Design of ANFIS based Estimation and Control for MIMO Systems

R. Sivakumar*, C. Sahana*, P. A. Savitha *
*(Department of Electronics and Instrumentation, St. Joseph's College of Engineering, Anna University, Chennai – 119, India)

ABSTRACT

This work is an attempt to illustrate the usage and effectiveness of soft computing techniques in the estimation and control of multi input and multi output systems. This paper focuses on neuro-fuzzy system ANFIS (Adaptive Neuro Fuzzy Inference system). An Adaptive Network based Fuzzy Interference System architecture extended to cope with multivariable systems has been used. The performance of the control strategy is studied on the control of Distillation Column problem. The results confirmed the control quality improvement with MPC and PID controller.

Keywords **- ANFIS, Distillation Column, PID controller, MPC, Neural modeling.**

1.INTRODUCTION

The controller always aims at the set point value of the given process variable. This is the main task of the properly designed controller. The controller should also provide some flexibility in case of change in set point and disturbances. Today there are many methods for designing intelligent controllers, such as predictive controller, fuzzy control, neural networks and expert systems. Various combinations of these controllers give a number of design possibilities. Artificial Neural Networks (ANNs) and Fuzzy Logic (FL) have been increasingly in use in many engineering fields since their introduction as mathematical aids [1-2]. Being branches of Artificial Intelligence (AI), both emulate the human way of using past experiences, adapting itself accordingly and generalizing.

Obtaining a mathematical model for a system can be rather complex and time consuming as it often requires some assumptions such as defining an operating point and doing linearization about that point and ignoring some system parameters, etc. This fact has recently led the researchers to exploit the neural and fuzzy techniques in modelling complex systems utilizing solely the input-output data sets. Although fuzzy logic allows one to model a system using human knowledge and experience with IF- THEN rules, it is not always adequate on its own. This is also true for ANNs, which only deal with numbers rather than linguistic expressions. This deficiency can be overcome by combining the superior features of the two methods. To achieve the most accurate set point, an appropriate extensions and improvements in the intelligent control is needed. This paper uses a control strategy that enhances a fuzzy controller with a self-learning capability for achieving prescribed control objectives. In this sense, an extended Adaptive-Network based Fuzzy Inference System (ANFIS) architecture is employed [3], so that a fuzzy inference system is built for achieving a desired input/output mapping. The learning method used allows the tuning of parameters both of the membership functions and the consequents in a Sugeno–type inference system.

This paper is organized as follows. The second section presents basic information of PID controller. The third section introduces the classic MPC, Dynamic Matrix Control and the distillation column benchmark problem. The following two sections present ANFIS controller and estimator design. Results and discussion are presented in the sixth section. The concluding remarks are given at the end of this paper.

2. PID CONTROLLER

A Proportional–Integral–Derivative (PID) controller is a generic control loop feedback mechanism widely used in industrial control systems. A PID is the most commonly used feedback controller. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs. The PID controller calculation (algorithm) involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. These values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, or the power supplied to a heating element.

Tuning a control loop is the adjustment of its control parameters (proportional band/gain, integral gain/reset, derivative gain/rate) to the optimum values for the desired control response. Stability (bounded oscillation) is a basic requirement, but beyond that, different systems have different behavior, different applications have different requirements, and requirements may conflict with one another.

3. MODEL PREDICTIVE CONTROL

3.1. Classical MPC

The basic block of MPC is shown in Fig 1. The main idea behind MPC-type controllers is illustrated in Fig 2 for a SISO system. At sampling time *k*, a set of *m* future manipulated variable moves (control horizon) are selected, so that the predicted response over a finite horizon *p* (prediction horizon) has certain desirable characteristics. This is achieved by minimizing an objective function based on the deviation of the future controlled variables from a desired trajectory over the prediction horizon *p* and the control energy over the control horizon *m*. The MPC optimization is performed for a sequence of hypothetical future control moves over the control horizon and only the first move is implemented [4]. The problem is solved again at time *k + 1* with the measured output y as the new starting point. Model uncertainty and unmeasured process disturbances are handled by calculating an additive disturbance as the difference between the process measurement and the model prediction at current time step.

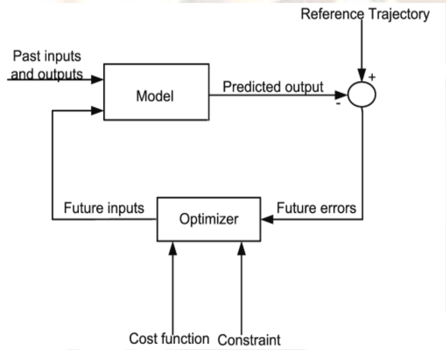

Figure 1 Basic Block of MPC

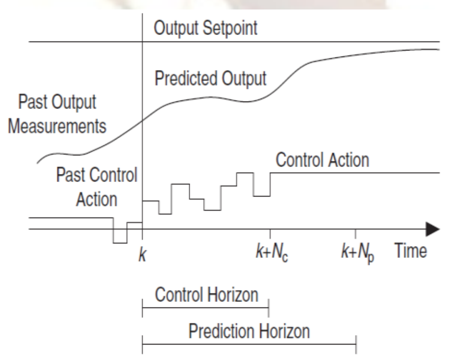

Figure 2 Schematic view of MPC

3.2. Dynamic Matrix Control

The outputs are evaluated at the discrete-time steps; it makes sense to use discrete models for output prediction. Thus step and impulse response models both of which are used in common MPC algorithms.

DMC is based on step response model which is written as:

$$y_k = \sum_{i=1}^{N-1} s_i \Delta u_{k-1} + s_N u_{k-N} \qquad (1)$$

where y_k is the model prediction at time step k, and u_{k-N} is the manipulated input N steps in the past. The difference between the measured output (y_k) and the model prediction is called the additive disturbance. Steps involved in implementing DMC on a process:

- Develop a discrete step response model with length (N) based on sample time (Δt).
- Specify the prediction (P) and control (M) horizons.
- N ≥ P ≥ M.
- Specify the weight (W) on the control action (W = 0 if no weighting).
- All calculations assume deviation variable form, so it is important to convert to/from physical units.

3.3. Distillation Column

The distillation column is probably the most popular and important process studied in the chemical engineering literature. Distillation is used in many chemical processes for separating feed streams and for the purification of final and intermediate product streams. Most column handle multi component feeds, but many can be approximated by binary or pseudo binary mixtures [5]. The objective is to split a liquid two component mixture into its fractions throughout stripping and rectifying processes.

The first 2 x 2 MIMO process is presented by Wood and Berry (1973). The study was performed on a 9 inch diameter, 8 tray column equipped with a total condenser and a basket type reboiler. The required control action for the manipulative variables in the composition loops, reflux and steam flow, were cascaded to the set points of the appropriate flow controllers. The transfer function characterizing the column dynamics were established by pulse testing. Parameters of the assumed first order plus time delay transfer function were determined from the transient data. The time delays were established and the gains and time constants are determined by least squares fit employing Rosenbrok's direct search technique. The process transfer function matrix of the distillation process is given by

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \dfrac{12e^{-s}}{16.7s+1} & \dfrac{-18.6e^{-3s}}{21.0s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.8e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad (2)$$

4. DESIGN OF ANFIS CONTROLLER

The basic idea behind the design of neuro-adaptive learning techniques is very simple. These techniques provide a method for the fuzzy modeling procedure to learn information about data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input-output data [6]. ANFIS constructs an input-output mapping based on both human knowledge (in the form of fuzzy if-then rules) and simulated input-output data pairs. It serves as a basis for building the set of fuzzy if-then rules with appropriate membership functions to generate the input output pairs. The parameters associated with the membership functions are open to change through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector, which provides a measure of how well the ANFIS is modeling the input-output data for a given parameter set. Once the gradient vector is obtained, back propagation or hybrid learning algorithm can be applied in order to adjust the parameters.

ANFIS can be used in modeling, estimating and controlling studies in chemical engineering processes similar to other artificial intelligence methods such as NNs and Fuzzy Logic (FL). In this paper, the designed ANFIS is utilized as an estimator and a controller. Estimation is done for compositions of top and bottom products, whereas adaptive control strategy that needs no separate process network model with ANFIS controller is utilized in a control system for set point tracking problem and disturbance rejection problem.

The ANFIS architecture of the first-order Takagi-Sugeno inference system is shown in Fig 3. The entire system consists of five layers and the relationship between the input and output of each layer is summarized as follows:

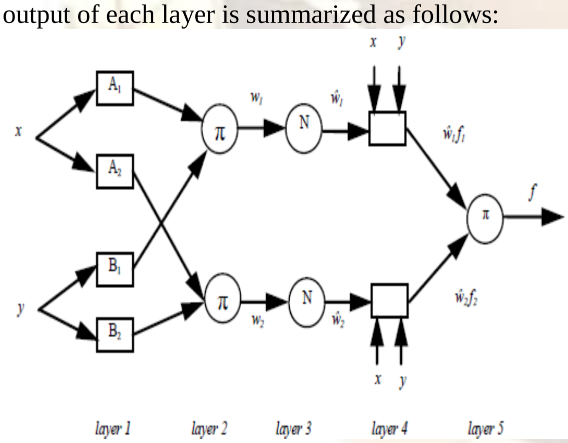

Figure 3 Structure of ANFIS

Layer 1: Every node i in this layer is an adaptive node with a node output defined by

$$o_{1,f} = \mu_A(x) \text{ for i=1,2 or} \quad (3)$$

$$o_{1,i} = \mu_{B_{i-2}}(y) \text{ for i=3,4} \quad (4)$$

where x or y is the input to the node, A_i or B_{i-2} is a fuzzy set associated with this node, characterized by the shape of the membership function in this node. Parameters in this layer are referred to as premise (antecedent) parameters.

Layer 2: Every node in this layer is a fixed node labeled π, which multiplies the incoming signals and output product.

$$o_{2,i} = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y) \text{ for i=1, 2} \quad (5)$$

Each output node represents the firing strength of a rule.

Layer 3: Every node of this layer is a circular node labeled N, the i[th] node calculates the ratio of the i[th] rule's firing strength to the sum of all rule's firing strengths.

$$o_{3,i} = \hat{w}_i = \frac{w_i}{w_1 + w_2} \text{ for i=1, 2} \quad (6)$$

output of this layer is called as normalized firing strengths.

Layer 4: Node i in this layer compute the contribution of the i[th] rule towards the model output with the following node function.

$$o_{4,i} = \hat{w}_i f_i = \hat{w}_i (p_i x + q_i y + r_i) \quad (7)$$

where $\hat{w}_i$ is the output of layer 3 and (p_i, q_i, r_i) is the parameter set. Parameters in this layer are referred to as consequent parameters.

Layer 5: The single node in this layer is a fixed node labeled Σ that computes the overall output as the summation of all incoming signals.

$$\text{Overall output}= o_5 = \sum_i w_i f_i \quad (8)$$

5. ANFIS AS AN ESTIMATOR

ANFIS can be used for the estimation of some dependent variables in chemical process. The designed ANFIS estimator is used to infer the compositions of top and bottom products. In estimator design process, different ANFIS structure are constructed and trained to find the architecture that gives the best performance as an estimator. As a first step to design an estimator, training data sets should be generated to train the estimator networks. These data sets consist of estimator inputs and desired output values. They are produced from the process input output data. Since, ANFIS is a data processing method, it is important that the input-output data must be within the sufficient operational range including the maximum and minimum values for both input and output variables of the system. If this is not provided, estimator performance cannot be guaranteed and thus the designed estimator will not be accurate. Having generated the training data, estimators that have different architectures are trained with the obtained data sets. Performances of the trained estimators are evaluated with model simulations and best estimator architecture is obtained. These simulations are made to verify and to generalize the ANFIS structures. Verification is done to show how good the estimator structure learned the given training data. This is carried out by simulating the column models with specific initial process inputs used in obtaining training data sets.

ANFIS estimator design consists of two parts: constructing and training. In constructing part, structure parameters are determined. These are type and number of input Membership Functions (MFs), and type of output MF. Any of several MFs such as Triangular, Trapezoidal and Gaussian can be used as an input MF. Frequently used MFs in literature are the Triangular and Gaussian. For this reason, they are chosen as input MF type in this study. Various numbers of MFs on each input can be chosen to define the linguistic labels significantly. Effective partition of the input space is important and it can decrease the rule number and thus increase the speed in both learning and application phase. Output MFs can be either a constant or in linear form. Both of these two forms are used for the output MF in this study. Having described the number and type of input MFs, the estimator rule base is constituted. Since, there is no standard method to utilize the expert knowledge; automatic rule generation (grid partition) method is usually preferred. According to this method, for instance, an ANFIS model with two inputs and three MFs on each input would result in 32 = 9 Takagi-Sugeno fuzzy if-then rules automatically. Although this method can require much computational knowledge especially in systems that have to be defined with many inputs, it is used in this study due to advantage of MATLAB software. Therefore, rule bases of the estimators are formed automatically with the number of inputs and number of MFs. After the ANFIS structure is constructed, learning algorithm and training parameters are chosen. The hybrid learning algorithm is used in this study. Parameters in the algorithm are epoch size (presentation of the entire data set), error tolerance, initial step size, step size decrease rate, and step size increase rate. Since there is no exact method in literature to find the optimum of these parameters a trial and error procedure is used.

MATLAB fuzzy logic toolbox is used to design ANFIS estimators' structures. Using the given training data set, the toolbox constructs an ANFIS structure using either a back propagation algorithm alone, or in combination with least squares type of method (hybrid algorithm). ANFIS model can be generated either from the command line, or through the ANFIS editor GUI. In this study, ANFIS Editor GUI is used to generate the ANFIS models with the chosen design parameters in construction phase. Written MATLAB code is used to train the ANFIS structure in the training step. The steps in ANFIS estimator design in this study utilizing the MATLAB fuzzy logic toolbox are as follows:

1. Generated training data is loaded to ANFIS Editor GUI.
2. Design parameters, number of input MF, type of input and output MF, are chosen. Thus, initial ANFIS structure is formed.
3. The code for the training is run with the initial structure.
4. ANFIS structure constituted after training is saved to use as an estimator.

If the operating input-output data are outside their training data range, estimator will not operate accurately. As a result, the training data set should possess sufficient operational range including the maximum and minimum values for input output variables. The data set should include data for each process variable, evenly distributed throughout the range for which estimation is desired. The maximum and minimum values of top and bottom products were determined by looking at the closed-loop responses of the system. Thus, model simulations are done to obtain the input-output data by using these values.

Estimator structure design and training are realized using MATLAB software. First, generated training data is loaded using the GUI Editor. Then, with chosen design parameters, initial estimator structure is constructed. Trainings of the structures are done by MATLAB. All structures are trained in the same way only by changing the training data.

The chosen MIMO system is trained with different membership functions. Fig 4, Fig 5 illustrates the ANFIS estimated output for top and bottom product respectively. Here 5 input Triangle membership functions and 100 epochs are used.

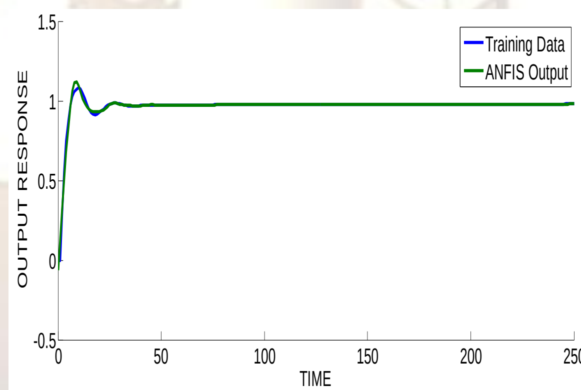

Figure 4 ANFIS estimation of top product with triangle membership function

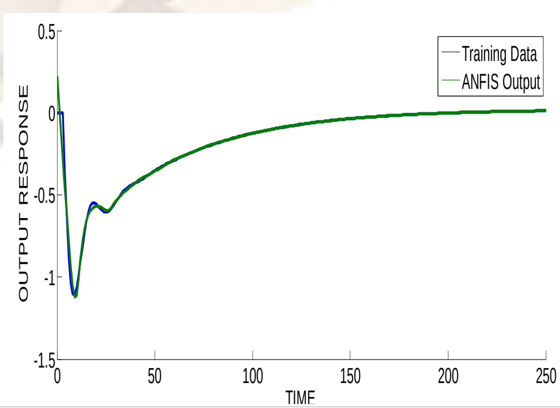

Figure 5 ANFIS estimation of bottom product with triangle membership function

Fig 6, fig 7 shows the ANFIS estimated output for top and bottom product with 5 inputs Gaussian type member-ship function and 100 epochs. From the results the latter combination of estimated data gives better result. This can be used as estimator for chosen MIMO system.

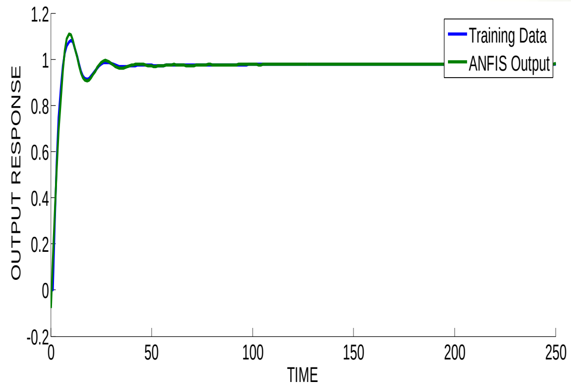

Figure 6 ANFIS estimation of top product with Gaussian membership function

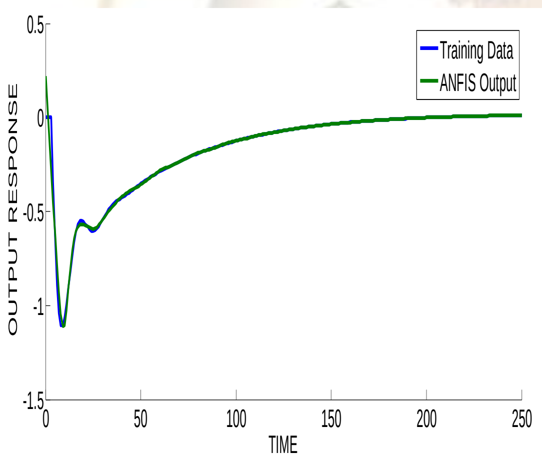

Figure 7 ANFIS estimation of bottom product with Gaussian membership function

6. ANFIS CONTROL FOR MIMO SYSTEMS

The best fit of ANFIS estimator data for the chosen MIMO is used for control study. The ANFIS controller is designed using MATLAB/SIMULINK software. The error and rate of change of error is used for rule formation in ANFIS controller. The Gaussian type input membership function is used for control design. Fig 8, fig 9 presents the ANFIS controller output.

The ANFIS controller response is compared with conventional multi loop PID controller tuned by BLT method and Model Predictive controller (MPC) tuned by Dynamic Matrix Control. From the fig 10, fig 11 the response of ANFIS controller is better than other two controllers. The performance indices ISE and IAE of all three controllers are listed in Table 1 and Table 2.

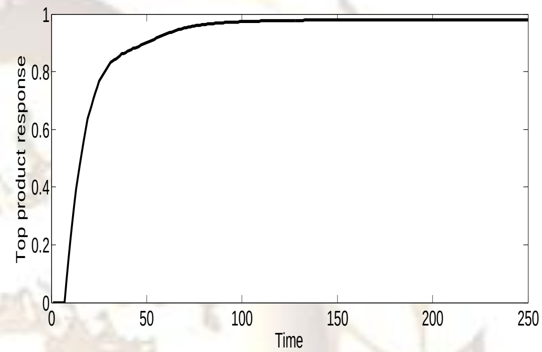

Figure 8 ANFIS controller response for top product

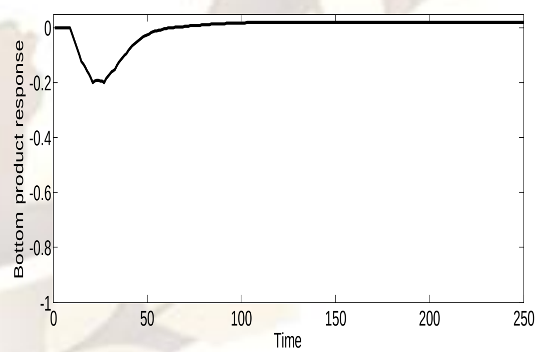

Figure 9 ANFIS controller response for bottom product

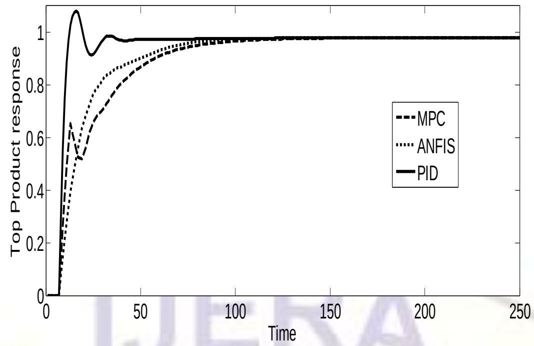

Figure 10 Comparison of PID, MPC and ANFIS controller of top product

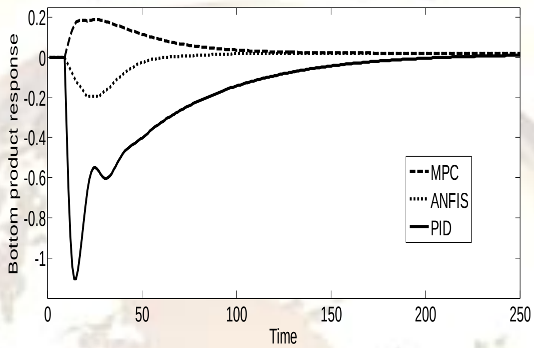

Figure 11 Comparison of PID, MPC and ANFIS controller of bottom product

Table 1 ISE, IAE comparison for top product

Performance	PID	MPC	ANFIS
ISE	11.74	9.33	2.791
IAE	3.427	1.064	0.5936

Table 21 ISE, IAE comparison for bottom product

Performance	PID	MPC	ANFIS
ISE	22.45	17.87	9.761
IAE	47.91	32.79	21.57

7.CONCLUSION

The performance of the control strategy is studied on the control of distillation column problem. The results confirmed the control quality improvement with MPC and multi-loop PID controller. Thus the optimization of the output is obtained and the performance criteria are compared with the ISE and IAE values.

REFERENCES

[1] E. H. Mamdani, S. Assilian, An Experiment in Linguistic Synthesis with a Logic Controller, *International Journal of Man-Machine Studies,8,* 1975, 1-13.

[2] S. Takagi, M. Sugeno, Fuzzy Identification of Fuzzy Systems and its Application to Modeling and Control, *IEEE Transaction on System, Man and Cybernetics*, *15*, 1985, 116-132.

[3] R. Jang, ANFIS: Adaptive Network based Fuzzy Inference System, *IEEE Transaction on System, Man and Cybernetics*, *23(3)*, 1993, 665-684.

[4] P. Tatjewski, M. L. Czuk, Soft Computing in Model based Predictive Control, *International Journal of Applied Maths and Computer Science*, *16(1)*, 2006, 7-26.

[5] S. Skogestad, Dynamics and Control of Distillation Column: A Tutorial Introduction, *Transaction of Institution of Chemical Engineers*, *75(A)*, 1997, 539-562.

[6] J. Fernandez de Canete, An Adaptive Neuro-Fuzzy Approach to Control a Distillation Column, *International Journal of Neural Computing and Applications*, 2000, 211-217.

[7] R. Sivakumar, K. Suresh Manic, V. Nerthiga, R. Akila, K. Balu, Application of Fuzzy Model Predictive Control in Multivariable Control of Distillation Column, *International Journal of Chemical Engineering and Applications*, *1(1)*, 2010, 39-42.

[8] R. Sivakumar, K. Balu, ANFIS based Distillation Column Control, *International Journal of Computer Applications Special issue on Evolutionary Computation*, *(2)*, 2010, 67-73.