An Effective Local Search Operation of Genetic Algorithm for Fuzzy Modeling

Makoto Ohki, Naruhito Kishimoto, Shigehiko Hasegawa and Masaaki Ohkita
Department of Electrical and Electronic Engineering, Faculty of Engineering, Tottori University
101, 4 Koyama-Minami, Tottori, Tottori 680-8552 Japan
Phone: +81-857-31-6701, Fax.: +81-857-31-0880
email: mohki@ele.tottori-u.ac.jp

ABSTRACT: We propose a new method which reduces the number of generations in a fuzzy modeling by using a genetic algorithms. We have already proposed two methods using cyclic and linear variable bit-selection probability. These two methods allow the optimization of the fuzzy model to be accelerated while these have different features each other. In this paper, these two different types of bit-selection probability are combined as the new method, namely a mixed variable bit-selection probability. By means of a mutation operator using the mixed variable bit-selection probability, a good fuzzy model giving a good performance can be quickly found. Efficiencies of the mixed variable bit-selection probability is shown in examples of function approximation which is considered as a generaliged problem of the fuzzy modeling.

KEYWORDS: genetic algorithm, fuzzy modeling, mutation operator, bit-selection probability, variable bit-selection probability

1. INTRODUCTION

A problem of the fuzzy modeling[1] consists of two kinds of problems, namely configuring a fuzzy rule set and determining shapes of membership function (MSF). These problems are considered as combinatorial and numerical optimization problems. A genetic algorothm (GA) [2, 3] can be applied to both those problems. There are a lot of research works [3, 4, 5] on the fuzzy modeling using the GA. Although, in most of the research works, the GA is applied to configuration of the fuzzy rule, another optimizing algorithms, for example, steepest descent and pattern search method, are applied to tune the shapes of the MSFs. This is caused by weak local search of the GA, and then the GA is slow in the numerical optimization.

To improve the local search of the GA, we have proposed two types of variable bit-selection probability (variable BSP : VBSP) [6, 7, 8, 9]. One of them is a linearly varying VBSP (LVBSP)[8]. In the LVBSP, the BSP is defined by a triangular function. The top of the triangular function gradually varies from the bit position of the most significan bit (MSB) to the bit position of the least significan bit (LSB) depending on the generation number. Since search area gradually becomes narrow in the optimization using the LVBSP, though the optimization progress is slow, the optimization intends to obtain a good solution. Another one is a cyclically varying VBSP (CVBSP)[9]. In the CVBSP, the BSP is defined by a window function. The position of the window function cyclically moves between the bit position of the MSB and the bit position of the LSB depending on the generation number. Since search area cyclically changes wide and narrow repeatedly in the optimization using the CVBSP, the optimization progress intends to be fast in the earlier generations. In this paper, we propose to combine these two types of the VBSP. We call this new VBSP a mixed VBSP (MVBSP). The MVBSP will be expected to absorb effectiveness of those two VBSPs.

2. FUZZY MODELING USING THE GA

In this paper, the simplified fuzzy reasoning is adopted. One of rules in the simplified fuzzy reasoning is represented by the following expression:

$$rule\ \langle i_1 i_2 ... i_N \rangle : if\ x_1\ is\ A_{1i_1}\ and, ..., \ and\ x_N\ is\ A_{Ni_N}\ then\ y\ is\ w_{\langle i_1 i_2 ... i_N \rangle}. \quad (1)$$

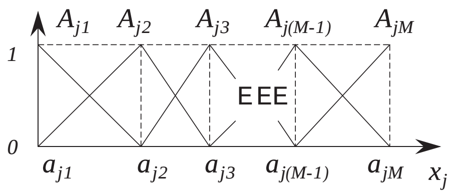


Figure 1: Antecedent membership functions.

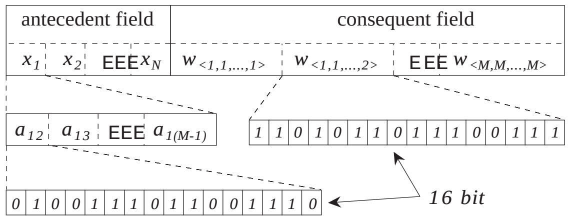


Figure 2: Genotype coding.

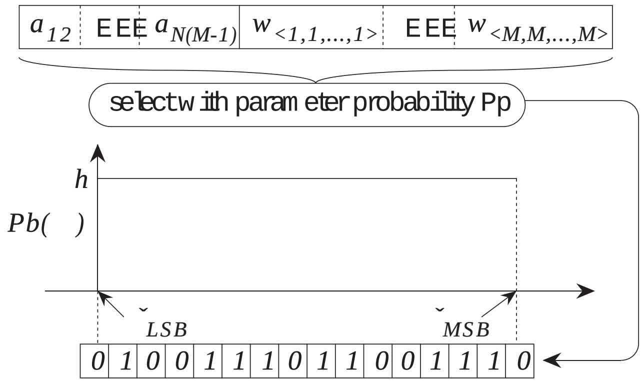


Figure 3: Conventional mutation operator.

where $x_1, ..., x_N$ denotes input variables, $A_{1i_1}, ..., A_{Ni_N}$ denote MSFs defined for their input variables, y denotes a output variable or reasoning result and $w_{\langle i_1 i_2 ... i_N \rangle}$ denotes a consequent crisp value of the rule $\langle i_1 i_2 ... i_N \rangle$. The identifier of the rule $\langle i_1 i_2 ... i_N \rangle$ shows that an MSF A_{ji_j} is defined for an input variable x_j in this rule. The MSF of the antecedent part is defined by a triangular function as shown in Fig.1. The reasoning result, y, are defined using a compatibility of the antecedent part, $\mu_{\langle i_1 i_2 ... i_N \rangle}$, by the following equation:

$$y = \sum_{\langle i_1 i_2 ... i_N \rangle} \mu_{\langle i_1 i_2 ... i_N \rangle} \cdot w_{\langle i_1 i_2 ... i_N \rangle} \Bigg/ \sum_{\langle i_1 i_2 ... i_N \rangle} \mu_{\langle i_1 i_2 ... i_N \rangle} \tag{2}$$

where the compatibility of the rule $\langle i_1 i_2 ... i_N \rangle$, $\mu_{\langle i_1 i_2 ... i_N \rangle}$ is defined by multiplication of the values of MSF, $\prod_{j=1}^{N} A_{ji_j}(x_j)$.

When the fuzzy model is tuned or optimized by using the GA, the fuzzy reasoning is required to be coded into genotype as an individual. In this paper, the fuzzy reasoning is coded into genotype as shown in Fig.2. Those parameters configuring the individual denote positions of antecedent MSF shown in Fig.1 and consequent crisp values.

In the GA, the initial set of the individuals or a initial population is generated by using random number. Performing each individual, parent individuals are selected. Applying the following two genetic operators, a crossover and mutation operators, to these parent individuals, new population is generated at each generation.

The crossover operator is implemented as follows. Consider two individuals applied to the crossover operator. Two boundaries between the parameters each other are selected in the antecedent and the consequent fields individually. The series of parameters divided by those crossover boundaries are substituted each other. Then new two individuals are obtained.

The mutation operator is implemented as shown in Fig.3 where the horizontal axis, ξ, denotes the bit position of the parameter, ρ_{LSB} and ρ_{MSB} denote the bit positions of the least significant bit (LSB) and most significant bit (MSB) respectively and h denotes the height of the function $Pb(\xi)$ which gives the value of the BSP. Some of the parameters are selected with a value of parameter-selection probability, Pp. The value of the bit of these selected parameters are inverted with the value of the BSP. The BSP in the conventional GA is expressed by a constant function as shown in Fig.3.

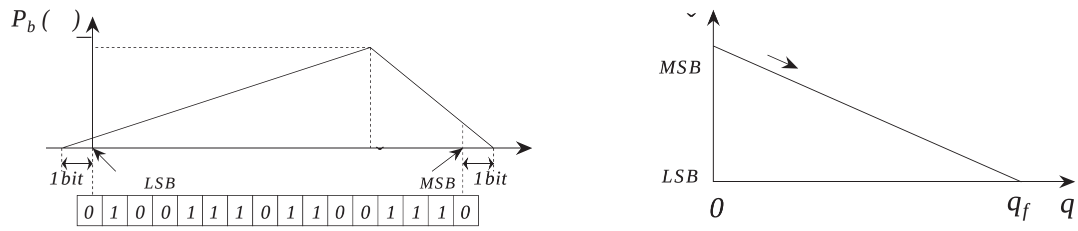
3. LOCAL SEARCH BY THE VBSP

When the conventional fixed BSP shown in the previous section is applied to the GA, state transition of the GA is always global. Since this allows the search of the GA to be global, the fixed BSP has brought about possibility to obtain a global optimum. However, huge times to perform the fuzzy model is required to converge on such a global optimum in the case of the fuzzy modeling, because of a lot of number of parameters to be optimized. When the fuzzy model is applied to a control system, such a global optimal fuzzy model may not be required. There are many cases that a local optimal fuzzy model can be satisfied to apply to the control system. Along this line, we present LVBSP and CVBSP as procedures to execute the state transition of the GA locally.

3.1 LINEAR VARIABLE BIT-SELECTION PROBABILITY

The LVBSP[8] is defined as shown in the Figs.4 (a) and (b) where η and ρ denote the height and the top position of the triangular function respectively, q denotes the number of generation, q_f denotes the number of the final generation and ξ_{LSB} and ξ_{MSB} denote the bit positions of the LSB and the MSB respectively. The shape of the LVBSP is defined by the triangular function whose top position is gradually moved by the linear function.

The LVBSP serves wide search area in the opening stage of the optimization when the optimization would give inferior performance and narrow search area in the final stage when the optimization would give good performance. And, since the LVBSP covers all the bit positions, the LVBSP can provide not only local search but also effect on global search.



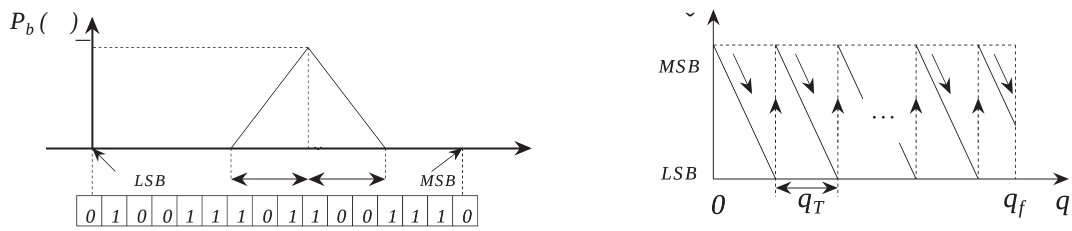
(a) Shape of the LVBSP. (b) Top position of the VBSP, ρ.

Figure 4: Definition of the LVBSP.

3.2 CYCLIC VARIABLE BIT-SELECTION PROBABILITY

The CVBSP[9] is defined as shown in the Figs.5 (a) and (b) where ω denotes the width of the triangular window. The shape of the LVBSP is defined by the triangular window whose top position cyclically moves depending on the generation number as shown in Fig.5 (b).

The CVBSP repeatedly varies the range of the search area from wide range to narrow range through the optimization. Since the CVBSP covers a part of the bit positions, it serves strong local search in the GA rather than the LVBSP. Therefor, the optimization using the CVBSP is expected to quickly converge to a local minimum.



(a) Shape of the CVBSP. (b) Top position of the CVBSP, ρ.

Figure 5: Definition of the CVBSP.

3.3 MIXED VARIABLE BIT-SELECTION PROBABILITY

Although the optimization using the LVBSP serves better results than that using the CVBSP after the optimization, that using the CVBSP quickly converges on a good local optimum rather than that using LVBSP. To provide the GA with the features both of the LVBSP and the CVBSP, this paper proposes a mutation operator which combines the LVBSP and the CVBSP. To combine these VBSPs, the population is configured by two sets of the individual, set A and set B, as shown in Fig.6. In the mutation operator, the LVBSP is applied to the set A while the CVBSP is applied to the set B. We call this method MVBSP.

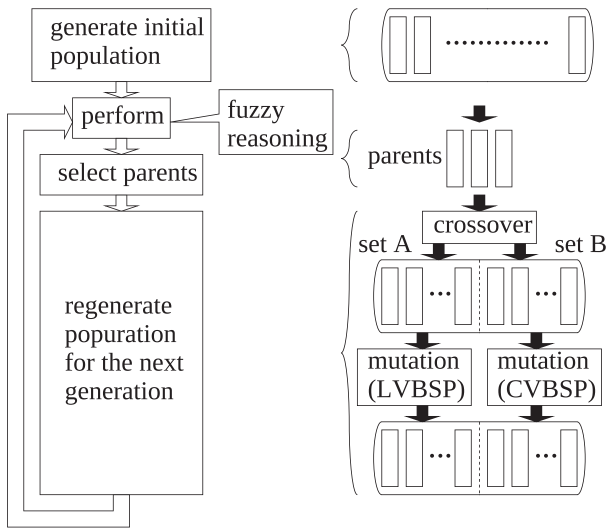


Figure 6: Procedures and configuration of the population in the GA

4. NUMERICAL EXAMPLE

A numerical example is examined to investigate effectiveness of the VBSP. In the numerical example, the fuzzy reasoning possess two input variables, x_1 and x_2, and one output variable y. These two input variables vary in a semi-open interval, $x_1, x_2 \in [0, 1)$. An input domain D_0 is divided into two domains, D_1 and D_2, to define an elite and two semi-elites. In these domains, the individuals are performed by a squared error given by the following equation:

$$e_k(g) = \sum_{x_1, x_2 \in D_k} (y_o - y_f)^2 \Big/ |D_k| \tag{3}$$

where k denotes the number of those domains, g denotes an individual to be performed, $|D_k|$ denotes the number of data to be performed, y_o denotes the objective result and y_f denotes the fuzzy reasoning result. The elite, E, has contents of an individual which gives the smallest value of the squared error, e_0, in the domain D_0 through the optimization. The semi-elites, S_1 and S_2, have contents of individuals which give the smallest value of the functions, e_1 and e_2, in the domains D_1 and D_2, at the current generation respectively.

A population in this numerical example consists of 18 individuals. Each individual in genotype is coded by the same manner shown in the previous section.

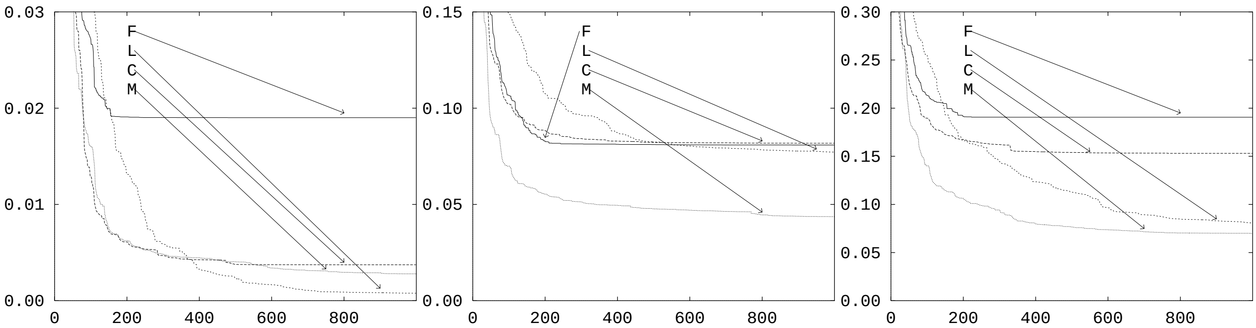
The following three functions are adopted to be approximated by the fuzzy reasoning model.

$$y = f_1(x_1, x_2) = 0.5x_1 + 1.2x_2, \tag{4}$$

$$y = f_2(x_1, x_2) = sin(3\pi x_1) - (x_2 - 0.5)^2, \tag{5}$$

and

$$y = f_3(x_1, x_2) = \gamma_1(x_1) + \gamma_2(x_2) \tag{6}$$



(a)results for Eq.(4) (b)results for Eq.(5) (c)results for Eq.(6)

Figure 7: Transition of averaged value of squared error for function approximation.

where

$$\gamma_1(x_1) = \begin{cases} -4x_1 + 1 & (0 \le x_1 < 0.5), \\ 0 & (0.5 \le x_1 < 0.75), \\ -4(x_1 - 0.75) + 1 & (0.75 \le x_1 < 1), \end{cases} \qquad \gamma_2(x_2) = \begin{cases} -1 & (0 \le x_2 < 0.5), \\ 1 & (0.5 \le x_2 < 0.75), \\ -8(x_2 - 0.75) + 1 & (0.75 \le x_2 < 1). \end{cases}$$

Figures 7 (a)–(c) illustrate optimization progresses in the examples of function approximation where curves pointed by F, L, C and M denote the results using the convention fixed BSP, the LVBSP, the CVBSP and the MVBSP respectively. In order to consider that the GA behaves as a stochastic model, ten times of the optimizations under the same condition have been executed. Each curves in Fig.7 shows averaged value of the squared error for the ten optimizations. Comparing the optimizations using the conventional fixed BSP with those using the LVBSP and the CVBSP, the features of these VBSPs are represented. The optimization using the LVBSP gradually decreases the squared errors and finally gives better results than those using the conventional fixed BSP and the CVBSP. The optimization using the CVBSP quickly decreases the squared errors rather than those using the conventional fixed BSP and the LVBSP. On the other hand, the optimization using the MVBSP quickly decreases the squared errors in the opening stage of the optimization and gradually decreases the squared errors in the middle and final stages. Besides, the optimizations using the MVBSP have obtained the best results among them for each function approximation.

5. CONCLUSION

In this paper, an efficient method to serve local search in the mutation operator of the GA for the fuzzy modeling has been proposed. Combining the LVBSP and the CVBSP, the efficient method, the MVBSP, has been proposed. By means of the MVBSP, the optimization can quickly obtain a good fuzzy model in the opening stage of the optimization and can gradually improve the performance in the middle and the final stages. To illustrate the efficiency of the MVBSP, examples of the function approximation are executed as generalized problems to optimize the fuzzy reasoning model.

REFERENCES

1. SUGENO M. 1988, Fuzzy Control, Nikkan Kogyo Shinbunsha (in Japanese).
2. GOLDBERG D.E., Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, New York, 1989.
3. SAKAWA M. & TANAKA M., Genetic Algorithm, Asakura-Shoten, Tokyo, 1995 (in Japanese).
4. Katayama R., Kajitani Y. and Nishida Y., A Self Generating And Tuning Method for Fuzzy Modeling Using Interior Penalty Method And Its Application to Knowledge Acquisition of Fuzzy Controller, Fuzzy Control Systems, CRC Press, chapt.9, pp.197-224, 1993.
5. Nomura H., Hayashi I. and Wakami N., A Self-Tuning Method of Fuzzy Reasoning By Genetic Algorithm, Fuzzy Control Systems, CRC Press, chapt.16, pp.337-354, 1993.

6. OHKI M., MORIYAMA T., MIYATA H. & OHKITA M., Self-Tuning of Fuzzy Reasoning with Genetic Algorithm, Proc. of Second World Congress of Nonlinear Analysts (WCNA'96) (Athens, Greece, July 10-17, 1996), part 8, pp.5291-5301, 1997.

7. OHKI M., SHIKATA T, MORIYAMA T. & OHKITA M., A Genetic Algorithm with Modified Bit-Selection Probability for Optimizing the Fuzzy Reasoning, Proc. of 5th European Congress on Intelligent Techniques and Soft Computing, (EUFIT'97), (Aachen, Germany Sept. 8-11, 1997), vol.1, pp.694-698, 1997.

8. OHKI M., MORIYAMA T. & OHKITA M., Optimization of Fuzzy Reasoning by Using Genetic Algorithm with Variable Bit-Selection Probability in its Mutation Operator, Trans. of IEE Japan, vol.119-C, no.7/8, pp.1196-1202, 1998 (in Japanese).

9. OHKI M., HASEGAWA S, MORIYAMA T. & OHKITA M., Genetic Algorithm with Cyclic Variable Bit-Selection Probability Varying Depending on Generation, Proc. of 6th European Congress on Intelligent Techniques and Soft Computing, (EUFIT'98), (Aachen, Germany Sept. 7-10, 1998), vol.1, pp.477-481, 1998.