International Journal of Innovative
Computing, Information and Control
Volume 8, Number 10(B), October 2012

ICIC International ©2012 ISSN 1349-4198
pp. 7341–7362

POLYNOMIAL NARX MODEL STRUCTURE OPTIMIZATION USING MULTI-OBJECTIVE GENETIC ALGORITHM

Sayed Mohammad Reza Loghmanian, Rubiyah Yusof*, Marzuki Khalid and Fatimah Sham Ismail

Centre for Artificial Intelligence and Robotics
Universiti Teknologi Malaysia
54100 Jalan Semarak, Kuala Lumpur, Malaysia
*Corresponding author: rubiyah@ic.utm.my

Received March 2011; revised October 2011

Abstract. *Model structure selection is an important step in system identification which involves the selection of variables and terms of a model. The important issue is choosing a compact model representation where only significant terms are selected among all the possible ones beside good performance. This research explores the use of multi-objective optimization to minimize the complexity of a model structure and its predictive error simultaneously. The model structure representation is a polynomial non-linear auto-regressive with exogenous input model. A new modified elitist non-dominated sorting genetic algorithm using clustered crowding distance (CCD) is proposed to find the exact model among non-dominated solutions, using some simulated examples which generate data set by mathematical equations. Simulation results demonstrated that the proposed algorithm can find the correct model with exact terms and values in all cases of problem. Furthermore, the effectiveness of the proposed algorithm is also studied by applying to the real process data sets, and the final model can be chosen from a set of non-dominated solutions referred as Pareto optimal front. The results show that the proposed clustered CD has better performance compared with the basic CD method.*
Keywords: NARX, Multi-objective genetic algorithm, NSGA-II, System identification, Model structure selection, Clustered crowding distance

1. **Introduction.** The main task in model structure selection is to determine and select the significant terms to be included in the final model. Selecting a model with a large number of terms increases the complexity of a model and computation time. Meanwhile, it may cause over-fitting of data. On the other hand, simplifying a model structure too much will give inferior performance even for training data, thus the need for structural optimization. Forward regression orthogonal least square (OLS) is one of the successful methods for model structure selection [1-5]. OLS selects significant terms based on Error Reduction Ratio (ERR). There are two disadvantages for this method. The estimator yields different orthogonalization paths and the number of possible paths ($M!$) increases as the number of possible terms M increases. The second is the difficulty in deciding the ERR value to stop regression especially when these values reduce gradually.

Another method for model structure selection is genetic programming (GP) which has been studied by some researchers [6-8]. In genetic programming, a population of model structures is represented as trees and evolves through many generations towards solution using evolutionary operators such as crossover, mutation and selection. One of the initial steps in GP is the definition of a set of terminals and a set of functions appropriate to the specific problem. The terminal set refers to the nodes in the tree and the set of function refers to operation to generate mathematical expressions.

Recently, genetic algorithm (GA) is extensively used for model structure selection. As in GP, a simple GA (SGA) consists of three basic operators: reproduction, crossover and mutation. The operations are systematically and repeatedly applied to the population until an acceptable solution is found. Optimization algorithm using genetic algorithm is based on the mechanics of natural selection and evolution. This algorithm evaluates multiple points in the solution space simultaneously and therefore, it has potential to converge to the global optimum solution [9,10]. Researchers such as Kristinsson and Dumont [11], Zibo and Naghdy [12], Jeong and Lee [13], Sheta and De Jong [14], Hong and Billings [15] gave related works concerning the application of genetic algorithms in system identification. Most of the works assumed that the structures of the models were known and GA was applied for estimating the parameters of those models [13,16].

An improved strategy for selecting and exploring potential regions in genetic algorithm has been proposed and known as modified genetic algorithm (MGA) that provides better searching mechanism in finding the correct model structure. Researchers have applied genetic algorithm for various applications [17-22] and found ways to better improve its performance. An MGA-based model structure selection algorithm has been developed by [20], which has been tested on a variety of linear and non-linear systems with promising results. Samad [21] explored the use of evolutionary computation in model structure selection and found out that a suitable penalty function parameter can be achieved by its relation to the smallest estimated and tolerable parameter value. These techniques optimize the model structure based on minimization of the predictive error and some strategies such as elitist and grouping of population to undergo further mutation and crossover [23]. However, the proposed methods need to identify the mentioned terms before starting the optimization process using trial and error.

The field of search and optimization has changed over the last years by the introduction of a number of non-classical, unconventional and stochastic search and optimization algorithms. A number of stochastic optimization techniques such as simulated annealing, tabu search and ant colony optimization, could be used to generate Pareto set [24]. The solutions attempt to obtain good approximation but they do not guarantee to identify optimal trade-offs. All these evolutionary algorithms (EAs) are characterized by a population of solution candidates and the reproduction process enables the combination of existing solutions to generate new solutions [25,26]. Srinivas and Deb [27] presented the non-dominated sorting GA (NSGA) as Pareto-based approach. The main advantage of the algorithm is the assignment of fitness according to non-dominated sets. Nevertheless, the performance is sensitive to the sharing parameter. To overcome the disadvantage, the elitist non-dominated sorting genetic algorithm (NSGA-II) has been proposed [26].

In this paper, elitist non-dominated sorting genetic algorithm is modified to model the non-linear dynamic system that is represented by NARX model. A new method that is called clustered crowding distance (CCD) based on modification of crowding distance (CD) is proposed. This approach categorizes all solutions in some subgroups, where each subgroup must consist of non-repetitive solutions, and then CD is applied to the subgroups. This method is applicable for problems with discrete design variables. This study will apply this method to optimizing the complexity of the structure of polynomial NARX model. Firstly, the simulated data that are generated from dynamic NARX structures model will be used. Since the structures are known, the final models can be validated. After proving the effectiveness of the algorithm using simulated models with different nonlinearity, the algorithm will be tested using real data of gas furnace. Model validity tests will also be conducted to test the adequacy and the effectiveness of the developed model.

2. **NARX Model.** The first step in model structure selection is to choose the size of model sets such as determining the order of the state for a state-space model, the polynomial degree for a polynomial model and the number of hidden nodes for a neural network model [28]. The advantage of NARX models consists in availability of various methods for their approximation and simplicity of controls design as well as its simplicity and fast convergence of identification. This type of model has broad applications such as prediction, simulation and identification. The main issue in identification of these polynomial models is finding the order of the polynomial and choosing the significant terms from a large collection of combination sets that exist for this model representation. The NARX model provides a unified representation of a wide class of non-linear systems. The model is stated as (1).

$$y(t) = f^{n_l}(u(t-1), \ldots, u(t-n_u), y(t-1), \ldots, y(t-n_y)) + e(t) \tag{1}$$

where n_u and n_y are the corresponding maximum lags for input and output, $e(t)$ accounts for possible noise and uncertainties and its inclusion to the equation is usually to avoid bias in the parameters, n_l is the degree of non-linearity and f^{n_l} can be assumed as a variety of non-linear function forms based on n_l. The identification of non-linear system is more difficult compared with linear model since it involves more terms. The NARX model can be expanded as the summation of terms with the degree of non-linearity in the range of $1 \leq n_l \leq L$ as follows:

$$y(t) = \sum_{l=0}^{L} \sum_{p=0}^{n_l} \sum_{n_1,n_l}^{n_y,n_u} c_{p,n_l-p}(n_1, \ldots, n_l) \prod_{i=1}^{p} y(t-n_i) \prod_{i=p+1}^{n_l} u(t-n_i) \tag{2}$$

where each n_lth degree of non-linearity term consists of pth degree factor in $y(t-n_i)$ and a (n_l-p)th degree factor in $u(t-n_i)$, multiplied by $c_{p,n_l-p}(n_1, \ldots, n_l)$ the term coefficients.

The maximum number of terms in polynomial NARX model is given by

$$M = \sum_{i=1}^{n_l} n_i, \text{ for } n_l = \text{order of non-linearity} \tag{3}$$

$$\text{and } n_i = \frac{n_{i-1}(n_y + n_u + i - 1)}{i}, \quad n_0 = 1 \tag{4}$$

The maximum number of all possible model structures is $(2^M - 1)$. Increasing the order of the dynamic terms, which are the values of n_y, n_u and the degree of non-linearity n_l, in order to get the desired prediction accuracy gives the full terms of the model which will result in a complex model and heavy computation. For example, a first order dynamic model for input and output with third degree of non-linearity will contain 9 terms and $2^9 - 1 = 511$ possible models, while for fifth degree of non-linearity will contain 14 terms and $2^{14} - 1 = 16,383$ possible models. This shows that in NARX model, the number of terms is enormously large and some of these terms are unnecessary. Therefore, it is important to select only the significant terms for the model that will produce a parsimonious model that will sufficiently represent the data set.

In most reported works in modelling nonlinear systems, one-step-ahead prediction has been used to verify the models. However, this is not a sufficient indicator of model performance because at each step the past inputs, outputs and residuals are available and used to predict just one increment forward. The one step-ahead prediction is given by (5):

$$\hat{y}_{OSA}(t) = \hat{F}[(y(t-1), \ldots, y(t-n_y), u(t-1), \ldots, u(t-n_u))] \tag{5}$$

where $\hat{F}$ is the estimated model.

Model validation such as correlation test is another validation technique that can detect deficiency using the prediction errors or residuals. If a model of a system is adequate then the residual or predictive error $\varepsilon(t)$ should be unpredictable from all linear and nonlinear combinations of past inputs and outputs. The derivation of simple tests which can detect these conditions is complex but it can be shown that [29] the following conditions should hold:

$$\begin{aligned}
\Phi_{\varepsilon\varepsilon}(\tau) &= \frac{E[\varepsilon(t)\varepsilon(t-\tau)]}{E[\varepsilon^2(t)]} = \delta(\tau) \quad \tau = 0 \\
\Phi_{u\varepsilon}(\tau) &= \frac{E[u(\tau)\varepsilon(t-\tau)]}{\sqrt{E[u^2(\tau)\varepsilon^2(t)]}} = 0 \quad \forall\tau \\
\Phi_{\varepsilon(\varepsilon u)}(\tau) &= \frac{E[\varepsilon(t)\varepsilon(t-1-\tau)u(t-1-\tau)]}{\sqrt{E[\varepsilon^2(t)]E[\varepsilon^2(t)u^2(t)]}} = 0, \quad \tau \geq 0 \\
\Phi_{u^2\varepsilon}(\tau) &= \frac{E[(u^2(t)-\bar{u}^2)\varepsilon(t-\tau)]}{\sqrt{E[(u^2(t)-\bar{u}^2)^2]E[\varepsilon^2(t)]}} = 0, \quad \forall\tau \\
\Phi_{u^2\varepsilon^2}(\tau) &= \frac{E[(u^2(t)-\bar{u}^2)\varepsilon^2(t-\tau)]}{\sqrt{E[(u^2(t)-\bar{u}^2)^2]E[\varepsilon^4(t)]}} = 0, \quad \forall\tau
\end{aligned} \tag{6}$$

where Φ represents the standard correlation function, E is the expectation operator, $\delta(\tau)$ is an impulse function and $\varepsilon(t)$ represents the prediction errors or residual,

$$\varepsilon = \hat{y} - y \tag{7}$$

where $\hat{y}$ is the predicted output, and

$$\bar{u}^2 = \frac{1}{N_p}\sum_{t=1}^{N_p} u^2(t) \tag{8}$$

These tests are able to indicate the adequacy of the fitted model. Generally, if the correlation functions are within the 95% confidence interval, i.e., $\pm 1.96/\sqrt{N_p}$, the model is regarded as adequate where N_p is the number of data points.

3. **Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II).** Deb [26] presented the elitist non-dominated sorting genetic algorithm (NSGA-II). This method uses an explicit diversity preserving mechanism. For a multi-objective optimization problem, any two solutions a and b can have one of two possibilities: one dominates the other or none dominates the other. In a minimization problem, without loss of generality, a solution a dominates b if the following two conditions are satisfied:

$$\begin{aligned}
&\forall m \; f_m(a) \leq f_m(b) \quad m = 1, 2, \ldots, g \quad a, b \in R^n \\
&\exists m \; f_m(a) < f_m(b) \quad m = 1, 2, \ldots, g \quad a, b \in R^n
\end{aligned} \tag{9}$$

If any of the above conditions are not violated, solution a dominates solution b. If there is not a solution like a which dominates solution b, b is called the non-dominated solution. The solutions that are non-dominated within the entire search space are denoted as Pareto optimal and constitute the Pareto optimal set or Pareto optimal front.

After creation of new population Q by using parent P of size N, two populations are combined together and a new population R of size $2N$ is constructed. Then the process of finding the non-dominated sorting, sorts the population in R. After non-dominated sorting, the new population is constructed by different non-dominated front. It starts with the best one (rank one or first front) and continues with second non-dominated front and so forth. Since the size of R is $2N$ and new population needs N individuals, in the

case of equality in their front level they will be selected according to crowding distance. The crowding distance is used by NSGA-II to maintain the diversity among solutions in a front. The crowding distance for an individual is calculated using hypercube [26].

For selection between two solutions with the same rank, the method chooses the one with larger distance. The advantages of NSGA-II are that an elite-preservation strategy and an explicit diversity-preserving mechanism are used simultaneously. These mechanisms do not allow an already found Pareto-optimal solution to be deleted and keep the diversity of solution. The implementation steps of NSGA-II can be summarized as follows.

Step 1. Generate binary initial population P^0 of size N according to the number and length of the decision variables.

Step 2. Calculate the objective functions for each chromosome (individual).

Step 3. Assign number of rank as the fitness function to each chromosome by the non-dominated sorting procedure and classify them into distinct Pareto fronts.

Step 4. Create offspring population P^t ($t = 1$) from initial population P^0 by GA operators namely selection, crossover and mutation.

Step 5. Compute the objective functions for each chromosome in P^t.

Step 6. Create a combine population of size $2N$ consisting of old and new generations, $R^t = P^t \cup P^{t-1}$.

Step 7. Perform non-dominated sorting and compute crowding distance for all chromosomes.

Step 8. Select lower ranked solutions and put them in the new population of size N, P^{t+1}.

Step 9. In the case of equality of front number among the solutions, choose the chromosomes with a higher crowding distance.

Step 10. A new population is ready, P^{t+1}, go to Step 4 and repeat until Pareto optimal front is obtained.

4. **Clustered Crowding Distance.** To estimate the density of the solutions surrounding a particular solution i in the population, the average distance of two solutions on either side of solution i along each of the objectives must be calculated. The quantity d_i serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbours as the vertices and is called crowding distance. In Figure 1, the crowding distance of the ith solution in its front is the average side length of the cuboid (dashed box). The following algorithm is applied to calculate crowding distance as presented by Deb [26] of each point (i) in a particular set F.

Step 1. For all solution in the set, assign $d_i = 0$.

Step 2. For each objective function $m = 1, 2, \ldots, M$, sort the set in worse order of f_m or find the sorted indices vector: $I^m = \text{sort}(f_m)$.

Step 3. For $m = 1, 2, \ldots, M$, assign a large distance to the boundary solutions, or $d_{I_1^m} = d_{I_j^m}$, and for all other solutions $j = 2$ to $(l - 1)$, assign:

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{\max} - f_m^{\min}} \tag{10}$$

The index I_j denotes the solution index of the jth member in the sorted list. Thus, for any objective, I_1 and I_l denote the lowest and highest objective function value indices, respectively (l = number of solution in F). The second term on the right side in Equation (10) is the difference in objective function values between two neighboring solutions on either side of solution I_j. Thus, this metric denotes half of the perimeter of the enclosing cuboid with the nearest neighboring solutions placed on the vertices of the cuboid. The

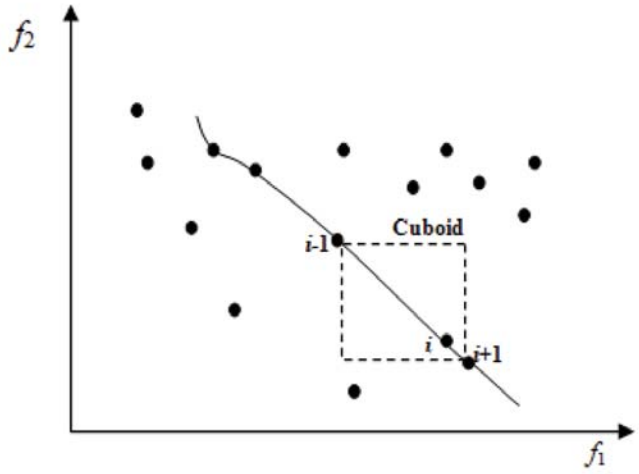

FIGURE 1. Calculation of crowding distance for ith solution

parameter $f_m^{\max}$ and $f_m^{\min}$ can be set as the population maximum and minimum values of the mth objective function. This method is referred as CD algorithm.

In problems with discrete design variables especially at the last generations some repetitive solutions will be appeared. Using the CD algorithm, all repetitive solutions are assigned the same crowding distance values and there is no criterion for selection among solutions with the same crowding distance. Thus, some solutions will be chosen several times while the rest not. Consequently the Pareto optimal front does not have density preservation. One way to overcome this problem may be the use of only non-repetitive solutions. However, the number of non-repetitive solutions is not sufficient to complete the empty slots in the population.

A new method called clustered crowding distance (CDD) was proposed and employed to overcome this problem for some mathematical benchmark problems with discrete design variables. The new method is applied for optimization of NARX model structure as its design variables such as number of required, number of input-output lags and nonlinearity order are discrete. This method classifies all solutions to some subsets, so that there are only non-repetitive solutions in each subset. Then the crowding distance of solutions is calculated for each batch and finally, the required number of solutions is selected from the first subset, followed by second subset and so forth. The following steps show the procedure of CCD.

Step 1. Classify all solutions into different fronts, F_i, where i is the number of fronts.
Step 2. In each set of F_i identify repetitive solutions.
Step 3. Form F_i^j, where j is the number of subset including non-repetitive solutions. For example, if $F_1 = \{a, b, a, b, c, d, d, d, c, d, e\}$, then

$$F_1^1 = \{a, b, c, d, e\}$$
$$F_1^2 = \{a, b, c, d\}$$
$$F_1^3 = \{d\}$$
$$F_1^4 = \{d\}$$

Step 4. Calculate crowding distance for every subset F_i^j based on Equation (10).
Step 5. Select solution to fill up the population from the first subset.

5. **Experimental Simulation.** Two objective functions can be formulated mathematically as follows:

$$\begin{cases} \text{Predictive error, } \min: MSE = \dfrac{1}{N_s}\displaystyle\sum_{i-1}^{N_s}(y_i(t) - \hat{y}_i(t))^2 \\ \text{Complexity, } \min: CX = n_u + n_y + n_l + n_{term} \end{cases} \tag{11}$$

where N_s is number of samples, $y(t)$ and $\hat{y}(t)$ are the desired and predicted system output respectively and n_u, n_y and n_l are the number of input and output lags and non-linearity order respectively and n_{term} is the number of sufficient term in NARX model. The limits of the variables are

$$\begin{cases} L_{nu} \le n_u \le U_{nu} \\ L_{ny} \le n_y \le U_{ny} \\ 1 \le n_l \le U_{nl} \\ 1 \le n_{term} \le 2^M - 1 \end{cases} \tag{12}$$

where L_{nu} and L_{ny} are the minimum number of input and output lags respectively and U_{nu}, U_{ny} and U_{nl} are the maximum number of input and output lags and nonlinearity order respectively and M is maximum number of term calculated by (3) and (4).

It is important to note that more precise models need higher upper limits of lags and order, but computational load will be increased. First step of initialization is associated with multi-objective optimization. That is, the number of input and output lags and nonlinearity order. When the chromosomes formed in the initial population of size N, to minimize the number of require terms (n_{term}) in NARX structure, another optimization process (sub-optimization) is carried out and gives the predictive error, MSE, as the first objective function. Then, the complexity for the second objective function is calculated using the summation of n_u, n_y, n_l and n_{term}.

Crossover and mutation operators create a new population from the previous population and both populations are placed in the mating pool with the size of $2N$. Based on NSGA-II regulations, non-dominated sorting and crowding distance are computed for all individuals in the mating pool for fitness functions evaluation. Selection process chooses N chromosomes among the $2N$ for the next generation according to lower rank, and if necessary for higher crowding distance. The above procedures are performed until end of iteration and the Pareto optimal front based on convergence and diversity metrics are obtained. The implementation can be summarized as in Figure 2.

5.1. **Case studies.** To show the effectiveness of the proposed algorithm, three simulated model with polynomial NARX structure referred as $S1$ to $S3$ were used. In this case, structures and parameters are known. These examples are used to investigate the capability of the algorithm to detect and select the correct terms and parameter. System $S1$ and $S2$ present second and third order polynomial NARX model, respectively and $S3$ has been used by Mao and Billings [30].

$$\text{(S1)}: \; y(t) = 0.95y(t-1) + 0.7u(t-1) - 0.5u(t-3) - 0.37y(t-3)u(t-3) \tag{13}$$

$$\text{(S2)}: \; y(t) = 0.5y(t-1) + 0.3u(t-2) + 0.3y(t-1)u(t-1) + 0.5u^3(t-1) \tag{14}$$

$$\text{(S3)}: \begin{aligned} y(t) = {} & 0.2y^3(t-1) + 0.7y(t-1)u(t-1) + 0.6u^2(t-2) \\ & - 0.5y(t-2) - 0.7y(t-2)u^2(t-2) \end{aligned} \tag{15}$$

The corresponding coefficients and lags are summarized in Table 1. Thus, the expected results are at least one among the final solutions of Pareto optimal front equivalent to the structure of $S1$, $S2$ and $S3$.

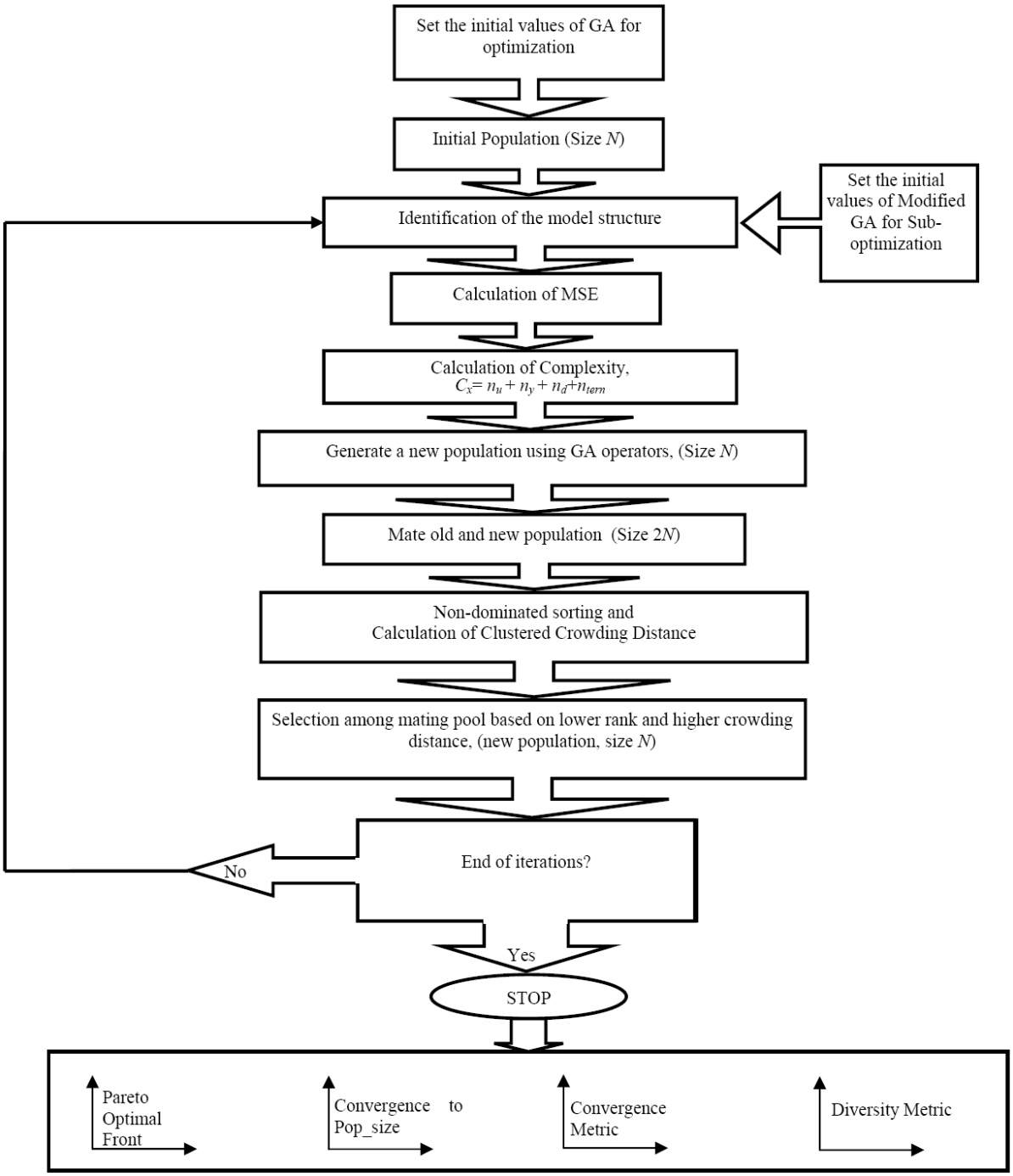

FIGURE 2. Flow chart of the process

TABLE 1. The expected results for $S1$ to $S3$

Systems	Input Lags	Output Lags	Cross lags	Degree of non-linearity
$\boldsymbol{S1}$	$0.7u(t-1)$ $-0.5u(t-3)$	$0.95y(t-1)$	$-0.37y(t-3)u(t-3)$	2
$\boldsymbol{S2}$	$0.3u(t-2)$ $0.5u^3(t-1)$	$0.5y(t-1)$	$0.3y(t-1)u(t-1)$	3
$\boldsymbol{S3}$	$0.6u^2(t-2)$	$-0.5y(t-2)$ $0.2y^3(t-2)$	$0.7y(t-1)u(t-1)$ $-0.7y(t-2)u^2(t-1)$	3

To show the application of the proposed algorithm on real process data set, the Box-Jenkins gas furnace data ($S4$) has been used [31]. For $S1$ to $S3$, the input $u(t)$ is a zero mean uniformly distributed white noise sequence. Five hundred data points were generated between -1 and $+1$. Then the desired output, $y(t)$, is calculated using (13) till (15) as well as for $S1$ to $S4$ the number of training and test data is shown in Table 2.

TABLE 2. Data division of case studies

	No. of Data	**No. of Training set**	**No. of Test set**
***S*1-*S*3**	500	400	100
***S*4, Gas furnace data**	296	200	96

5.2. **Parameters in genetic algorithm.** The control parameters of CCD GA include population size, crossover and mutation probability, crossover and selection strategy. The choice of these parameters can affect the behavior and performance of GA whether in single or multi-objectives cases, [32,33].

The GA parameters used in the study were:

(i) Population size: population size of 20, 50 and 100.

(ii) Crossover probabilities: probabilities of 0.05, 0.3, 0.6, 0.9.

(iii) Mutation probabilities: probabilities of 0.001, 0.01, 0.1.

(iv) Crossover mechanism: single crossover and double crossover.

6. Results and Discussion.

6.1. **CCD GA performance.** This study has analyzed the effect of different population sizes of CCD GA method as shown in Figure 3 for $S3$. The convergence for 20 population size is faster than the others, while the diversity metrics indicate the roughly same results. However, it will take longer time to converge to the correct solution. For a small number of populations, most of the highly superior individuals dominate the population towards later generation giving them the chance of being selected for the next operation. The disadvantage is that there will be a chance of not selecting the better solution within the whole population and will result in poor exploration. To investigate more, the algorithm was applied 30 times for all three above population size and the solution at last generation was investigated. In some attempts for population size of 20, the expected solution was not within non-dominated solutions. While for sizes of 50 and 100 in all attempts there were expected solutions. Thus, it seems that, the population size of 50 is the best choice for this study and due to time comsumption increasing to 100 is not necessary.

By varying the crossover probability rate, P_c, and mutation probability rate, P_m, the performance of the algorithms was observed and the results are shown in Figures 4 till 7. The results indicate that there are no specific rules for values of P_c and P_m and these values have to be chosen by trial and error. However, the $P_m = 0.1$ have given better result among the others. Although, the crossover probability of 0.9 shown acceptable performances.

Figures 8(a) and 8(b) show the convergence and diversity metrics respectively for the multi-objective optimization of NARX model structure using CCD NSGA-II with these two crossover strategies. The result shows that a double crossover strategy slightly gives better diversity preservation. For comparison, all the results are summarized in Table 3.

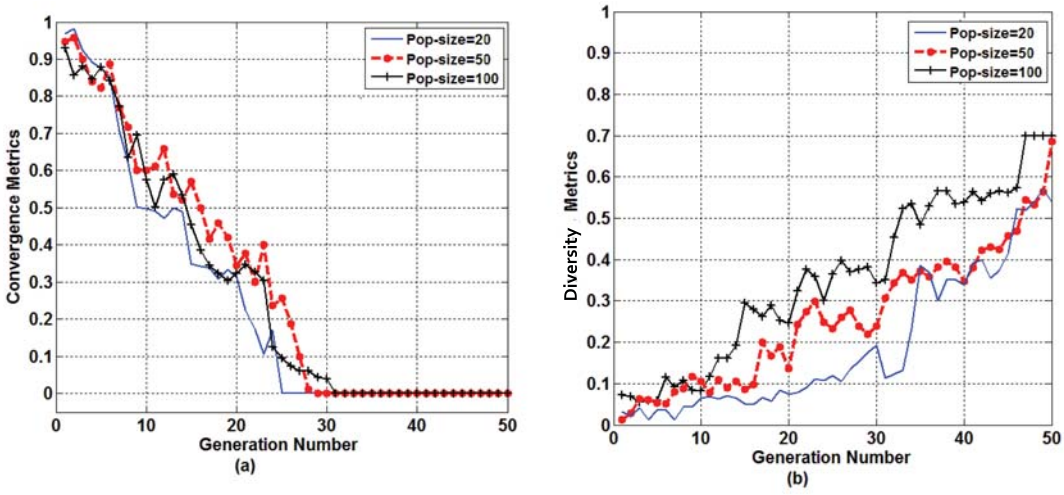

FIGURE 3. The effect of varied population size on (a) convergence and (b) diversity of system $S3$

TABLE 3. The initial parameters and values used in the algorithm

Parameter	Value	
	$S1$ to $S3$	$S4$
Population size	50	50
No of Generation	50	50
Probability of crossover (P_c)	0.9	0.8
Probability of mutation (P_m)	0.1	0.1
Max No of input lags	5	5
Max No of output lags	5	5
Max No hidden neurons	8	8
Error function	MSE (test set)	MSE (test set)

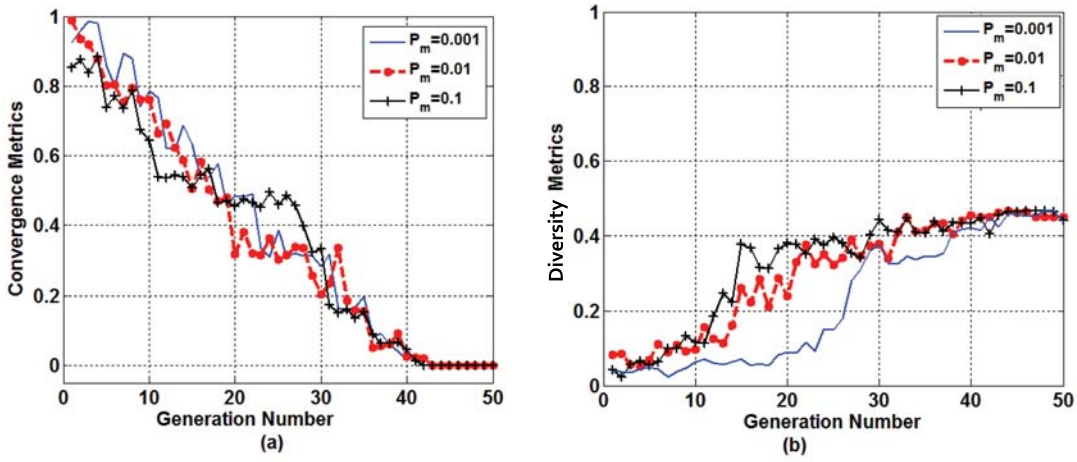

FIGURE 4. The effect of varying mutation rate at $P_c = 0.05$ for system $S3$, (a) convergence metrics and (b) diversity metrics

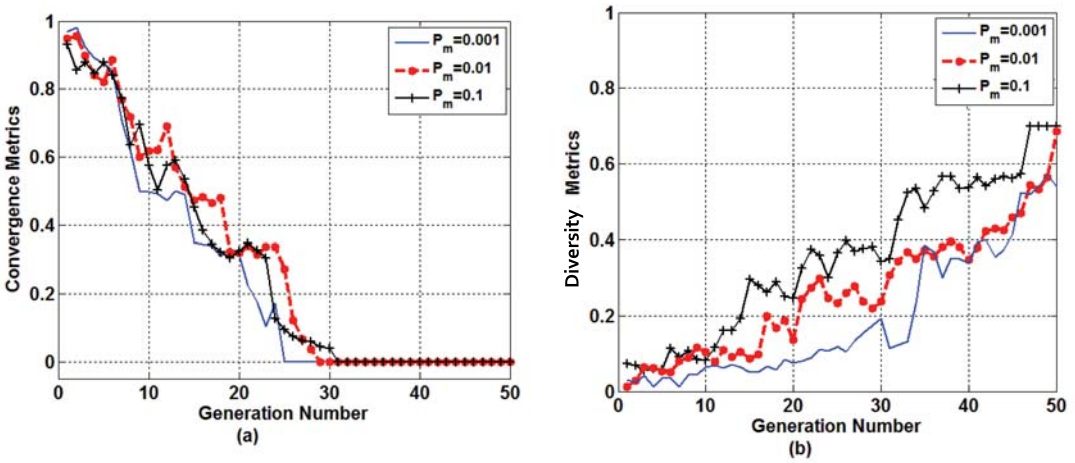

FIGURE 5. The effect of varying mutation rate at $P_c = 0.3$ for system $S3$, (a) convergence metrics and (b) diversity metrics

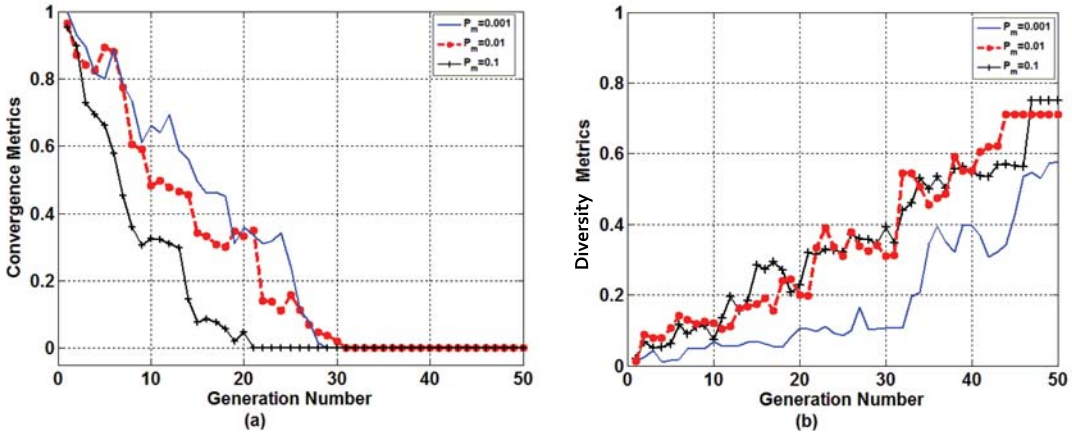

FIGURE 6. The effect of varying mutation rate at $P_c = 0.6$ for system $S3$, (a) convergence metrics and (b) diversity metrics

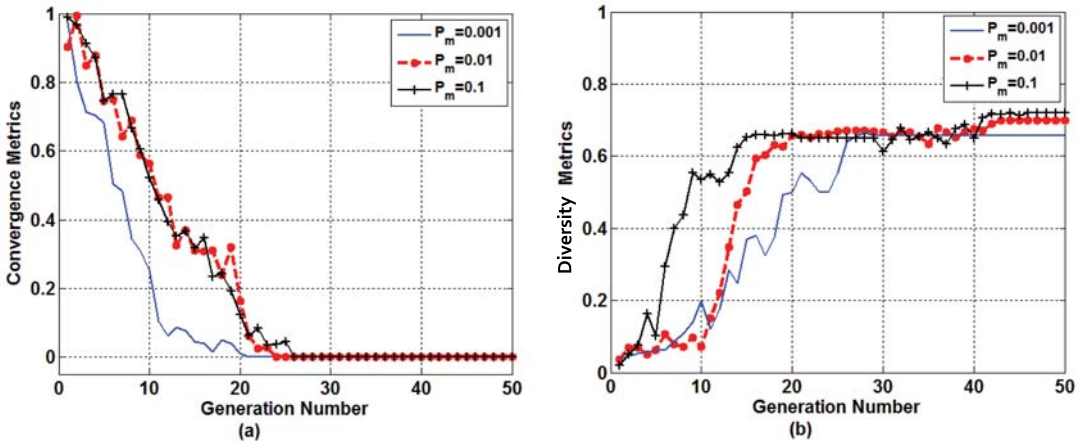

FIGURE 7. The effect of varying mutation rate at $P_c = 0.9$ for system $S3$, (a) convergence metrics and (b) diversity metrics

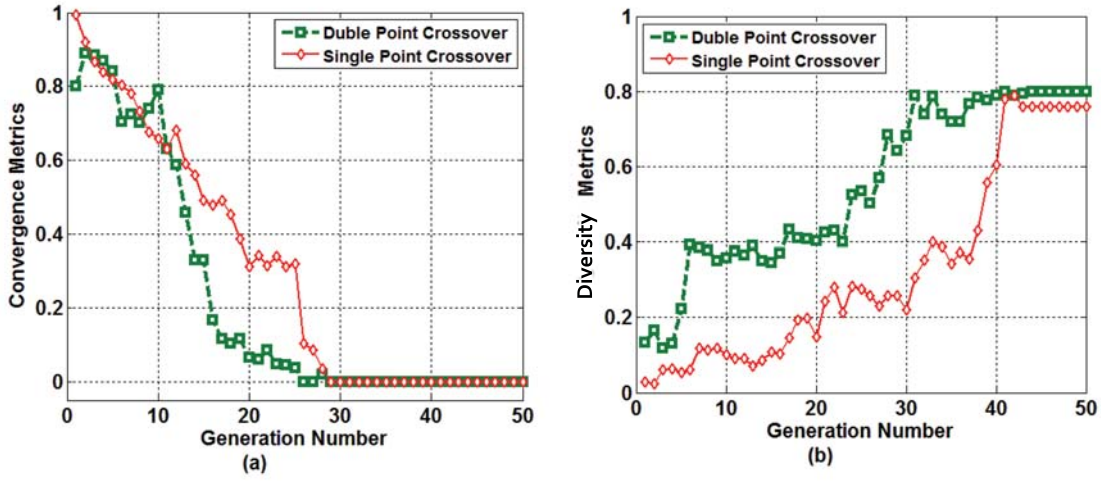

FIGURE 8. The effect of varying crossover point for system $S3$

6.2. **Simulated NARX models.** The obtained data sets were modelled as NARX model and the proposed multi-objective optimization algorithm was applied to optimize the MSE and complexity of the model structure using modified CCD NSGA-II. All initial values and parameters for NARX model and CCD NSGA-II were set according to the presented values in Table 3.

Two goals must be investigated in every multi-objective optimization problem. These are convergence and diversity of solutions in Pareto optimal front. These two metrics were investigated for all simulated systems $S1$ to $S3$. For instance, the graphs for $S3$ are shown in Figure 9. The convergence metrics moved to zero, thereby implying that CCD NSGA-II solutions start from a random set of solutions and approach the Pareto optimal front. A value of zero of the convergence metrics indicates that all non-dominated solutions match the Pareto optimal points.

After about 25 generations, the CCD NSGA-II population comes very close to the Pareto optimal front. As shown in Figure 9(b), the diversity increases until about 45, after which the diversity remains the same.

The collections of all non-dominated solutions in Pareto optimal set are indicated in Figures 10 to 12. It can be observed that, each Pareto optimal front contains less than 50 solutions. It happens due to two reasons: (a) some solutions in last generation do

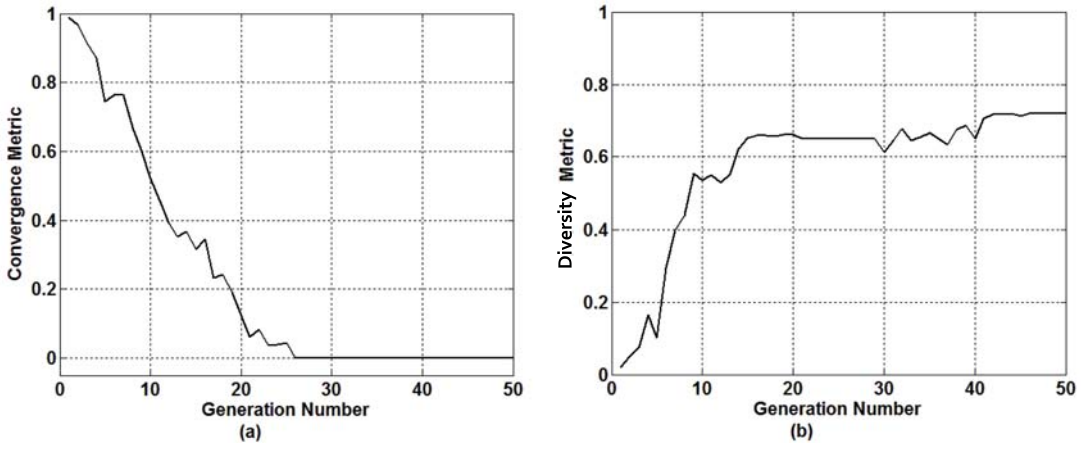

FIGURE 9. Two metrics for $S2$ (a) convergence (b) diversity

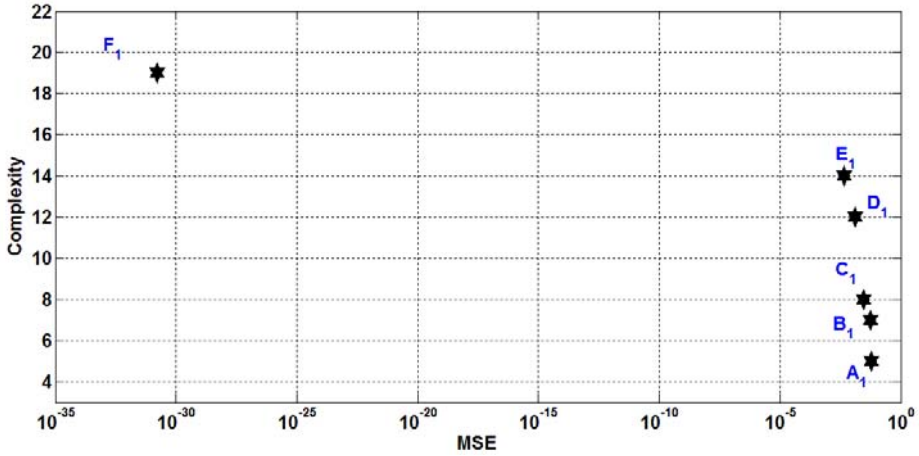

FIGURE 10. Pareto optimal front (last generation) for $S1$

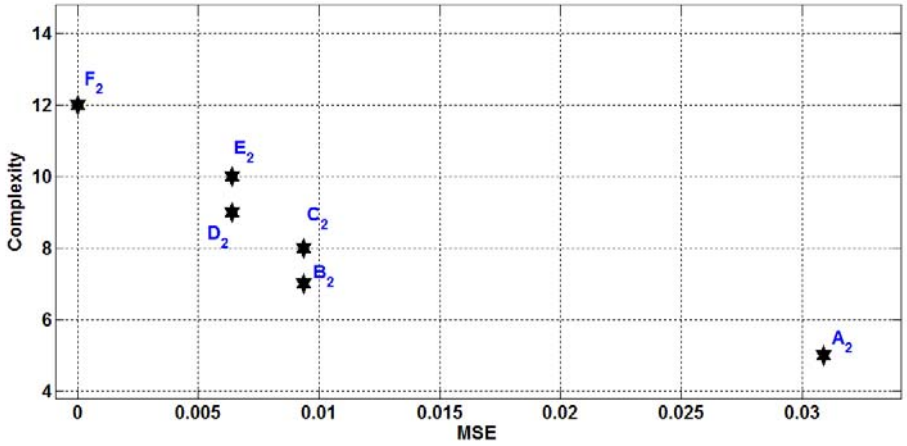

FIGURE 11. Pareto optimal front (last generation) for $S2$

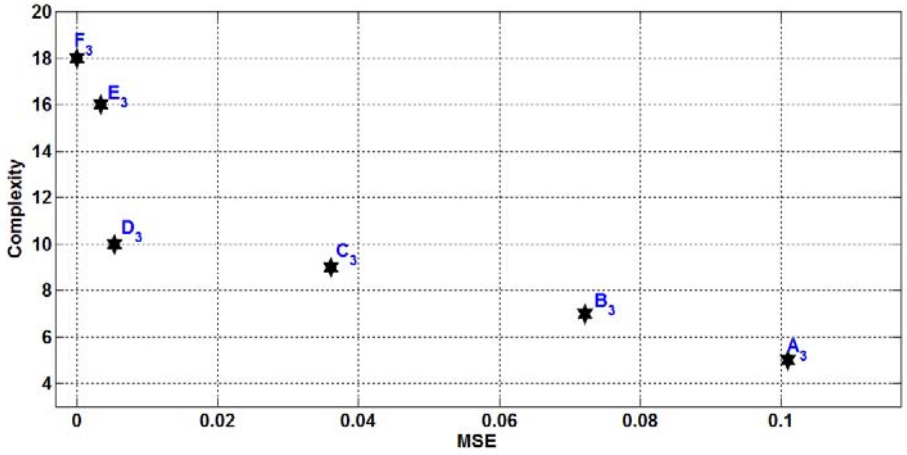

FIGURE 12. Pareto optimal front (last generation) for $S3$

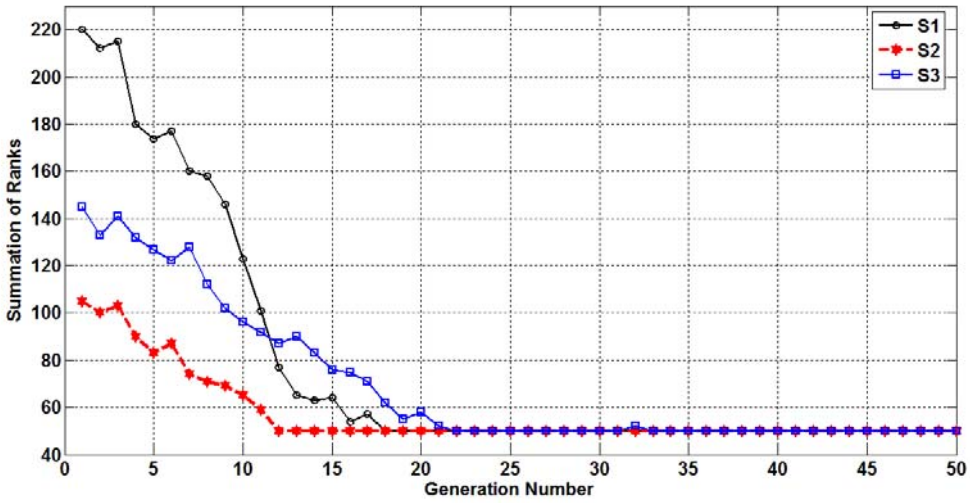

FIGURE 13. The convergence process of summation of rank for $S1$ till $S3$

TABLE 4. Details of the solutions in Pareto optimal front for Figure 10 for $S1$

	Simulated Model	Non-dominated Solutions in Pareto optimal front					
	$S1$	A_1	B_1	C_1	D_1	E_1	F_1
$[n_u\ n_y\ n_l]$	[3 3 2]	[1 1 1]	[1 2 1]	[3 1 1]	[3 1 2]	[3 2 2]	[3 3 2]
Complexity	15	5	7	8	12	14	19
MSE (test set)	----	6.22×10^{-2}	5.48×10^{-2}	2.85×10^{-2}	1.33×10^{-2}	4.55×10^{-3}	1.63×10^{-31}
Lags		Coefficients					
$y(t\text{-}1)$	0.95	0.728	0.839	0.946	0.936	0.942	0.950
$y(t\text{-}2)$	----		-0.160			----	----
$y(t\text{-}3)$							10^{-16}
$u(t\text{-}1)$	0.7	0.680	0.695	0.690	0.686	0.691	0.700
$u(t\text{-}2)$	----			----	----	----	----
$u(t\text{-}3)$	-0.5			-0.608	-0.513	-0.498	-0.500
$y(t\text{-}1)\ y(t\text{-}1)$	----				----	----	----
$y(t\text{-}1)\ y(t\text{-}2)$	----					----	10^{-16}
$y(t\text{-}1)\ y(t\text{-}3)$	----						----
$y(t\text{-}2)\ y(t\text{-}2)$	----					----	----
$y(t\text{-}2)\ y(t\text{-}3)$	----						----
$y(t\text{-}3)\ y(t\text{-}3)$	----						----
$y(t\text{-}1)\ u(t\text{-}1)$	----				----	----	----
$y(t\text{-}1)\ u(t\text{-}2)$	----				----	----	----
$y(t\text{-}1)\ u(t\text{-}3)$	----				0.305	----	10^{-17}
$y(t\text{-}2)\ u(t\text{-}1)$	----					----	10^{-16}
$y(t\text{-}2)\ u(t\text{-}2)$	----					----	----
$y(t\text{-}2)\ u(t\text{-}3)$	----					-0.356	----
$y(t\text{-}3)\ u(t\text{-}1)$	----						10^{-16}
$y(t\text{-}3)\ u(t\text{-}2)$	----						10^{-16}
$y(t\text{-}3)\ u(t\text{-}3)$	-0.37						-0.37
$u(t\text{-}1)\ u(t\text{-}1)$	----				----	----	10^{-16}
$u(t\text{-}1)\ u(t\text{-}2)$	----				----	----	----
$u(t\text{-}1)\ u(t\text{-}3)$	----				----	----	----
$u(t\text{-}2)\ u(t\text{-}2)$	----				----	0.002	----
$u(t\text{-}2)\ u(t\text{-}3)$	----				0.215	-0.007	----
$u(t\text{-}3)\ u(t\text{-}3)$	----				0.228	0.254	----
No. of repetition	----	13	14	9	8	5	1

not have rank one and consequently are not the Pareto optimal front members; (b) some solutions have been repeated several times. To investigate that it is the latter, former or both reason, the graphs of summation of ranks versus the number of generation are plotted as in Figure 13. The figure clearly shows that in all cases the graphs tend to 50 (population size) after some generations indicating the existence of repetitive solutions in the Pareto optimal front.

Table 4 shows the details of the solutions in Pareto optimal front given in Figure 10. The number of repetition of the six solutions, A_1 to F_1, is indicated in the table.

All non-dominated solution A_1 to F_1 can be a solution for the generated data by 13. Solution A_1 is better than F_1 in complexity, but is worst in MSE. Since the data set was generated by a simulated model, and the model has been known, trade-off is not necessary. Thus, the more important is presence of the expected solution (simulated model) among non-dominated solutions in Pareto optimal front. This strategy is applied for another simulated example $S2$ and $S3$. Solution F_1 with most complexity and MSE ≈ 0, presents the expected model of $S1$. It contains seven terms more as compared with $S1$, but the coefficients are power of 10^{-16}, 10^{-17} and 10^{-18}. Consequently, F_1 is the expected result that represents $S1$.

In Table 4, the shaded cells indicate that the proposed solution with the given n_u, n_y and n_l, does not include the terms in column 1 and "----" shows that essentially the model may include this term, but the sub-optimizer does not choose these due to the selection of the best structure. This is also applicable to the tables in other case studies.

System $S2$ is a nonlinear polynomial NARX like $S1$, but with degree of non-linearity of 3. Table 5 indicates the details of non-dominated solutions in Pareto optimal front for

TABLE 5. Details of the solutions in Pareto optimal front of Figure 11 for $S2$

	Simulated Model	Non-dominated solutions in Pareto optimal front					
	$S2$	A_2	B_2	C_2	D_2	E_2	F_2
$[n_u\ n_y\ n_l]$	[2 1 3]	[1 1 1]	[2 1 1]	[3 1 1]	[2 1 2]	[3 1 2]	[2 1 3]
Complexity	10	5	7	8	9	10	12
MSE (test set)	----	3.09×10^{-2}	9.37×10^{-3}	9.35×10^{-3}	6.42×10^{-3}	6.41×10^{-3}	8.05×10^{-33}
Lags		Coefficients					
$y(t\text{-}1)$	0.5	0.740	0.526	0.528	0.517	0.517	0.500
$u(t\text{-}1)$	----	0.312	0.303	0.302	0.297	0.297	----
$u(t\text{-}2)$	0.3		0.292	0.292	0.295	0.294	0.300
$u(t\text{-}3)$				----		----	
$y(t\text{-}1)\ y(t\text{-}1)$	----				----	----	----
$y(t\text{-}1)\ u(t\text{-}1)$	0.3				0.282	0.282	0.300
$y(t\text{-}1)\ u(t\text{-}2)$	----				----	----	----
$u(t\text{-}1)\ u(t\text{-}1)$	----				----	----	10^{-17}
$u(t\text{-}1)\ u(t\text{-}2)$	----				----	----	----
$u(t\text{-}2)\ u(t\text{-}2)$	----				----	----	----
$y(t\text{-}1)\ y(t\text{-}1)\ y(t\text{-}1)$	----						----
$y(t\text{-}1)\ y(t\text{-}1)\ u(t\text{-}1)$	----						----
$y(t\text{-}1)\ y(t\text{-}1)\ u(t\text{-}2)$	----						----
$y(t\text{-}1)\ u(t\text{-}1)\ u(t\text{-}1)$	----						10^{-19}
$y(t\text{-}1)\ u(t\text{-}1)\ u(t\text{-}2)$	----						----
$y(t\text{-}1)\ u(t\text{-}2)\ u(t\text{-}2)$	----						----
$u(t\text{-}1)\ u(t\text{-}1)\ u(t\text{-}1)$	0.5						0.500
$u(t\text{-}1)\ u(t\text{-}1)\ u(t\text{-}2)$	----						----
$u(t\text{-}1)\ u(t\text{-}2)\ u(t\text{-}2)$	----						----
$u(t\text{-}2)\ u(t\text{-}2)\ u(t\text{-}2)$	----						----
No. of repetition	----	15	11	8	10	4	2

TABLE 6. Details of the solutions in Pareto optimal front of Figure 12 for $S3$

	Simulated Model	Non-dominated solutions in Pareto optimal front					
	$S3$	A_3	B_3	C_3	D_3	E_3	F_3
$[n_u\ n_y\ n_l]$	[2 2 3]	[1 1 1]	[1 2 1]	[1 2 2]	[2 2 2]	[2 3 2]	[2 2 3]
Complexity	12	5	7	9	10	16	18
MSE (test set)	----	1.01 $\times10^{-1}$	7.21 $\times10^{-2}$	3.61 $\times10^{-2}$	5.40 $\times10^{-3}$	3.45 $\times10^{-3}$	4.67 $\times10^{-32}$
Lags	Coefficients						
$y(t-1)$	----	0.179	0.269	----	0.092	----	----
$y(t-2)$	-0.5	0.037	-0.498	-0.660	-0.605	-0.659	0.500
$u(t-1)$	----		0.056	----	----	----	----
$u(t-2)$					----	----	10^{-17}
$y(t-1)\ y(t-1)$	----			0.423	----	0.138	----
$y(t-1)\ y(t-2)$	----			----	----	----	----
$y(t-2)\ y(t-2)$	----			0.047	----	-0.0004	----
$y(t-1)\ u(t-1)$	0.7			0.705	0.708	0.699	0.700
$y(t-1)\ u(t-2)$	----				----	-0.165	----
$y(t-2)\ u(t-1)$	----			----	----	----	----
$y(t-2)\ u(t-2)$	----				----	0.111	----
$u(t-1)\ u(t-1)$	----			0.248	----	0.002	----
$u(t-1)\ u(t-2)$	----				----	0.113	----
$u(t-2)\ u(t-2)$	0.6				0.563	0.553	0.600
$y(t-1)\ y(t-1)\ y(t-1)$	0.2						0.200
$y(t-1)\ y(t-1)\ y(t-2)$	----						----
$y(t-1)\ y(t-2)\ y(t-2)$	----						----
$y(t-2)\ y(t-2)\ y(t-2)$	----						10^{-17}
$y(t-1)\ y(t-1)\ u(t-1)$	----						----
$y(t-1)\ y(t-1)\ u(t-2)$	----						10^{-18}
$y(t-1)\ y(t-2)\ u(t-1)$	----						----
$y(t-1)\ y(t-2)\ u(t-2)$	----						----
$y(t-2)\ y(t-2)\ u(t-1)$	----						10^{-19}
$y(t-2)\ y(t-2)\ u(t-2)$	----						----
$y(t-1)\ u(t-1)\ u(t-1)$	----						----
$y(t-1)\ u(t-1)\ u(t-2)$	----						----
$y(t-1)\ u(t-2)\ u(t-2)$	----						----
$y(t-2)\ u(t-1)\ u(t-1)$	----						10^{-17}
$y(t-2)\ u(t-1)\ u(t-2)$	----						----
$y(t-2)\ u(t-2)\ u(t-2)$	0.7						0.700
$u(t-1)\ u(t-1)\ u(t-1)$	----						----
$u(t-1)\ u(t-1)\ u(t-2)$	----						----
$u(t-1)\ u(t-2)\ u(t-2)$	----						10^{-17}
$u(t-2)\ u(t-2)\ u(t-2)$	----						----
No. of repetition	----	16	9	10	10	4	1

Figure 11. Solution F_2 contains two more terms $y(t-1)u^2(t-1)$ and $u^2(t-1)$ compared with the expected solution. Since their coefficients are almost zero, thus they are non-effective terms. Consequently the proposed algorithm could identify the exact solution with correct terms and coefficients (F_2).

As shown in Figure 12 for $S3$, six solutions of A_3 to F_3 have been repeated several times, which are indicated in Table 6. The Pareto optimal front started with the solution with minimum complexity and maximum error, A_3, and ended with the solution with maximum complexity and minimum error. The last solution can be the expected solution (F_3). Solution F_3 is the expected solution for $S3$, if six terms $u(t-2)$, $y^3(t-2)$, $y^2(t-1)$ $u(t-2)$, $y^2(t-2)$ $u(t-1)$, $y(t-2)$ $u^2(t-1)$, $u(t-1)$ $u^2(t-2)$ have not been considered because of their worthless coefficients. Consequently, model F_3 is the expected result and represents $S3$.

6.3. **Real gas furnace process.** The results for previous section for simulated examples showed the effectiveness of the proposed algorithm. In this section, the algorithm is applied to model a real process, Box-Jenkins Gas furnace process ($S4$). The evolution of multi-objective optimization and convergence of two goals for both processes are shown in Figure 14. The graphs of the convergence metric tends to zero as expected and the diversity metric increases with the generation number to form the uniform Pareto optimal front.

Figure 15 shows the summation of ranks in the last generation of both processes. Since the graphs converge to the population size of 50 and the Pareto optimal front, as shown in Figure 16 contains less than 50 points, thus there are some repetitive solutions. After performing the multi-objective optimization, the Pareto optimal front of $S4$ presents seven non-dominated solutions, as shown in Table 7, that are not superior or inferior to each other. Trade-off among them can help designer to choose one as the final solution. Repetitive solutions in the Pareto optimal front reduce the variety of solutions, and thus designer can trade-off among fewer candidates.

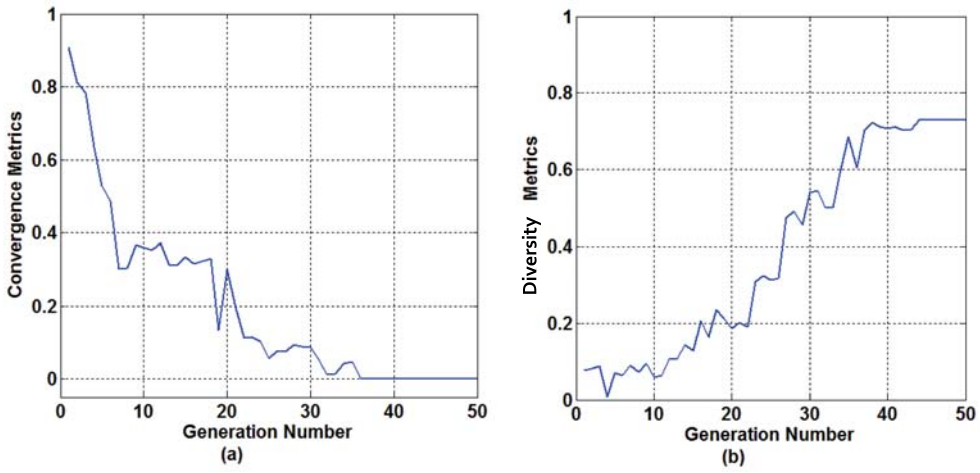

FIGURE 14. Two metrics for $S4$, (a) convergence (b) diversity

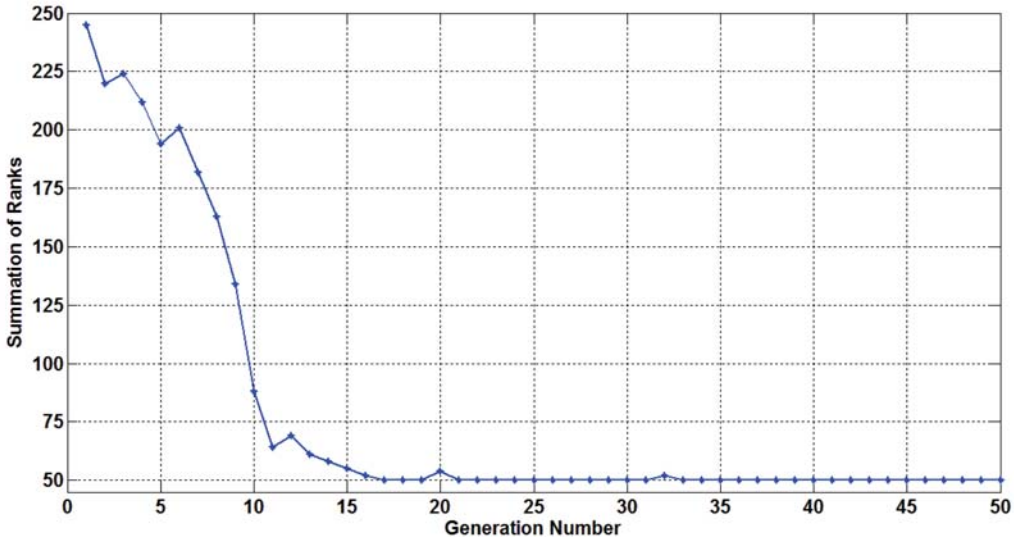

FIGURE 15. The convergence process of summation of rank for $S4$

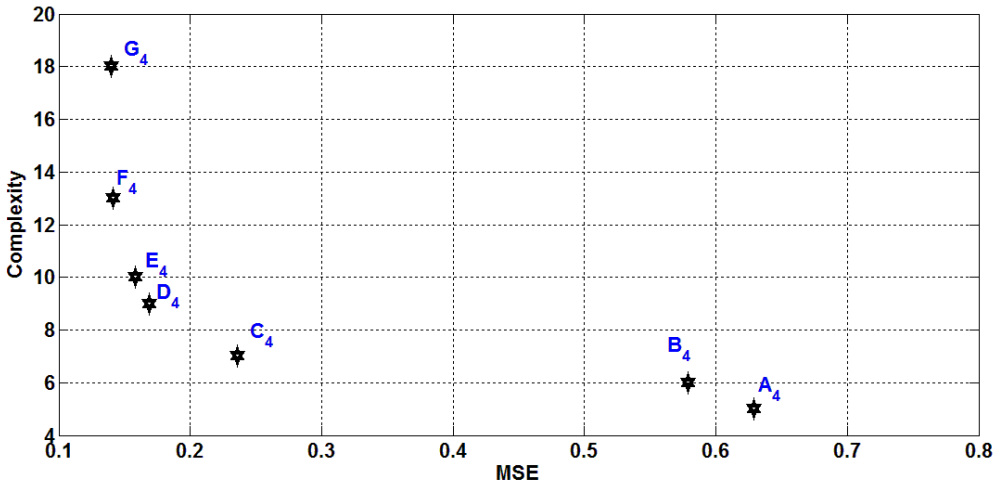

FIGURE 16. Pareto optimal front (last generation) for $S4$

TABLE 7. Details of the solutions in the Pareto optimal front of Figure 16

	Non-dominated solutions in Pareto optimal front						
	A_4	B_4	C_4	D_4	E_4	F_4	G_4
$[n_u\ n_y\ n_l]$	[1 1 1]	[2 1 1]	[1 2 1]	[1 3 1]	[1 4 1]	[1 3 2]	[1 3 2]
Complexity	**5**	**6**	**7**	**9**	**10**	**13**	**18**
MSE (test set)	**0.629**	**0.579**	**0.236**	**0.169**	**0.158**	**0.141**	**0.140**
Lags	**Coefficients**						
$y(t\text{-}1)$	1.000	1.000	1.743	2.217	2.095	1.809	1.081
$y(t\text{-}2)$			-0.741	-1.753	-1.333	-0.728	----
$y(t\text{-}3)$				0.536	----	----	----
$y(t\text{-}4)$					0.238		
$u(t\text{-}1)$	-0.381	----	-0.154	-0.120	-0.127	----	-0.888
$u(t\text{-}2)$		-0.408					
$y(t\text{-}1)\,y(t\text{-}1)$						----	0.037
$y(t\text{-}1)\,y(t\text{-}2)$						----	0.047
$y(t\text{-}1)\,y(t\text{-}3)$						----	-0.107
$y(t\text{-}2)\,y(t\text{-}2)$						0.081	0.210
$y(t\text{-}2)\,y(t\text{-}3)$						-0.005	-0.176
$y(t\text{-}3)\,y(t\text{-}3)$						0.004	0.284
$y(t\text{-}1)\,u(t\text{-}1)$						-0.187	-0.357
$y(t\text{-}2)\,u(t\text{-}1)$						----	-0.086
$y(t\text{-}3)\,u(t\text{-}1)$						0.100	0.158
$u(t\text{-}1)\,u(t\text{-}1)$						----	0.039
No. of repetition	11	8	10	8	7	3	3

Solution A_4 has a simplest structure as compared with others, but the error is about 0.629 and is not acceptable. Anyway, mathematically this is the starting point of Pareto optimal front since it has minimum complexity. Increasing the complexity from A_4 to G_4 the MSE gets better. The number of terms in G_4 is much more than F_4, while MSE is 0.001 less than solution F_4. Thus, F_4 can be a better model for this system. Solution F_4 is a polynomial with degree of nonlinearity of two and seven terms. For further investigation, the correlation tests for F_4 are shown in Figure 17. The correlation tests show that all functions fall within the 95% confidence bands.

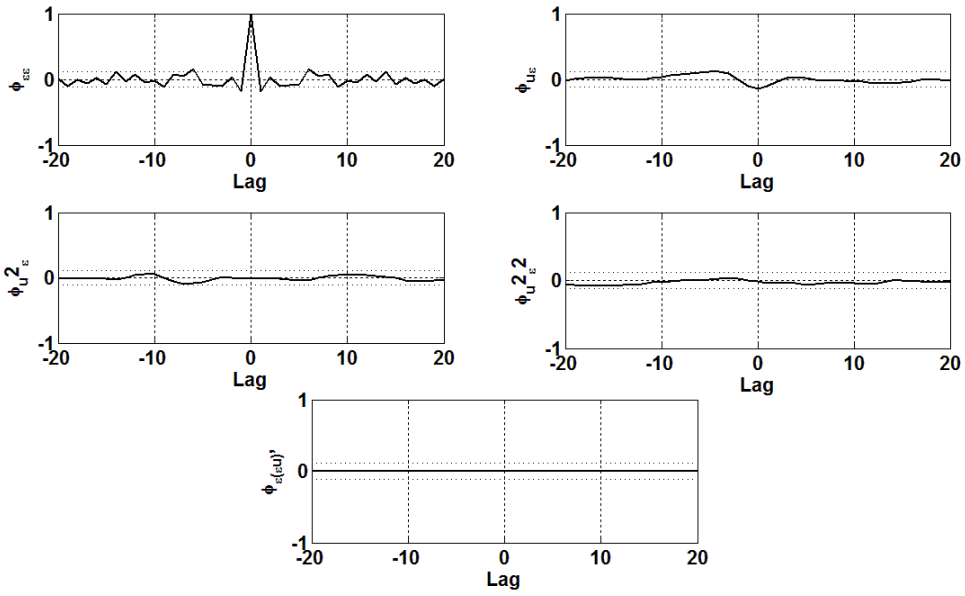

FIGURE 17. Correlation test for solution F_4 of $S4$

6.4. **Comparison study.** As mentioned earlier the clustered crowding distance was used to improve the diversity preservation of non-dominated solutions in NSGA-II for identification of dynamic systems. To compare the performance of NSGA-II using crowding distance to proposed CCD on multi-objective optimization of dynamic system structure, three systems $S1$, $S2$ and $S3$ are considered. Figures 18 to 20 show the convergence and diversity performance of both methods. In all cases the graphs for CD converge to the low values, while using CCD gives better diversity.

Figures 21 illustrate the Pareto optimal front and non-dominated solutions for $S3$. As shown in Figure 21(b) some non-dominated solutions have not been selected by the algorithm and only three solutions have been chosen.

The above results show the effectiveness of using CCD to maintain the diversity of non-dominated solutions in multi-objective optimization of the problems with discrete design variable. With reasonable number of extra solutions as obtained by CCD algorithm,

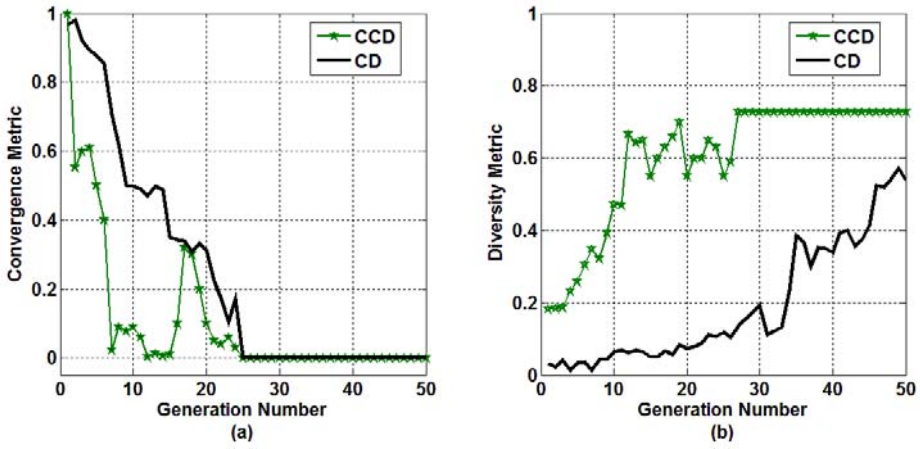

FIGURE 18. Two metrics for $S1$ using CD and CCD (a) convergence (b) diversity

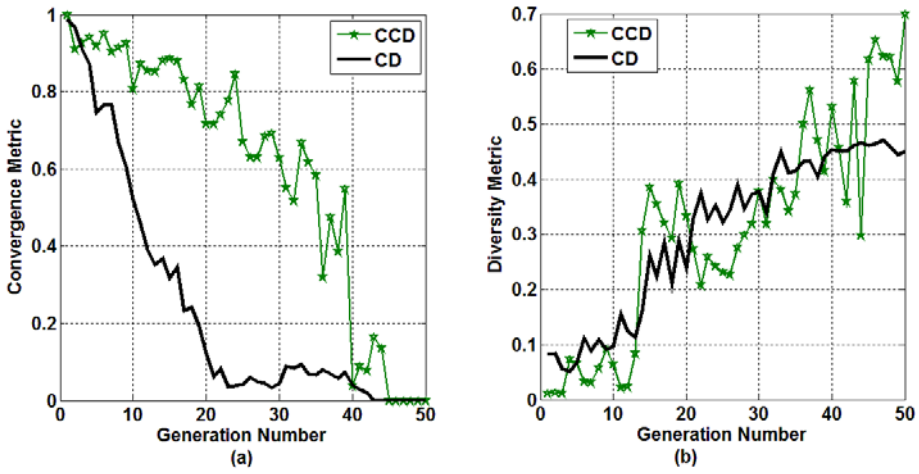

FIGURE 19. Two metrics for $S2$ using CD and CCD (a) convergence (b) diversity

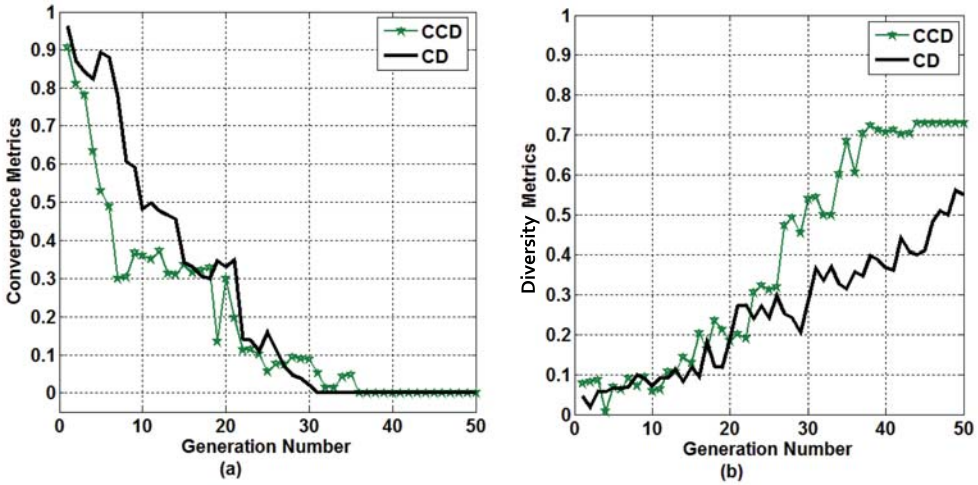

FIGURE 20. Two metrics for $S3$ using CD and CCD (a) convergence (b) diversity

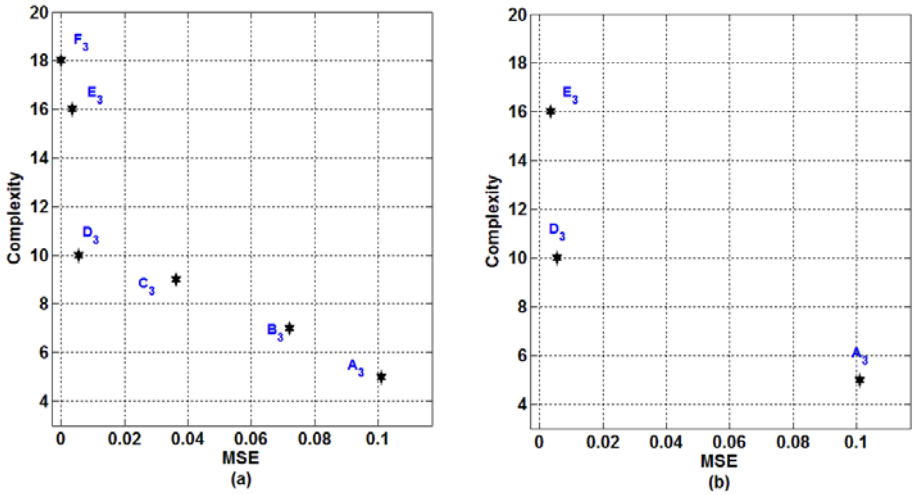

FIGURE 21. Pareto optimal front of $S3$ using (a) CCD and (b) CD

designer would be able to have more choice in selecting the final model that fits rigorous validation test such as correlation tests.

7. **Conclusion.** A new clustered crowding distance (CDD) has been developed based on elitist non-dominated sorting genetic algorithm (NSGA-II) to optimize NARX model structure. The proposed method has been shown to be effective in identifying the correct structure of the system using two objective functions for three simulated polynomial NARX model with different nonlinearity and for a real process data set. There is more than one possible solution in multi-objective optimization problems. Therefore, trade-off among the possible solutions in Pareto optimal and correlation tests can help designer to select the final solution. The results show that CCD method has better performance compare with initial crowding distance algorithm. This multi-objective optimization method finds the correct terms of NARZ model structure without any manual setting in the number of input and output lags and non-linearity order. However, multi-objective optimization needs large computational load to obtain the optimal solutions.

Acknowledgment. This work is supported by Universiti Teknologi Malaysia (UTM) and Ministry of Higher Education of Malaysia under GUP Flagship 00G17.

REFERENCES

[1] M. Korenberg, S. A. Billings, Y. P. Liu and P. J. McIlroy, Orthogonal parameter estimation algorithm for non-linear stochastic systems, *International Journal of Control*, vol.48, no.1, pp.193-210, 1988.

[2] S. Chen, S. A. Billings and W. Luo, Orthogonal least square methods and their application to non-linear system identification, *International Journal of Control*, vol.50, no.5, pp.1873-1896, 1989.

[3] S. A. Billings, S. Chen and R. J. Backhouse, The identification of linear and nonlinear model of a turbocharged automotive diesel engine, *Mechanical Systems and Signal Processing*, vol.3, no.2, pp.123-142, 1989.

[4] S. Chen, C. F. N. Cowan and P. M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, *IEEE Trans. on Neural Networks*, vol.2, pp.302-309, 1991.

[5] G. L. Zheng and S. A. Billings, Qualitative validation and generalization in non-linear system identification, *International Journal of Control*, vol.72, no.17, pp.1592-1608, 1999.

[6] B. McKay, M. Willis and G. Barton, Steady-state modeling of chemical process systems using genetic programming, *Computers Chem. Engng.*, vol.21, no.9, pp.981-996, 1997.

[7] K. Rodriquez-Vazquez and P. J. Fleming, A genetic programming/NARMAX approach to nonlinear system identification, *Genetic Algorithms in Engineering Systems: Innovations and Application*, pp.409-414, 1997.

[8] G. J. Gray, T. Weinbrenner, D. J. Murray-Smith, Y. Li and K. C. Sharman, Issues in nonlinear model structure using genetic programming, *Genetic Algorithms in Engineering Systems: Innovations and Application*, vol.446, pp.308-313, 1997.

[9] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company Inc., MA, 1989.

[10] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, 3rd Edition, Springer Verlag, Berlin, 1999.

[11] K. Kristinsson and G. A. Dumont, System identification and control using genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, vol.22, no.5, pp.1033-1046, 1992.

[12] Z. Zibo and F. Naghdy, Application of genetic algorithms to system identification, *IEEE International Conference on Evolutionary Computation*, vol.2, pp.777-787, 1995.

[13] I. K. Jeong and J. J. Lee, Adaptive simulated annealing genetic algorithm (ASAGA) for system identification, *Engineering Application in Artificial Intelligence*, vol.9, pp.523-532, 1996.

[14] A. F. Sheta and K. De Jong, Parameter estimation of nonlinear systems in noisy environments using genetic algorithms, *Proc. of the IEEE International Symposium on Intelligent Control*, Dearborn, MI, pp.360-365, 1996.

[15] X. Hong and S. A. Billings, Parameter estimation based on stacked regression on evolutionary algorithms, *IEE Proc. of Control Theory and Application*, vol.146, no.5, pp.406-414, 1999.

[16] Y. Liu, L. Ma and J. Zhang, GA/SA/TS hybrid algorithms for reactive power optimization, *Power Engineering Society Summer Meeting*, Seatle, Washington, pp.245-249, 2000.
[17] K. S. Tang, K. F. Man, S. Kwong and Q. He, Genetic algorithms and their applications, *IEEE Signal Processing Magazine*, pp.27-37, 1996.
[18] N. Chaiyaratana and A. M. S. Zalzala, Recent developments in evolutionary and genetic algorithms: Theory and application, *Conference on Genetic Algorithms in Engineering Systems: Innovation and Applications*, vol.446, pp.270-277, 1997.
[19] D. Beasley, Possible applications of evolutionary computation, in *Evolutionary Computation 1: Basic Algorithms and Operators*, T. Back, D. B. Fogel and T. Michalewics (eds.), Bristol, Institute of Physics, 2000.
[20] R. Ahmad, H. Jamaluddin and M. A. Hussain, Model structure selection for a discrete-time non-linear system using genetic algorithm, *Proc. of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol.218, no.12, pp.85-98, 2004.
[21] M. F. A. Samad, *Evolutionary Computation for Model Structure Selection in System Identification*, Ph.D. Thesis, Universiti Teknologi Malaysia, 2009.
[22] L. Yao and C.-C. Lin, On a genetic algorithm based scheduled fuzzy PID controller, *International Journal of Innovative Computing, Information and Control*, vol.5, no.10(B), pp.3593-3602, 2009.
[23] H. Jamaluddin, M. F. A. Samad, R. Ahmad and M. S. Yaacob, Optimum grouping in a modified genetic algorithm for discrete-time nonlinear system identification, *Proc. of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol.221, no.7, pp.975-989, 2007.
[24] R. S. Sexton, R. E. Dorsey and J. D. Johnson, Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing, *European Journal of Operational Research*, vol.114, pp.589-601, 1999.
[25] E. Zitzler, K. Deb and L. Thiele, Comparison of multi-objective evolutionary algorithm: Empirical results, *Evolutionary Computation*, vol.8, pp.173-195, 2000.
[26] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*, John Wiley, Chichester, 2001.
[27] N. Srinivas and K. Deb, Multi-objective optimization using non-dominated sorting in genetic algorithms, *Evolutionary Computation*, vol.2, no.3, pp.221-248, 1994.
[28] L. Ljung, *System Identification Theory for the User*, Prentice-Hall Inc., New Jersey, 1999.
[29] S. A. Billings and W. S. F. Voon, Correlation based model validity tests for non-linear models, *International Journal of Control*, vol.44, no.1, pp.235-244, 1986.
[30] K. Z. Mao and S. A. Billings, Algorithms for minimal model structure detection in nonlinear dynamic system identification, *International Journal of Control*, vol.68, no.2, pp.311-330, 1997.
[31] G. E. P. Box, G. M. Jenkins and G. C. Reinsel, *Time Series Analysis Forecasting and Control*, Prentice-Hall Inc., New Jersey, 1994.
[32] J. J. Grefenstette, Optimization of control parameters for genetic algorithms, *IEEE Transactions on Systems, Man, and Cybernetics*, vol.16, no.1, pp.122-128, 1986.
[33] R. Qingsheng, Z. Jin and Y. Zhongxing, Analysis of common genetic operators, *Proc. of the 5th International Conference on Signal Processing*, Beijing, China, pp.1684-1687, 2000.