Available online at www.sciencedirect.com

Journal of Sound and Vibration 297 (2006) 382–390

JOURNAL OF SOUND AND VIBRATION

www.elsevier.com/locate/jsvi

# Reinforcement genetic approach to coefficient estimation for multivariable nonlinear discrete-time dynamical systems

Wei-Der Chang*, Jun-Juh Yan

*Department of Computer and Communication, Shu-Te University Kaohsiung 824, Taiwan*

Received 9 May 2005; received in revised form 29 March 2006; accepted 10 April 2006
Available online 12 June 2006

## Abstract

In this paper, we propose a novel genetic algorithm (GA) with a multi-crossover fashion to estimate the associated coefficients for a class of nonlinear discrete-time multivariable dynamical systems. Unlike the traditional crossover method of using two chromosomes, the proposed method uses three chromosomes to achieve a crossover. According to the adjusting direction by crossing three chromosomes, more excellent offspring can be produced. To solve the identification problem of multivariable nonlinear discrete-time systems, each of estimated system coefficients represents a gene, and a collection of genes is referred to as a chromosome in the view of GA. The chromosomes in the population are then evolved using the proposed multi-crossover method. An illustrative example of multivariable nonlinear systems is given to demonstrate the effectiveness, as compared with the traditional crossover method, of the proposed method.

## 1. Introduction

It is rather important and essential to previously establish a mathematical representation for some real plant for the control system engineering. In accordance with this estimated mathematical model, the controller can be then designed by using a variety of control strategies of interest so that the prospective control purpose is achieved. Applying artificial intelligence techniques to systems modeling have been successively proposed in recent years such as based on using neural networks [1,2], fuzzy logic systems [3,4], or neuro-fuzzy systems [5,6]. In these publications, they put emphases on the system modeling that the architecture of the controlled plant is assumed to be unknown. In addition, if the plant structure is assumed to be known in advance, the remainder problem is how to accurately estimate the associated system coefficients under this kind of structure. Usually, the least-squares scheme is used for solving this. In Ref. [7], the least-squares method has been successfully used to estimate the coefficients in the static and dynamical systems respectively, and mathematical analyses and discussions have been developed. The recursive schemes are in essence local search techniques that search for the optimum by using gradient method. They often fail in the search for global optimum if the search space is not differentiable or linear in the coefficients [8].

---

*Corresponding author. Tel.: +886 76158000 x4808; fax: +886 76158000 x4899.
*E-mail address:* wdchang@mail.stu.edu.tw (W-D Chang).

0022-460X/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsv.2006.04.006

Based on simulating the natural evolutionary model, genetic algorithm (GA) that is one of artificial intelligence is a powerful tool to search for global solutions for an optimization problem over the search domain [9,10]. The algorithms provide a superior performance over the traditional optimal techniques, e.g., the steepest descent method. This is due to searching for solution from only one single direction on the search space [11]. However, GA method can be regarded as a search method from multiple directions, because they inherently have crossover and mutation operations during searching procedures. This implies that it is easier to escape from a local minimum.

In the conventional GA, all the variables of interest must first be encoded as binary digits (genes) and then forming a string (chromosome). It follows that three standard genetic operations, i.e., reproduction, crossover, and mutation are performed to produce a new generation. Such procedures are repeated until the pre-specified number of generations is achieved, or the required accuracy is satisfied. Based on this kind of GA, a great number of researches have been proposed for system coefficients estimation and control. In Ref. [8], they applied the binary-coding GA into estimating the locations of poles and zeros of a transfer function and used this estimated model to design a discrete time pole placement adaptive controller. Jiang and Wang [12] proposed a searching method for parameters estimation of nonlinear systems based on using the binary-coding GA. Furthermore, Yao and Sethares [13] demonstrated a modified binary GA for the nonlinear digital filters design and also for weights tuning in feedforward and recurrent neural networks. All of them, the evaluated parameters of filters were first encoded as binary alphabets in order to be suitably computed in the traditional way. After a series of manipulations, the resulting final binary alphabets are then returned as real numbers.

Another kind of genetic operation called the real-coded GA has been also introduced to a wide variety of real applications in recent years as stated in Refs. [14–17]. All genes in a chromosome used in real-coded GA are real numbers. It is more suitable to represent genes directly as real values for most of real optimization problems during genetic operations. Because exchange processes from a real value to a binary digit may suffer from the loss of precision relying on number of used bits. Expectably, this will be very complicated and difficult works if the numerical values are large and even have the decimal. For a real-coded GA the length of chromosomes becomes much shorter than one by using the binary-coded way. This implies that the computer programming for such algorithms can be easily run. On the basis of using this kind of real-coded GA and the proposed novel modified crossover, the key topic of this article is to mainly focus on coefficients estimation of nonlinear discrete-time multivariable dynamical systems. A number of algorithms about the subject of linear and nonlinear multivariable identifications such as subspace-based algorithms [18], RBF neural net model algorithms [19], and recursive relaxation computations [20] have been successively proposed. Besides, differently from the traditional crossover which is to use only two chromosomes in parents generation to cross by each other, the modified crossover approach proposed in this paper is based on the style of a multiple crossover, and this can provide a more precise adjusting direction for problem solutions and enhance the evolutionary convergence speed. All evaluated coefficients of nonlinear multivariable systems are directly referred to as genes and form a chromosome, which will be manipulated by using the multiple crossover GA.

## 2. Problem statement

To represent a class of nonlinear dynamical systems, two types of nonlinear models are generally available including a nonlinear polynomial expression and a nonlinear rational expression. Basically, a polynomial model is defined by linear in parameters and nonlinear in the regression terms, and can represent a wide range of linear and nonlinear systems. For a single-input and single-output nonlinear system, a general nonlinear discrete-time expression is mathematically defined by

$$y(k) = f\left\lfloor u(k-1),\ u(k-2), \ldots,\ u(k-N_u),\ y(k-1),\ y(k-2), \ldots,\ y\left(k-N_y\right)\right\rfloor + w(k) \tag{1}$$

where $u$ is an external input, $y$ is the output of the system, $w$ represents the measurement noise, $N_u$ and $N_y$ are the numbers of past inputs and outputs required, respectively. The nonlinear function $f(\cdot)$ generally contains past inputs and outputs of system, and can be expressed in terms of polynomials

$$f(k) = \sum_{j=1}^{num} \theta_j p_j(k), \tag{2}$$

where *num* represents number of regression terms of the nonlinear model, $p_j(k)$ is assumed to be known and is the product of past inputs and outputs such as $u(k-2)$, $u^2(k-1)y(k-1)$, or $y(k-1)y(k-2)$, and $\theta_j$ is referred to as the associated coefficient of the nonlinear model and will be evaluated. Expanding this kind of nonlinear discrete-time polynomial expression to multivariable architectures, we have

$$\begin{aligned} y_1(k) &= \sum_{j=1}^{num_1} \theta_{1j} p_{1j}(k) + w_1(k), \quad y_2(k) = \sum_{j=1}^{num_2} \theta_{2j} p_{2j}(k) + w_2(k), \ldots, \\ y_n(k) &= \sum_{j=1}^{num_n} \theta_{nj} p_{nj}(k) + w_n(k), \end{aligned} \tag{3}$$

where the case $n$ is the number of system outputs and $p_{ij}$, for $i \in \underline{n}$ and $\underline{n} = \{1, 2, \cdots, n\}$, is the products of different kinds of past inputs and outputs such as $u_1(k-2)u_2(k-1)$, $u_2^2(k-1)y_1(k-1)$, or $y_1(k-1)y_2(k-2)$. Moreover, it is assumed that all outputs $y_i$, for $i \in \underline{n}$, in Eq. (3) must be measurable and finite in each sampling step when the external inputs are applied.

For a given nonlinear multivariable systems as demonstrated in Eq. (3), the work of estimation is to evaluate the associated coefficients from a series of input and output observations. Suppose that an undetermined nonlinear discrete-time multivariable polynomial model which will match the actual plant of Eq. (3) is given by

$$\hat{y}_1(k) = \sum_{j=1}^{num_1} \hat{\theta}_{1j} \hat{p}_{1j}(k), \quad \hat{y}_2(k) = \sum_{j=1}^{num_2} \hat{\theta}_{2j} \hat{p}_{2j}(k), \ldots, \hat{y}_n(k) = \sum_{j=1}^{num_n} \hat{\theta}_{nj} \hat{p}_{nj}(k), \tag{4}$$

where $\hat{y}_i$, $\hat{\theta}_{ij}(k)$, and $\hat{p}_{ij}(k)$ are the outputs, the associated coefficients, and the product terms of past inputs and outputs of this estimated model, respectively, for $i \in \underline{n}$ and $j \in \{num_1, num_2, \ldots, num_n\}$. For simplification, let $\Theta = \left\lfloor \hat{\theta}_1,\ \hat{\theta}_2, \ldots,\ \hat{\theta}_m \right\rfloor = \left\lfloor \hat{\theta}_{11},\ \hat{\theta}_{12}, \ldots,\ \hat{\theta}_{1num_1},\ \hat{\theta}_{21},\ \hat{\theta}_{22}, \ldots,\ \hat{\theta}_{2num_2}, \ldots,\ \hat{\theta}_{n1},\ \hat{\theta}_{n2}, \ldots,\ \hat{\theta}_{nnum_n} \right\rfloor$ be a new undetermined coefficients vector and where $m = num_1 + num_2 + \ldots + num_n$ is the total number of the associated coefficients of multivariable systems. From the evolutionary point of view, $\Theta$ is also called a chromosome and all $\hat{\theta}_i$, for $i \in \underline{m}$, in $\Theta$ stand for genes.

## 3. A modified multiple crossover computation

A proper performance index or an objective function should be first defined for estimation problems. Usually, the GA method only requires computing the objective function to guide its search. There is no requirement for derivatives that are often used in solving for the traditional optimization problems. To derive the coefficients estimation algorithm, the objective function $J$ to be minimized is the total sum of squared errors between the actual outputs in Eq. (3) and estimated outputs in Eq. (4) over a period of T discrete-time intervals as follows:

$$\begin{aligned} J &= \sum_{k=1}^{T} \left\{ [y_1(k) - \hat{y}_1(k)]^2 + [y_2(k) - \hat{y}_2(k)]^2 + \cdots + [y_n(k) - \hat{y}_n(k)]^2 \right\} \\ &= \sum_{k=1}^{T} \left[ e_1^2(k) + e_2^2(k) + \cdots + e_n^2(k) \right] = \sum_{k=1}^{T} \sum_{i=1}^{n} e_i^2(k). \end{aligned} \tag{5}$$

The task of estimation is to determine a set of proper model coefficients in Eq. (4) so that the objective function $J$ in Eq. (5) is minimized. Under the appearances of measurable noises, Fig. 1 represents the block

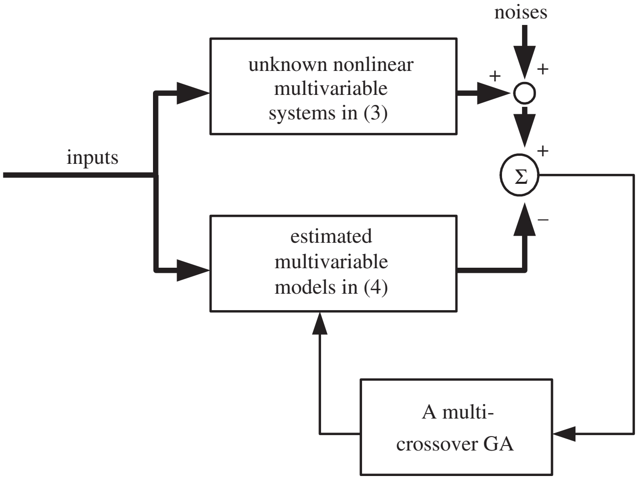

Fig. 1. Block diagram of multivariable systems coefficient estimation using the multi-crossover GA.

diagrams for this kind of coefficients estimation based on using the proposed modified multi-crossover real-coded GA that will be explained in the following.

### 3.1. Search space for $\Theta$

The genetic operations start with a given population including some chromosomes that are randomly generated. Every chromosome in the population represents a set of possible solution to the optimization problem of coefficients estimation, and these chromosomes are then evolved to generate better offspring according to the objective function $J$ by applying three genetic operators. A search space for model coefficients $\Theta$ must be previously given as

$$\Omega_\Theta = \left\{ \Theta \in \Re^m \middle| \hat{\theta}_{1\,\min} \leqslant \hat{\theta}_1 \leqslant \hat{\theta}_{1\,\max},\ \ \hat{\theta}_{2\,\min} \leqslant \hat{\theta}_2 \leqslant \hat{\theta}_{2\,\max}, \ldots,\ \ \hat{\theta}_{m\ \min} \leqslant \hat{\theta}_m \leqslant \hat{\theta}_{m\ \max} \right\}. \tag{6}$$

All genes $\hat{\theta}_i$, for $i \in \underline{m}$, in a chromosome will be evolved within this constrained space $\Omega_\Theta$ during the genetic operations. Once a produced chromosome goes beyond $\Omega_\Theta$, then the original is retained. Moreover, let $N$ represent the number of chromosomes in the population, i.e., the size of population, and parameters $p_r$, $p_c$, and $p_m$ are referred to as probabilities of reproduction, crossover, and mutation, respectively.

### 3.2. Reproduction

There are two well-known selection mechanisms, roulette wheel and tournament selections, used for reproduction operation. The roulette wheel selection can be visualized by imagining a wheel where each chromosome occupies an area that is related to its value of objective function. When a spinning wheel stops, a fixed marker determines which chromosome will be selected to reproduce into the mating pool [15]. This kind of selection mechanism needs more numerical computations. For the tournament selection, however, it is quite simple and suitable for checking whether a chromosome can reproduce or not according to the corresponding objective function. The selection formula is that $p_r \times N$ chromosomes with minimum objective functions are more added into the population, and $p_r \times N$ chromosomes with maximum ones are then discarded from the population. This implies that the size of resulted population is the same as the original. After the selection, all chromosomes are completely put in the mating pool. The next step is to generate new offspring by applying the following proposed multiple crossover and mutation operations on chromosomes in the mating pool.

### 3.3. Crossover

#### 3.3.1. Traditional crossover

The fashion of the traditional crossover that uses only two chromosomes to cross is first introduced. Let $N$ chromosomes in the mating pool be randomly divided into $N/2$ pairs, which serve as parents and will be crossed by each other. Suppose that both $\Theta_1$ and $\Theta_2$ are selected and c is a random number chosen from [0, 1]. If $c \geqslant p_c$, then the following crossover operations for $\Theta_1$ and $\Theta_2$ are performed

$$\begin{aligned}
&\text{if } J(\Theta_1) < J(\Theta_2),\\
&\quad \Theta_1' = \Theta_1 + r(\Theta_1 - \Theta_2),\\
&\quad \Theta_2' = \Theta_2 + r(\Theta_1 - \Theta_2),\\
&\text{else}\\
&\quad \Theta_1' = \Theta_1 + r(\Theta_2 - \Theta_1),\\
&\quad \Theta_2' = \Theta_2 + r(\Theta_2 - \Theta_1),
\end{aligned} \tag{7}$$

where $J(\Theta_1)$ and $J(\Theta_2)$ represent corresponding objective functions due to chromosomes $\Theta_1$ and $\Theta_2$, respectively, and $r \in [0, 1]$ is a random number determining the crossover grade of these two. Figs. 2 and 3 schematically show the changes of both $\Theta_1$ and $\Theta_2$ vectors after running the crossover operation, respectively. If $c < p_c$, no crossover operation is performed.

#### 3.3.2. Proposed multiple crossover

Unlike the technique of traditional crossover using two chromosomes, a novel multiple crossover is proposed. Suppose that three chromosomes $\Theta_1$, $\Theta_2$, and $\Theta_3$ are randomly picked in the mating pool and

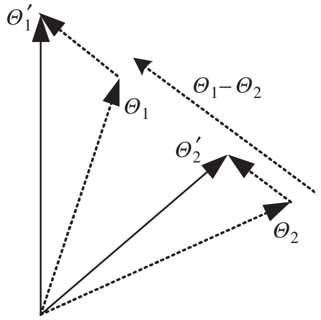

Fig. 2. $J(\Theta_1) < J(\Theta_2)$.

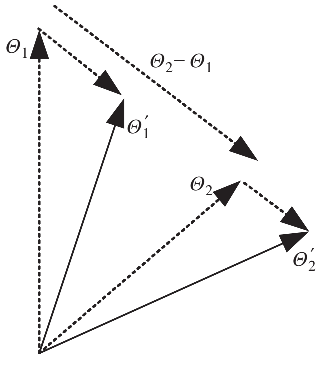

Fig. 3. $J(\Theta_1) > J(\Theta_2)$.

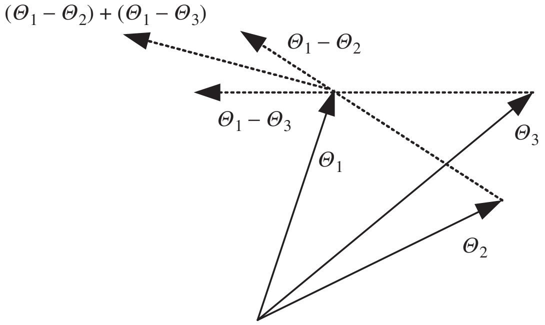

Fig. 4. A new adjusting direction by crossing three chromosomes.

crossed by one another, and that the objective function $J(\Theta_1)$ is the smallest among three. The diagram of the proposed multi-crossover model is schematically sketched in Fig. 4. It is easily seen from Fig. 4 that a modified adjusting direction of $(2\Theta_1-\Theta_2-\Theta_3)$ that combines $\Theta_1-\Theta_2$ and $\Theta_1-\Theta_3$ vectors is produced for updating $\Theta_1$, $\Theta_2$, and $\Theta_3$. All three chromosomes in such a way are changed simultaneously along this adjusting direction. Also, let $c$ be a random number selected from [0, 1]. If $c \geqslant p_c$, then the following multiple crossover formulas are performed

$$\begin{aligned}\Theta_1' &= \Theta_1 + r(2\Theta_1-\Theta_2-\Theta_3),\\ \Theta_2' &= \Theta_2 + r(2\Theta_1-\Theta_2-\Theta_3),\\ \Theta_3' &= \Theta_3 + r(2\Theta_1-\Theta_2-\Theta_3),\end{aligned} \tag{8}$$

where $r \in [0,1]$ is a random value determining the crossover grade of these three. If $c < p_c$, no crossover operation is performed.

### 3.4. Mutation

The mutation operation follows the multiple crossover and provides a possible mutation on some selected chromosomes $\Theta$. Only $p_m \times N$ random chromosomes in the current population are chosen to be mutated. The formula of mutation operation for a selected $\Theta$ is given by

$$\Theta' = \Theta + \sigma \times \Phi, \tag{9}$$

where $\sigma$ is a small positive constant and $\Phi \in \Re^m$ is a random perturbation vector to produce small disturbances on $\Theta$.

The procedure that has completely run above three genetic operators, i.e., reproduction, crossover, and mutation, is called a generation. The algorithms stop if the designed objective function is satisfied or the pre-specified number of generation $G$ is achieved. Notice again that if a generated chromosome during genetic operations is outside the search space $\Omega_\Theta$, then the original will be retained. The complete design steps for coefficients estimation of nonlinear discrete-time multivariable systems using a multi-crossover real-coded GA can be summarized as follows.

*Data*: An unknown nonlinear discrete-time multivariable systems in Eq. (3), sampling interval $T$ in Eq. (5), searching boundary parameters $\hat{\theta}_{1\,\min}$, $\hat{\theta}_{1\,\max}$, $\hat{\theta}_{2\,\min}$, $\hat{\theta}_{2\,\max},\ldots,\hat{\theta}_{m\,\min}$, and $\hat{\theta}_{m\,\max}$ in Eq. (6) for $\Omega_\Theta$, population size $N$, probabilities of reproduction $p_r$, crossover $p_c$, and mutation $p_m$, parameter $\sigma$ in Eq. (9), and number of generations $G$ (or the objective function $J$ is less than a tolerance $\varepsilon$).

*Goal*: Search for the associated coefficients of nonlinear discrete-time multivariable polynomial model in Eq. (4) such that the value of $J$ in Eq. (5) is minimized.

1. Create a population with the size of $N$ chromosomes, in which all genes (model coefficients) are randomly generated from $\Omega_\Theta$.
2. Evaluate the objective function $J$ of Eq. (5) for each chromosome in the population.
3. If the pre-specified number of generations $G$ is reached or there is a chromosome with objective function $J$ less than $\varepsilon$, then stop.
4. Carry out operators of reproduction, proposed multiple crossover in Eq. (8), and mutation in Eq. (9). If any resulted chromosome during the operations is outside the $\Omega_\Theta$, then the original is kept.
5. Go back to Step 2.

## 4. Case study

Consider a multivariable nonlinear discrete-time system with two inputs and two outputs given by [21]

$$\begin{aligned} y_1(k) &= 0.8y_1(k-1) + u_1(k-2) - 1.2u_1(k-1)u_2(k-2) + 0.4u_1^2(k-2) - 0.1y_2(k-1) + w_1(k), \\ y_2(k) &= 0.5y_2(k-1) + u_2(k-2) + u_1^2(k-1) + 0.5y_2(k-2)u_2^2(k-1) + w_2(k). \end{aligned} \tag{10}$$

Comparing Eq. (10) with Eq. (3), the corresponding coefficients and functions can be easily observed; for example, $\theta_1 = 0.8, p_{11}(k) = y_1(k-1), \theta_2 = 1.0, p_{12}(k) = u_1(k-2)$ and so forth. In the simulation, the inputs $u_1$ and $u_2$ are set to be uniformly distributed between the range $(0, 1)$, the measurement noises $w_1$ and $w_2$ are also independent random variables uniformly distributed in the range $(-0.001,\ 0.001)$. Additionally, the related variables employed in GA are given by

$$\begin{aligned} &T = 50, \quad N = 30, \quad p_r = 0.2, \quad p_c = 0.3, \quad p_m = 0.2, \quad \sigma = 0.005, \quad G = 5000, \\ &\hat{\theta}_{i\,\min} = -2.0, \quad \hat{\theta}_{i\,\max} = 2.0, \quad i \in \underline{9}. \end{aligned}$$

Tables 1 and 2 list the final estimation results. All of multivariable nonlinear system coefficients are accurately solved. Furthermore, to verify the effectiveness of the proposed method than that of the traditional crossover, Fig. 5 also shows a comparison of convergence speed. The optimal objective function among the population of $N = 30$ chromosomes is plotted with respect to the number of generations. It is obvious that the proposed method gives a quicker convergence than the traditional crossover method.

Table 1
A comparison of coefficients $\theta_1$ to $\theta_5$ to between the actual and estimated values by the proposed method

| Coefficient | $\theta_1$ | $\theta_2$ | $\theta_3$ | $\theta_4$ | $\theta_5$ |
|---|---|---|---|---|---|
| Actual value | 0.8 | 1.0 | −1.2 | 0.4 | −0.1 |
| Estimated value | 0.7993 | 1.0017 | −1.1993 | 0.3976 | −0.0998 |

Table 2
A comparison of coefficients $\theta_6$ to $\theta_9$ between the actual and estimated values by the proposed method

| Coefficient | $\theta_6$ | $\theta_7$ | $\theta_8$ | $\theta_9$ |
|---|---|---|---|---|
| Actual value | 0.5 | 1.0 | 1.0 | 0.5 |
| Estimated value | 0.5003 | 1.0003 | 0.9997 | 0.4992 |

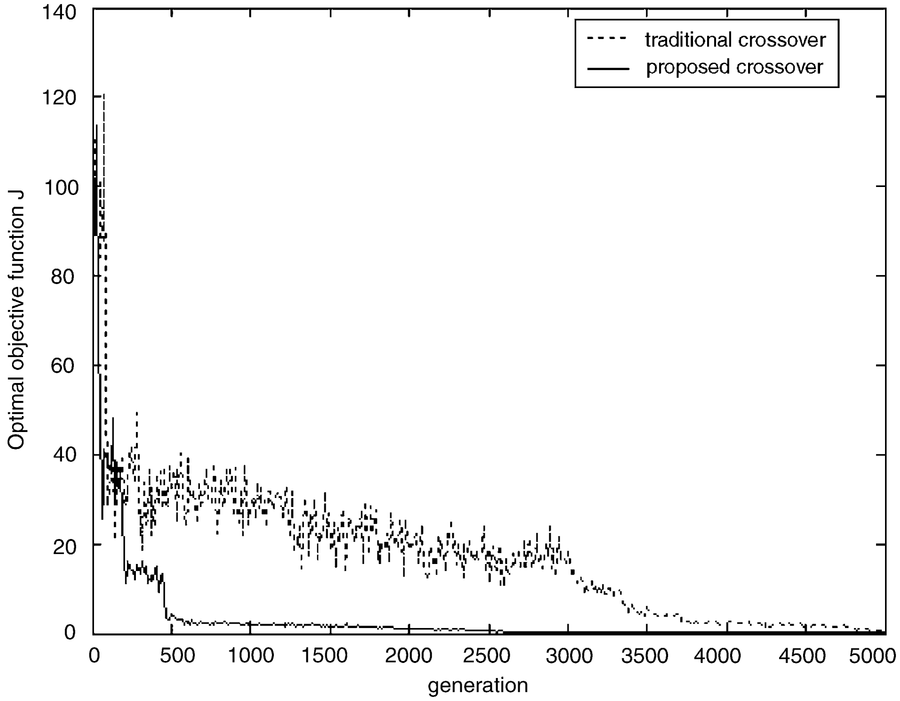

Fig. 5. Convergence trajectories of optimal objective function $J$ using the proposed and traditional crossovers, respectively.

## 5. Conclusions

This paper has successfully applied the multi-crossover GA to solve the system identification problem for multivariable nonlinear discrete-time polynomial systems. The proposed method utilizes three parent chromosomes to achieve a crossover and a new adjusting direction is then obtained. Based on the adjusting direction, the excellent offspring with better performance can be generated. We also define the suitable objective function for such the multivariable systems. All of chromosomes within the population undergo three evolutionary operations: reproduction, multiple crossover, and mutation such that the given objective function is minimized and then the accurate coefficient estimations are derived. Simulation results have shown that the validity of the proposed multiple crossover is verified and its convergence speed is quicker than that of the traditional crossover.

## References

[1] S. Omatu, M. Khalid, R. Yusof, *Neuro-Control and Its Applications*, Springer, London, 1995.

[2] R.J. Wai, H.H. Lin, F.J. Lin, Hybrid controller using neural networks for identification and control of induction servo motor drive, *Neurocomputing* 35 (2000) 91–112.

[3] L.X. Wang, *Course in Fuzzy Systems and Control*, Prentice-Hall, Englewood NJ, 1997.

[4] Y. Tang, N. Zhang, Y. Li, Stable fuzzy adaptive control for a class of nonlinear systems, *Fuzzy Sets and Systems* 104 (1999) 279–288.

[5] Y.G. Leu, T.T. Lee, W.Y. Wang, Observer-based adaptive fuzzy-neural control for unknown nonlinear dynamical systems, *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 29 (5) (1999) 583–591.

[6] C.T. Chen, S.T. Peng, Intelligent process control using neural fuzzy technique, *Journal of Process Control* 9 (1999) 493–503.

[7] K.J. Astrom, B. Wittenmark, *Adaptive Control*, Addison-Wesley, Massachusetts, 1995.

[8] K. Kristinsson, G.A. Dumont, System identification and control using genetic algorithms, *IEEE Transactions on Systems, Man, and Cybernetics* 22 (5) (1992) 1033–1046.

[9] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand, New York, 1991.

[10] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA, 1989.

[11] Y.P. Huang, C.H. Huang, Real-valued genetic algorithm for fuzzy grey prediction system, *Fuzzy Sets and Systems* 87 (1997) 265–276.

[12] B. Jiang, B.W. Wang, Parameter estimation of nonlinear system based on genetic algorithm, *Control Theory and Applications* 17 (1) (2000) 150–152 (in Chinese).

[13] L. Yao, W.A. Sethares, Nonlinear parameter estimation via the genetic algorithm, *IEEE Transactions on Signal Processing* 42 (4) (1994) 927–935.

[14] S. Tsutsui, D.E. Goldberg, Search space boundary extension method in real-coded genetic algorithm, *Information Sciences* 133 (2001) 229–247.

[15] A. Blanco, M. Delgado, M.C. Pegalajar, A real-coded genetic algorithm for training recurrent neural networks, *Neural Networks* 14 (2001) 93–105.

[16] K. Deb, S. Gulati, Design of truss-structures for minimum weight using genetic algorithms, *Finite Elements in Analysis and Design* 37 (2001) 447–465.

[17] M. Zamparelli, Genetically trained Cellular neural networks, *Neural Networks* 10 (6) (1997) 1143–1151.

[18] O.A.Z. Sotomayor, S.W. Park, C. Garcia, Multivariable identification of an activated sludge process with subspace-based algorithms, *Control Engineering Practice* 11 (2003) 961–969.

[19] S. Tan, J. Hao, J. Vandewalle, Efficient identification of RBF neural net models for nonlinear discrete-time multivariable dynamical systems, *Neurocomputing* 9 (1995) 11–26.

[20] M. Lu, K. Liu, Recursive relaxation identification of linear multivariable systems with its parallel algorithm, *Information Sciences* 89 (1996) 211–223.

[21] K.Z. Mao, S.A. Billings, Variable selection in non-linear systems modeling, *Mechanical Systems and Signal Processing* 13 (2) (1999) 351–366.