1} + 0.157y_{k-2} - 0.097$

R_3 If y_{k-1} large and y_{k-2} small then $g = -0.269y_{k-1} + 0.090y_{k-2} + 0.137$

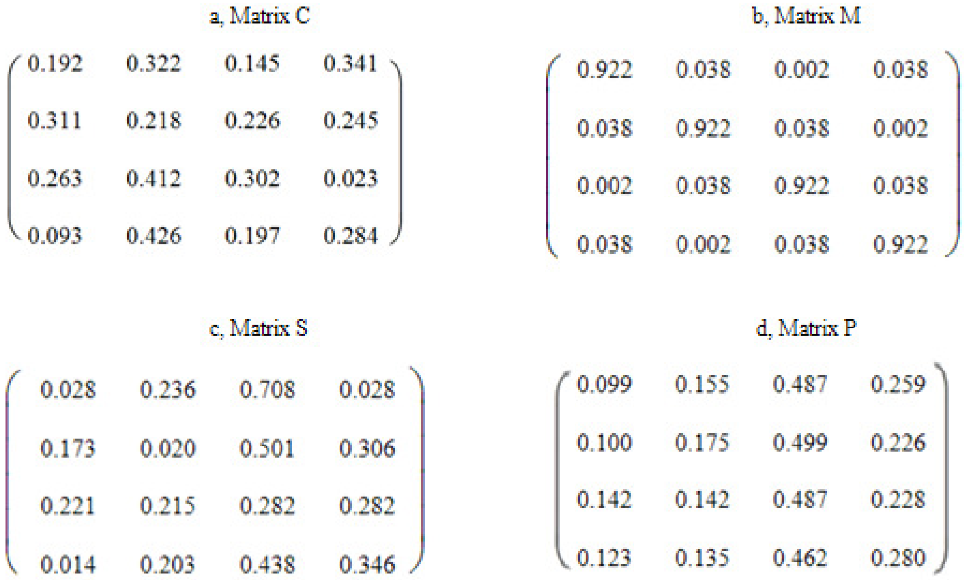
R_4 If y_{k-1} large and y_{k-2} large then $g = 0.323y_{k-1} + 0.054y_{k-2} + 0.176$

The parameters of fuzzy terms are as following:

y_{k-1} small $= (-4.789, -1.445, 1.272)$; large $= (-0.608, 0.975, 4.789)$

y_{k-2} small $= (-3.175, -0.628, 0.660)$; large $= (-0.792, 0.989, 2.712)$

For example: Consider to one population (n = 1) including 2 bit strings (l = 2), the state space has 4 states: 00, 01, 10, 11 then C, M and S are 4 x 4 size matrices as below:



a, Matrix C

$$\begin{pmatrix} 0.192 & 0.322 & 0.145 & 0.341 \\ 0.311 & 0.218 & 0.226 & 0.245 \\ 0.263 & 0.412 & 0.302 & 0.023 \\ 0.093 & 0.426 & 0.197 & 0.284 \end{pmatrix}$$

b, Matrix M

$$\begin{pmatrix} 0.922 & 0.038 & 0.002 & 0.038 \\ 0.038 & 0.922 & 0.038 & 0.002 \\ 0.002 & 0.038 & 0.922 & 0.038 \\ 0.038 & 0.002 & 0.038 & 0.922 \end{pmatrix}$$

c, Matrix S

$$\begin{pmatrix} 0.028 & 0.236 & 0.708 & 0.028 \\ 0.173 & 0.020 & 0.501 & 0.306 \\ 0.221 & 0.215 & 0.282 & 0.282 \\ 0.014 & 0.203 & 0.438 & 0.346 \end{pmatrix}$$

d, Matrix P

$$\begin{pmatrix} 0.099 & 0.155 & 0.487 & 0.259 \\ 0.100 & 0.175 & 0.499 & 0.226 \\ 0.142 & 0.142 & 0.487 & 0.228 \\ 0.123 & 0.135 & 0.462 & 0.280 \end{pmatrix}$$

Figure 5. The matrices of the crossover (a), mutation (b), selection (c) and the result matrix (d).

6. CONCLUSION

This paper presented a technique for proving the convergence of GA combination with fuzzy theory. Data obtained from knowledge in the real world, is used for fuzzy clustering to get fuzzy rules. After that, using GA to optimize these rules. When the sample set is large, the number of rules is based on the mumber number of clustering centers . From this result, we propose a effective method to prove the convergence of a important algorithm in GA by using fuzzy clustering and GA.

REFERENCES

1. Siamak Tafazoli, Mathieu Leduc and Xuehong Sun - Hysteresis Modeling using Fuzzy Subtractive Clutering, International Journal of Computational Cognition 4 (3) (2006) 15-27.
2. A E Nix, M D Vose, Modeling genetic algorithms with markov chains, Annals of Mathematics and Artificial Intelligence 5 (1992) 79-88
3. Michael D. Vose , *The Simple Genetic Algorithm: Foundations and Theory*, MIT Press Publisher, ISBN-13 (1999).
4. Agus Priyono, Muhammad Ridwad Jais Alias, Riza AtiQ O.K.Rahmat, Azmi Hassan, Mohd.Alauddin Mohd.Ali - Generation of fuzzy rules with subtractive clusterring, Universiti Teknologi Malaysia, Journal Teknologi 43 (D) (2005) 143-153.
5. Mohammad Ghasemi Gol, Hadi Saoghi Yazdi, Reza Monsefi - *A New Hierarchical Clustering Algorithm on Fuzzy Data (FHCA)*, International Journal of computer and electrical engineering 2 (1) (2010) 134 - 140 .
6. Doan C. D., Liong S. Y., and Dulakshi S. K. Karunasinghe - Derivation of effective and effcient data set with subtractive clustering method and genetic algorithm, Journal of Hydroinfomatics **7** (4) (2005) 219 - 233.
7. Lothar M. Schmitt - Fundamental Study Theory of genetic algorithms, Theoretical Computer Science **59** (2001) 1-61
8. Gunter Rudolph - Convergence Analysis of Canonical Genetic Algorithms, IEEE transaction on neural networks **5** (1) (1994) 96 - 101.
9. Mohanad Alata, Mohammad Molhim, and Abdullah Ramini - Optimizing of Fuzzy C-Means Clustering Algorithm Using GA, World Academy of Science, Engineering and Technology (2008) 224-229.

TÓM TẮT

MÔ HÌNH MARKOV TRONG CHỨNG MINH SỰ HỘI TỤ CỦA GIẢI THUẬT DI TRUYỀN MỜ

Trần Mạnh Tuấn[1], Lê Bá Dũng[2]

[1]Trường Đại học Công nghệ thông tin và Truyền thông, Đại học Thái Nguyên,
Xã Quyết Thắng, Thành phố Thái Nguyên

[2]Viện Công nghệ thông tin, Viện HLKHCNVN, 18 Hoàng Quốc Việt, Cầu Giấy, Hà Nội

[*]Email: tuan.cntt.dhtn@gmail.com

Giải thuật di truyền (GA) đã được nhiều tác giả và các nhà nghiên cứu trên toàn thế giới quan tâm đến. Đã có các kết quả trong những lĩnh vực khác nhau của đời sống. Nhưng sự hội tụ của GA vẫn còn là một vấn đề ngỏ. Trong bài báo này, chúng tôi đề xuất một phương pháp dùng mô hình Markov để chứng minh sự hội tụ của GA. Trước hết, ở mục 2, chúng tôi nhắc lại các khái niệm cơ bản trong mô hình Markov. Sau đó, trong mục 3, chúng tôi trình bày vai trò quan trọng của mô hình Markov trong GA. Tiếp theo, trong mục 4, chúng tôi chỉ ra sự hội tụ yếu của GA dựa trên mô hình Markov. Cuối cùng, trong mục 5, chúng tôi minh họa các lí thuyết trên bằng các kết quả thực nghiệm.

Từ khóa: giải thuật di truyền, lí thuyết mờ, các luật mờ, mạng nơron.