

PERTURBATION PARAMETERS TUNING OF MULTI-OBJECTIVE OPTIMIZATION DIFFERENTIAL EVOLUTION AND ITS APPLICATION TO DYNAMIC SYSTEM MODELING

Article history

Received

10 Feb 2015

Received in revised form

3 March 2015

Accepted

25 March 2015

*Corresponding author

zakimizakaria@unimap.edu.my

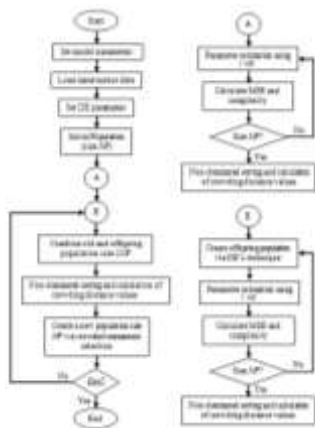
Mohd Zakimi Zakaria,^{a,*} Hishamuddin Jamaluddin,^b Robiah Ahmad,^c Azmi Harun,^a Radhwan Hussin,^a Ahmad Nabil Mohd Khalil,^a Muhammad Khairy Md Naim,^a Ahmad Faizal Annuar,^a

^aSchool of Manufacturing Engineering, Universiti Malaysia Perlis, Ulu Pauh Main Campus, 02600 Arau, Perlis, Malaysia

^bDepartment of Applied Mechanics, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

^cUTM Razak School, Universiti Teknologi Malaysia, 54100 Jalan Semarak, Kuala Lumpur, Malaysia

Graphical abstract



Abstract

This paper presents perturbation parameters for tuning of multi-objective optimization differential evolution and its application to dynamic system modeling. The perturbation of the proposed algorithm was composed of crossover and mutation operators. Initially, a set of parameter values was tuned vigorously by executing multiple runs of algorithm for each proposed parameter variation. A set of values for crossover and mutation rates were proposed in executing the algorithm for model structure selection in dynamic system modeling. The model structure selection was one of the procedures in the system identification technique. Most researchers focused on the problem in selecting the parsimony model as the best represented the dynamic systems. Therefore, this problem needed two objective functions to overcome it, i.e. minimum predictive error and model complexity. One of the main problems in identification of dynamic systems is to select the minimal model from the huge possible models that need to be considered. Hence, the important concepts in selecting good and adequate model used in the proposed algorithm were elaborated, including the implementation of the algorithm for modeling dynamic systems. Besides, the results showed that multi-objective optimization differential evolution performed better with tuned perturbation parameters.

Keywords: Model structure selection; System identification; Multi-objective optimization; NSGA-II; Differential evolution

Abstrak

Kertas kerja ini membentangkan penalaan pengubahan parameter terhadap algoritma multi-objective optimization differential evolution dan penggunaannya terhadap pemodelan sistem dinamik. Pengubahan algoritma yang dicadangkan terdiri daripada pengendali penyilangan dan mutasi. Pada mulanya, satu set nilai-nilai parameter yang ditala sebaiknya dengan melaksanakan beberapa simulasi algoritma untuk setiap variasi parameter yang dicadangkan. Nilai set untuk kadar penyilangan dan mutasi telah dicadangkan dalam melaksanakan algoritma untuk pemilihan struktur model dalam pemodelan sistem dinamik. Pemilihan struktur model adalah salah satu prosedur dalam teknik pengenalan sistem. Kebanyakan penyelidik memberi tumpuan kepada masalah

dalam memilih model teringan yang terbaik bagi mewakili sistem dinamik. Oleh itu, masalah ini memerlukan dua fungsi objektif untuk mengatasinya iaitu meminimumkan ramalan ralat dan model kerumitan. Salah satu masalah utama dalam mengenal pasti sistem dinamik adalah untuk memilih model yang minimum daripada kemungkinan model yang besar yang perlu dipertimbangkan. Konsep-konsep penting dalam memilih model yang baik dan mencukupi digunakan dalam algoritma yang dicadangkan dan diuraikan, termasuk pelaksanaan algoritma untuk model sistem dinamik. Keputusan menunjukkan bahawa algoritma multi-objective optimization differential evolution memberikan prestasi yang lebih baik dengan penalaan pengubahan parameter.

Kata kunci: Pemilihan struktur model; Pengenalan sistem; Pengoptimuman multi-objektif; NSGA-II; Differential evolution

© 2015 Penerbit UTM Press. All rights reserved

1.0 INTRODUCTION

System identification has been studied and applied in many fields, such as engineering, biology, chemical, economics, agricultural, ecology, and others. Identifying the dynamic systems require a set of input-output data of the system, a type of class models, parameter estimation algorithm, and finally, model validation [1]. One of the main problems of system identification is to find an optimal model, which is the simplest model that can adequately represent the dynamic systems or a parsimonious model. Basically, there are two major sub-problems in system identification [2]: (1) to determine the model structure describing the functional relationship between the input and the output variables of the dynamic system; and (2) to estimate the coefficients or parameters of model that specify the chosen or selected model structure. Besides, Aguirre [3] claims that one of the most challenging problems in non-linear system identification is structure selection.

Model structure selection is for selecting the significant model terms from a redundant dictionary of the candidate regressors to be included in the final model [4]. The number of candidate regressors depends on the setting of model size parameters, such as input and output lags, as well as the degree of non-linearity. In other words, the model terms increase rapidly as the input and the output lags, as well as the degree of non-linearity increased. Therefore, selecting only the best and good model terms to be included in the final model structure becomes challenging. In fact, many techniques and methods have been researched [3, 5-9] in solving model structure selection as one of the system identification problems. The earlier work on determination of subset or model structure selection used the deterministic method. One of the famous deterministic methods for model structure selection is forward regression orthogonal least square (OLS) [8]. However, this method has shown disadvantages in structure determination of NARMAX (Non-linear AutoRegressive Moving Average with eXogenous

inputs) model. The disadvantages of OLS algorithms are [10]: (1) the high computational load increases the number of possible terms and the estimator yields different orthogonalisation path. These lead to a complex calculation and numerically ill-conditioned equation; as well as (2) problem in stopping regression with the value of error reduction ratio (ERR) is reduced gradually. Beside OLS as deterministic method in selection of model structure, Aguirre [3] proposed a model structure selection algorithm called term clusters. The concept used is similar to OLS, which applies ERR criterion as the objective of function for selection of the model terms. The difference is the possible model terms with the similar characteristics that are considered as a group. However, not all possible term groups are represented in a certain model. If a certain cluster in the possible term groups is not required to compose a model, it is said to be spurious and invalid model terms. Although the term clusters seem better in identification of the dynamic system, it still requires an exhaustive search when the number of possible terms is huge.

Recently, differential evolution (DE), one of the EAs introduced by Storn and Price [11], was applied to MOO problems. Feoktistov [12] claims DE is suitable for solving the huge and complex problems with just small and simple algorithm. Thus, many works have implemented the DE to MOO problems [13-16]. One of the earlier works on DE that applied the MOO problems were conducted by Abbas and Sarker [17], in which Pareto DE (PDE) was introduced. The proposed algorithm was compared to SPEA [18] on two tests of MOO problems and it had been found that PDE was more outstanding. Furthermore, DE was extended to MOO problem by [14] calling it Multi-Objective Differential Evolution (MODE). Meanwhile, Babu et al., [14] applied MODE in solving optimization problems in chemical engineering. Industrial adiabatic styrene reactor was considered as a problem to be optimized by considering productivity, selectivity, and yield as the main objectives. The results were compared with NSGA and it showed superiority. Furthermore, Adeyemo and Otieno [13] proposed

Multi-Objective Differential Evolution Algorithm (MDEA) in solving MOO problems in engineering field. The MDEA was tested using well-known test problems that was introduced by Zitzler et al., [19]. The comparison results showed that MDEA outperformed other MOO algorithms.

Based on these research trends on DE, in this study, the applications point was addressed. Although the use of DE had become famous in solving MOO problems, its use in model structure selection had not been studied before. In other words, no previous study had used DE for multi-objective model structure optimization as far as the authors' knowledge. In this paper, DE was adapted to MOO problem using the technique proposed by Deb et al., [20] in order to produce the non-dominated solutions. The proposed algorithm is named Multi-Objective Optimization using Differential Evolution (MOODE). Although some reported works used the same technique [16, 21], the applications addressed were different. Moreover, Peng et al., [16] applied their proposed algorithm called non-dominated sorting DE (NSDE) to the multi-objective optimal model of phasor measurement unit (PMU) placement, while Krink and Paterlini [21] proposed an algorithm named DE for multi-objective portfolio optimization (DEMPO) to solve multi-objective portfolio optimization in investment management.

In this paper, MOODE with tuned perturbation parameters was applied to determine the set of model structures of dynamic systems that had an adequate system from the available input-output data. The special case of NARMAX model called NARX (Non-linear AutoRegressive with eXogenous inputs) model was used to obtain the structure of the system considered. The paper is organized as follows. Section 2 briefly describes the model representation, i.e. the NARX model used. Section 3 describes the proposed algorithm MOODE for selection of model structure. Furthermore, section 4 reports the simulation studies for perturbation parameter tuning for MOODE. The final section, section 5, summarizes the contribution of the study.

2.0 MODEL REPRESENTATION

In a two-component gel, it is easy to modify the molecular structure of either of the two components. Representing the dynamic non-linear system from the acquired input output data needs a type of model representation to be defined. Most non-linear systems are modelled and identified by using mathematical and signal models, block diagram models, and simulation models [22]. In this study, the mathematical and signal models were considered as model representation. A very common polynomial linear discrete-time system model representation is the ARX (AutoRegressive with eXogenous input) model, whereby the system output can be predicted by the

past inputs and outputs of the system [1]. This model is defined as

$$y(t) = C + a_1 y(t-1) + \dots + a_{n_y} y(t-n_y) + b_1 u(t-1) + \dots + b_{n_u} u(t-n_u) + e(t) \quad (1)$$

where the constant, output, input, and noise signal are represented by C , $y(t)$, $u(t)$, and $e(t)$ respectively, while n_y , and n_u represent the output and the input lags respectively. The coefficients of the model are represented by $a_1 \dots a_{n_y}$ and $b_1 \dots b_{n_u}$. The non-linear version for ARX model is called NARX model. Chen and Billings [23] presented a Non-linear AutoRegressive Moving Average with eXogenous inputs (NARMAX) model with the special case as a NARX model. This model provides a wide class of a non-linear model representation. The NARX model can be defined as;

$$y(t) = F^l(C, y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)) + e(t) \quad (2)$$

where $F^l(\cdot)$ is a polynomial non-linear function with l degree of non-linearity. The NARX model is transformed into a linear regression model, represented as

$$y(t) = \sum_{i=1}^M \theta_i \phi_i(t) + e(t), \quad n_y \leq t \leq N \quad (3)$$

where θ_i and $\phi_i(t)$ are unknown coefficients or parameters and non-linear regressors respectively, M is the maximum number of terms of the regressors, and N is the size of data. The maximum number of possible terms, L_t , in the NARX model that expressed in equation (2) can be calculated as [23]

$$L_t = M + 1 \quad (4)$$

where $M = \sum_{i=1}^l n_i$ and

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

For example, a NARX model with $n_y = n_u = 2$ and $l = 2$, as a second degree of non-linearity order of discrete-time system, would obtain 15 terms respectively. Thus, the possible models need to be considered can be calculated as $2^L - 1$, which is 32, 767. This can be stated that an increase in the input output lags, as well as the degree of non-linearity increases the maximum number of terms of NARX model and possible models that need to be searched. Thus, the user defined that the parameters, such as input lags, output lags, and degree of non-linearity, affected the model structure selection. The search space became large and impractical when the larger user defined the parameters used [5]. Therefore, the need of intelligent tools for selecting the significant terms included in the final model became challenging and complicated. Hence, by employing DE as an alternative method in model structure selection, this searching method can tackle the larger search space that is involved in selecting the significant terms. The proposed

algorithm MOODE was assigned in solving system identification problems for model structure selection particularly. The following section elaborates the MOODE in the determination of model structure.

3.0 PROPOSED ALGORITHM

3.1 Differential Evolution

DE is a powerful heuristic method that has yielded promising results in finding global optimization. Storn and Price [11] proposed DE, and designed it with a simple structure, easy to apply, convergence property, highly quality of solution, and robustness. Similar with other EAs, DE evolves the population that represents the candidate solutions by using genetic operators, i.e. mutation and crossover, as well as selection operators. The special issue in DE was compared to other EAs is the creation of mutation vector (so-called chromosome) by calculating the weighted difference vector of two randomly selected individuals, which were scaled down and added to the third randomly selected individual in the population. The mutant vector went through another genetic operator, which was crossover operator in producing a new population. The mutation operator in DE consisted of four strategies or schemes that were proposed by Storn [24]:

$$\text{Scheme DE/rand/1} \\ v_{(i,G+1)} = x_{(r1,G)} + F \cdot (x_{(r2,G)} - x_{(r3,G)}) \quad (5)$$

$$\text{Scheme DE/best/1} \\ v_{(i,G+1)} = x_{(best,G)} + F \cdot (x_{(r1,G)} - x_{(r2,G)}) \quad (6)$$

$$\text{Scheme DE/best/2} \\ v_{(i,G+1)} = x_{(best,G)} + F \cdot (x_{(r1,G)} + x_{(r2,G)} - x_{(r3,G)} - x_{(r4,G)}) \quad (7)$$

$$\text{Scheme DE/rand-to best/1} \\ v_{(i,G+1)} = x_{(i,G)} + \lambda \cdot (x_{(best,G)} - x_{(i,G)}) + F \cdot (x_{(r2,G)} - x_{(r3,G)}) \quad (8)$$

where $v_{(i,G+1)}$ is mutant vector for the next generation at index $i=1,2,3,\dots, NP$; NP is population size; $x_{(r1,G)}$ is target vector for the current generation at random index integer r_1 ; $r_1, r_2, r_3, r_4 \in \{1,2,\dots, NP\}$; F is a real and constant rate $\in [0,2]$, so-called mutation rate, MR , that control the amplification of the differential variation of $(x_{(r2,G)} - x_{(r3,G)})$ [11], whereas $\lambda = F$; $x_{(best,G)}$ is the best vector of the population and $x_{(i,G)}$ is vector at index i at the current generation. The value of F depends on the problem that is to be solved. Resulting in the use of larger values of F is the generated population in becoming more diverse and slower convergence, whereas using the lower values causes faster convergence. The perturbation of vectors in DE was continued by applying the crossover operator. The crossover operator functioned as parameter mixing between

the target vector and mutant vector to yield the trial vector. According to Storn and Price [11], the trial vector is formed as in the following

$$u_{i,G+1} = v_{i,G+1} = \begin{cases} v_{ji,G+1} & \text{if } (\text{rand}(j) \leq CR) \text{ or } j = \text{ranr}(i) \\ x_{ji,G} & \text{if } (\text{rand}(j) > CR) \text{ or } j \neq \text{ranr}(i) \end{cases} \quad (9) \\ , j = 1, 2, \dots, D.$$

where $u_{(ji,G+1)}$ is the trial vector at index j and $i=1,2,3,\dots, NP$; D is the size of vector or called dimensional vector. CR is an abbreviation of crossover rate $\in [0,1]$ that has to be chosen by the user; whereas $\text{rand}(j)$ is the output of a uniform random number generator in the range of $[0,1]$, and $\text{ranr}(i)$ is a randomly chosen index in the range of $[1,2,\dots, D]$ that is taken from the mutant to ensure that the trial vector does not duplicate all the target vector, $x_{(ji,G)}$, in order to get at least one parameter from the mutant vector, $v_{(ji,G+1)}$. The type of this crossover operator is one-point crossover.

After producing the trial vector, the so-called offspring vector, the comparison between the trial vector and the target vector was performed by using the greedy criterion. If the trial vector, $u_{(i,G+1)}$, yields a better objective function value compared to the target vector, $x_{(i,G)}$, then the trial vector is selected; whereas if otherwise, the target vector is retained. The following equation shows the selection process in D

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & \text{if } (f(u_{i,G+1}) < f(x_{i,G})) \\ x_{i,G} & \text{otherwise} \end{cases} \quad (10)$$

where $f(\cdot)$ denotes as an objective function for a particular vector. The selection process is repeated until specified termination criterion is achieved. However, in this study, the proposed algorithm MOODE was not used in this selection process. The MOODE algorithm has its own selection process that is suitable to select a set of solutions instead of single optimal solution. The next sub-section elaborates the MOODE for determination of model structure.

3.2 MOODE

The MOODE algorithm is the combination of DE and NSGA-II. There are two major parts of the proposed algorithm: the first one is partly based on DE and the second one is inspired by the NSGA-II by Deb et al., [20]. The main difference was compared to single-objective optimization of DE [11] to produce a set of possible solutions instead of a single solution. The set of solutions was represented in the points of the Pareto-optimal front. The Pareto-optimal front is a dense set of the selected solutions between two contradicting objective functions, i.e. model predictive error and model complexity that were considered in the selection of model structure. Therefore, the algorithm

was designed to generate the population of possible solutions that has to be scattered along the points of Pareto-optimal front and adequate in both objective functions under consideration. In MOO, this approach is typical for ease of user to select the potential solutions. Details of the implementation of the MOODE for NARX model structure selection, which modelled the dynamic system, are listed in the following:

Step 1: NARX model representation. The model parameters, such as input and output lags, n_u and n_y , as well as degree of nonlinearity, l are specified. Available input-output data set was loaded before creating the regressor of the NARX model. The regressor was formed based on the pair of input-output data, while the size of regressor was based on the specified model parameters that were calculated from equation (4).

Step 2: Parameter setting. The DE parameters, such as population size, NP, crossover rate, CR, differentiation rate (so-called mutation rate), MR, lower boundary, L, upper boundary, H and maximum generation, Gen, were specified, except the value of vector size, D, which was equal to the size of regressor that was previously generated. The values of lower and upper boundaries were set to create the individuals of vector of the population.

Step 3: Create initial population with size NP. Random vectors were generated, so-called chromosomes of the population that were based on the value of lower boundary, L, and upper boundary, H. Let the values of L and H be 0.1 and 0.4, respectively, then a vector of the population was created randomly as [0.1636; 0.1052; 0.2708; 0.3540; 0.3522; 0.2344; 0.3899] where the vector size, D, was seven. This vector represented a possible model structure with the model parameters n_u , n_y , and l were 3, 3, and 1, respectively, which had been in equation (1), the full ARX model could be expressed as

$$y(t) = [C+a]_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) + b_1 u(t-1) + b_2 u(t-2) + b_3 u(t-3) + e(t)$$

where the maximum number of possible terms, L_t , as in equation (4); seven and equaled to D. If the variables, so-called genes of the vector were equal or more than 0.3, it means the particular terms in the full model existed, while if otherwise, the terms were excluded. Therefore, back to the early example of the vector, the model structure could be represented as

$$y(t) = a_3 y(t-3) + b_1 u(t-1) + b_3 u(t-3) + e(t)$$

where only the fourth, fifth, and seventh of vectors were equal or more than 0.3. Thus, the fourth, fifth, and seventh terms in the full model existed in presenting the model structure of dynamic system.

Step 4: Define two objective functions. The two objective functions used in this study were: the first was model predictive error using MSE, OFn1, while the second was model complexity, OFn2. These objective functions were inspired by Zakaria et al., [25]. To calculate the OFn1, which is MSE, the selected term of the identified model would go through the parameter estimation. A simple parameter estimation method was used in this study; least square estimation (LSE) algorithm. The LSE can be expressed as [26]

$$\theta_j = [(\Phi^T \Phi)]^{-1} \cdot \Phi^T Y \quad (11)$$

where θ_j is parameter or coefficient of identified term; Φ is the regressors, and Y is the actual output data. Thus, the OFn1 can be calculated as

$$OFn1 = 1/N \sum_{i=1}^N [(y_i - \hat{y}_i)]^2, \quad i=1,2,\dots,N \quad (12)$$

where $\hat{y}_i = \theta_j \cdot y_i$ is predicted output, and N is the number of data. As for OFn2, the model complexity can be computed as

$$OFn2 = \sum_{i=1}^D newvar_i, \quad i=1,2,\dots,D \quad (13)$$

where the summation of the value of new variables was equal to the model complexity. Both objective functions would be aligned along the variables of the vector, for example: [0.1636; 0.1052; 0.2708; 0.3540; 0.3522; 0.2344; 0.3899; OFn1; OFn2]. This strategy had been useful for the next step.

Step 5: Produce parent population, P_t , with size NP.

Each vector of the population had its own objective functions (OFn1 and OFn2), which had been evaluated in Step 4. Based on these objective functions, each vector of the population was ranked and crowding distance of vectors was calculated. Details on ranking and calculating of crowding distance of solutions are shown in Deb [27]. The vectors of the parent population were sorted, that was based on its rank and the value of crowding distance. This sorting is priority for the number of rank; the first top one (so-called non-dominated solution) was numbered as one, the second top one was numbered as two, and so on. If the ranking of the vectors had been the same; the value of crowding distance was considered to sort them, which was the priority for higher value of crowding distance.

Step 6: Create new vectors. This step was partly based on the DE technique in producing new generation of vectors. From the parent population, three vectors of the population were chosen randomly, which had been evolved to create new vectors through two genetic operators: mutation and crossover. The mutation scheme used in this study is as in

equation (5). The trial vector was generated from the mutant vector that would go through the crossover operator. The boundary of the variable for the vector was confirmed before it went through the evaluation process of the new vectors.

Step 7: Create offspring population, Q_t , with size NP. After the new vectors were produced in the population, Step 4 was repeated in order to define the objective functions for the new vectors. The production of offspring population had been the same as the parent population in Step 5. Each vector of the offspring population was sorted based on its rank and the value of crowding distance.

Step 8: Create new generation of population, P_{t+1} , with size NP. This step was inspired by Deb et al., [20]. The parent population and the offspring population were combined, $R_t = P_t \cup Q_t$, and became size $2NP$. This combined population would be through the non-dominated sorting and calculation of the crowding distance value as in Steps 5 and 7. Set P_{t+1} and vectors of R_t were counted until size NP. This counting was based on rank and the value of crowding distance of each vector of R_t . Lastly, the final population of P_{t+1} was selected by using the crowded tournament selection and Step 6 is taken until the maximum number of generation, Gen, was reached.

Step 9: Illustrate the solutions of a set of possible model structures that were minimized to two objective functions: model predictive error and model complexity (OFn1 and OFn2) by plotting in one graph with OFn2 versus OFn1. The graph shows the trade-off between the objective functions. The set of possible model structures represents as the points of Pareto-optimal front in order to help the user to choose the potential models before going through the validation test.

The flowchart of the NARX model structure determination that was formulated by using MOODE is shown in Figure 1. This flowchart summarizes the details of the implementation of MOODE to select a set of model structures. From this implementation, the significant concepts in selecting good and adequate model in the MOODE were:

- i. the model representation used to represent the behavior of dynamic system;
- ii. the objective functions used to analyze the candidate of model structures and choose a set of potential models;
- iii. the selection algorithm used to select the potential solutions to be considered in the next generation.

The main difference compared to the approach proposed by Loghmanian et al., [28] that used NSGA-II was the generation of new candidate solutions,

which used the operators of DE instead of operators of GA. Furthermore, the model representation used in the MOODE was changed to the real number in the range of [0.1, 0.4] as in Step 3, whereas NSGA-II was used as bit-string representation. The use of real number to represent the model structure seemed more dynamic and suitable for the larger search space. Therefore, this resulted in a better solution for model structure selection.

4.0 RESULTS AND DISCUSSION

The parameter setting for MOO using evolutionary algorithms (MOEAs) was crucial for identifying the best performance of the algorithms. Numerous works for parameter tuning had looked into variation operators in evolutionary algorithms, such as crossover and mutation operator [29]. These parameters are very sensitive in driving the algorithms to the best performance, as well as in finding good results. In MOEAs, parameter tuning was very important before the algorithms were executed. Parameter tuning refers to finding good parameter values, which were fixed at the initialization stage. These parameter values were unchanged while executing the MOEAs [30]. However, many studies have shown that finding good parameter values for the MOEAs is a hard task [29-32]. In fact, most studies showed that different parameter values were needed for different test problems. Therefore, the need of parameter tuning had been important when different test problems were considered. Hence, the parameters of variation operators considered in this study had been based on Storn and Price's [11]. The parameters used in this study are listed in Table 1.

Table 1 Parameters used in the MOODE

Parameters	Values
Population size	50
Maximum generation	100
Crossover rate (CR)	0.9/0.7/0.6/0.5
Mutation rate (MR)	0.5/0.3/0.2/0.1

The parameters (crossover and mutation rate) were studied with four different values for each. For each setup, the algorithms were performed at one hundred runs and the average of both objective functions (OFn1 and OFn2) against the number of generation was plotted. The algorithm investigated was based on the following simulated dynamic systems [33]:

$$\text{Model 1: } y(t) = 0.797y(t-1) - 0.237y(t-3) + 0.441u(t-1) + 0.105y(t-4)u(t-4) + 0.333u(t-3)u(t-5) + e(t)$$

$$\text{Model 2: } y(t) = 0.3u(t-1)u(t-2) - 0.8y(t-2)u^2(t-1) - 0.1u^2(t-1)u(t-2) + e(t)$$

where input $u(t)$ is a random sequence $[-0.5,0.5]$ and $e(t)$ is random white noise $[-0.01,0.01]$. Both models were generated till four hundred data points were generated and used to identify a NARX model adequately representing the data set. The illustration of the input-output of models can be seen in Figures 2 and 3. The values of input and output lag in estimating Model 1 were five and second degree of non-linearity order. Thus, the total candidate terms had been 66

and $2^{66} - 1 = 7.38 \times 10^{19}$, which were the total possible models to choose from. As for Model 2, the values of input and output lag used were two and third degree of non-linearity respectively. Hence, the total possible terms were 35 and $2^{35} - 1 = 3.44 \times 10^{10}$, which was the total number of possible model structures to choose from.

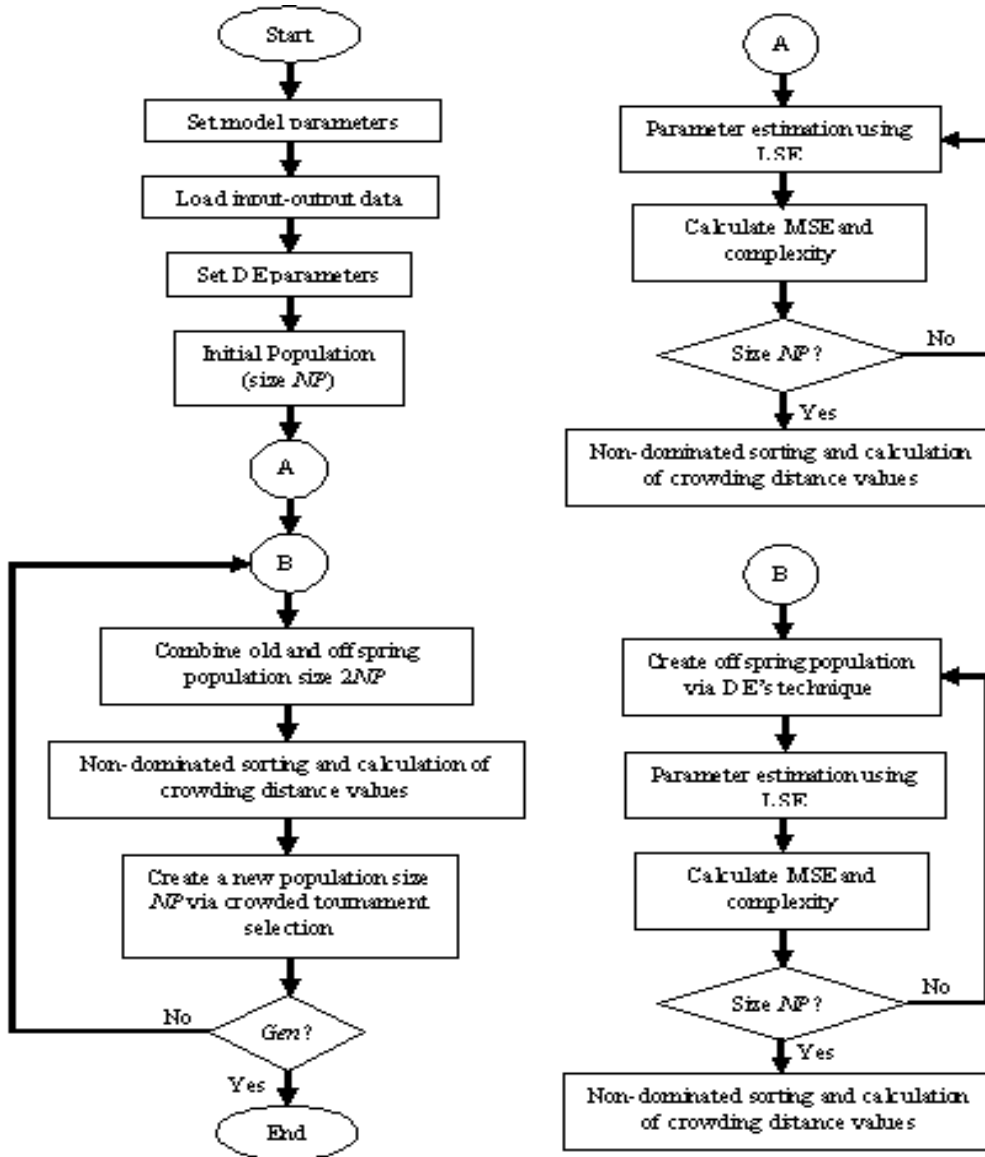


Figure 1 The flowchart of MOODE

The structure of the models is known where Model 1 and Model 2 consisted of five and three terms respectively. Therefore, the proposed algorithm was needed to find at least one correct model structure

from the huge number of possible model structures to choose from.

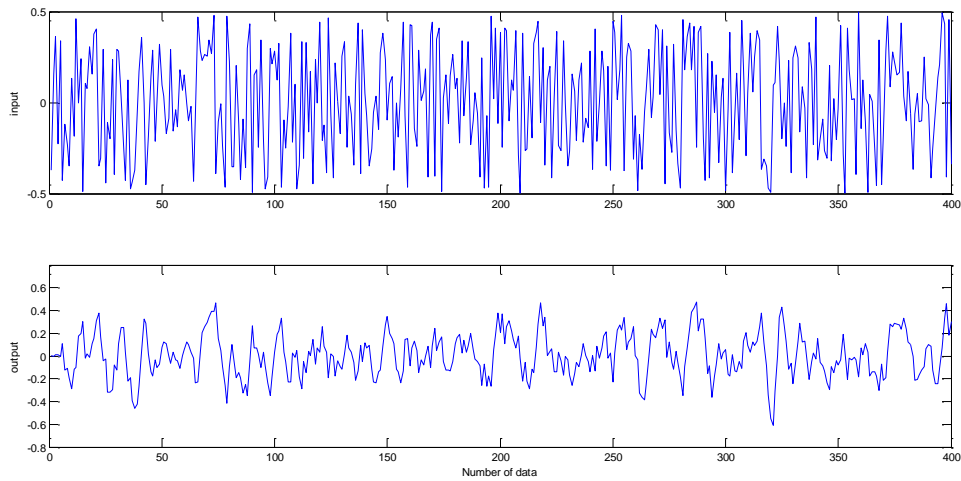


Figure 2 Input and output data for Model 1

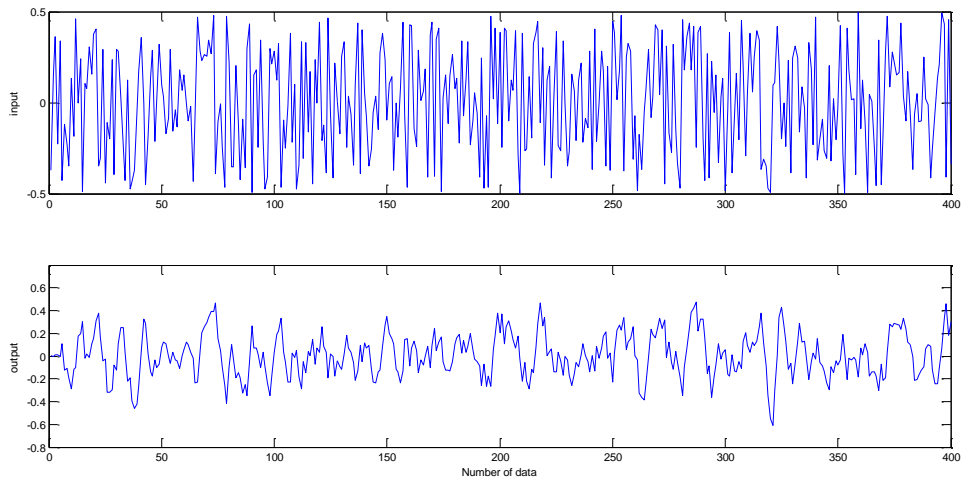


Figure 3 Input and output data for Model 2

In this simulation, by varying the mutation and the crossover rates, the convergence of the algorithm with respect to two objective functions had been observed. The average of each objective function was considered, as the algorithm was executed in one hundred runs. The averages of two objective functions were calculated as follows:

$$\text{Average of } OFn1 = \frac{1}{r} \sum_{i=1}^r \left(\frac{1}{NP} \sum_{j=1}^{NP} OFn1_j \right)_i \quad (14)$$

$$\text{Average of } OFn2 = \frac{1}{r} \sum_{i=1}^r \left(\frac{1}{NP} \sum_{j=1}^{NP} OFn2_j \right)_i \quad (15)$$

where r and NP are the number of runs and size of population, respectively. This task was investigated to set the appropriate values of CR and MR that were used for identification of the dynamic systems. The results are shown in Figures 4-11 for the convergence test, while Tables 2 and 3 list the average values of both objective functions for Model 1 and Model 2, respectively.

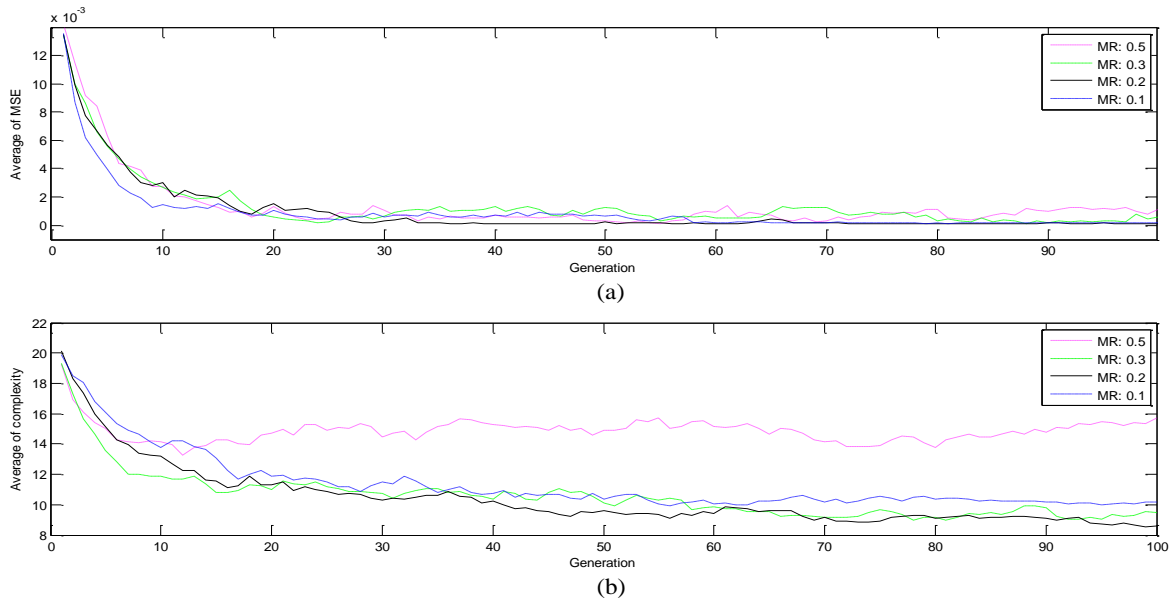


Figure 4 Convergence curve of varied MR for Model 1 with fixed CR = 0.9: (a) average of MSE (OFn1); (b) average of complexity (OFn2)

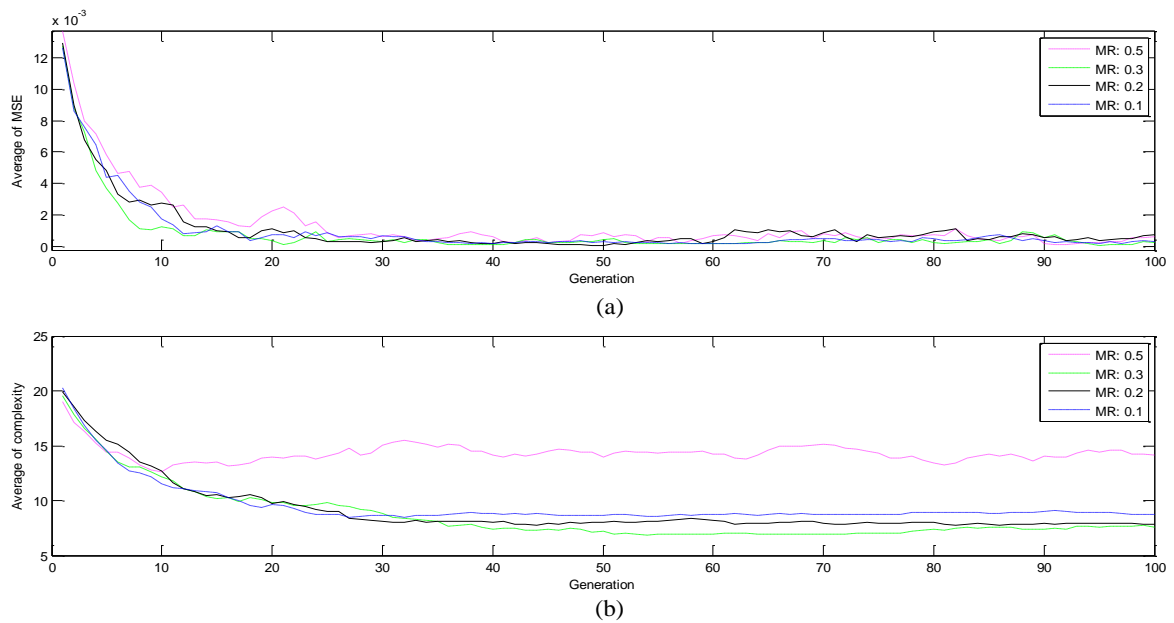


Figure 5 Convergence curve of varied MR for Model 1 with fixed CR = 0.7: (a) average of MSE (OFn1); (b) average of complexity (OFn2)

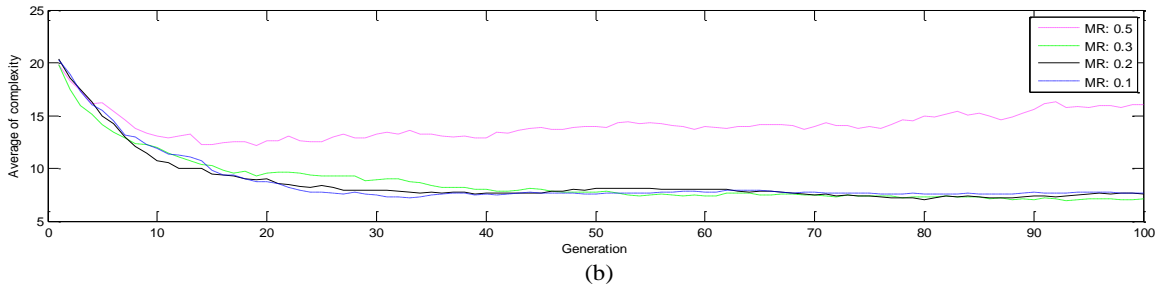
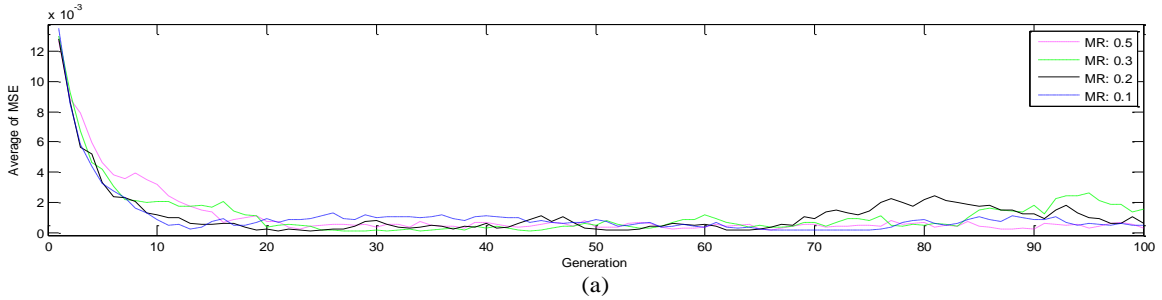


Figure 6 Convergence curve of varied MR for Model 1 with fixed CR = 0.6: (a) average of MSE (OFn1); (b) average of complexity (OFn2)

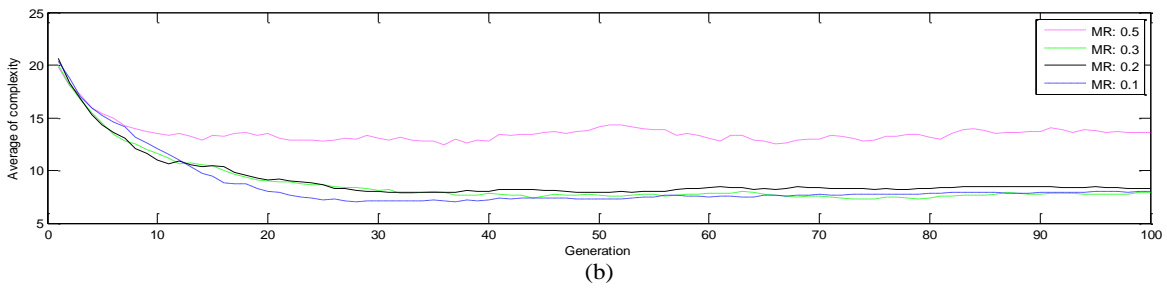
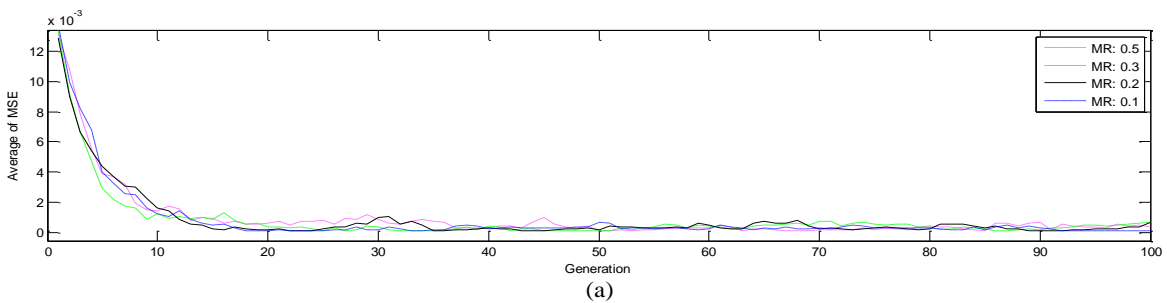


Figure 7 Convergence curve of varied MR for Model 1 with fixed CR = 0.5: (a) average of MSE (OFn1); (b) average of complexity (OFn2)

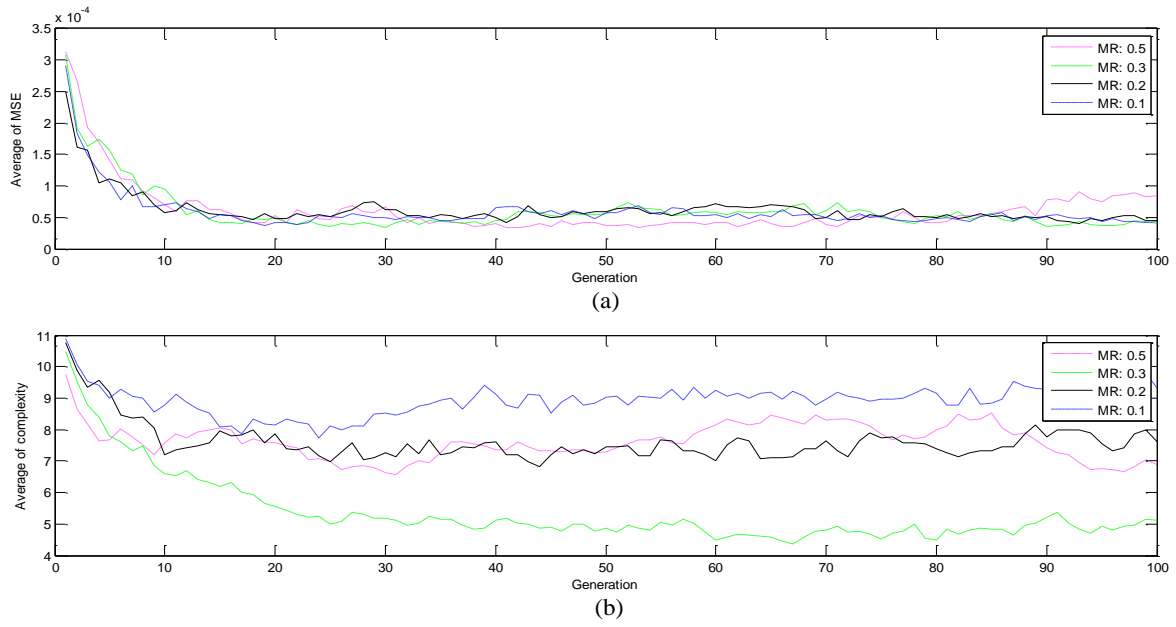


Figure 8 Convergence curve of varied MR for Model 2 with fixed CR = 0.9: (a) average of MSE (OFn1); (b) average of complexity (OFn2)

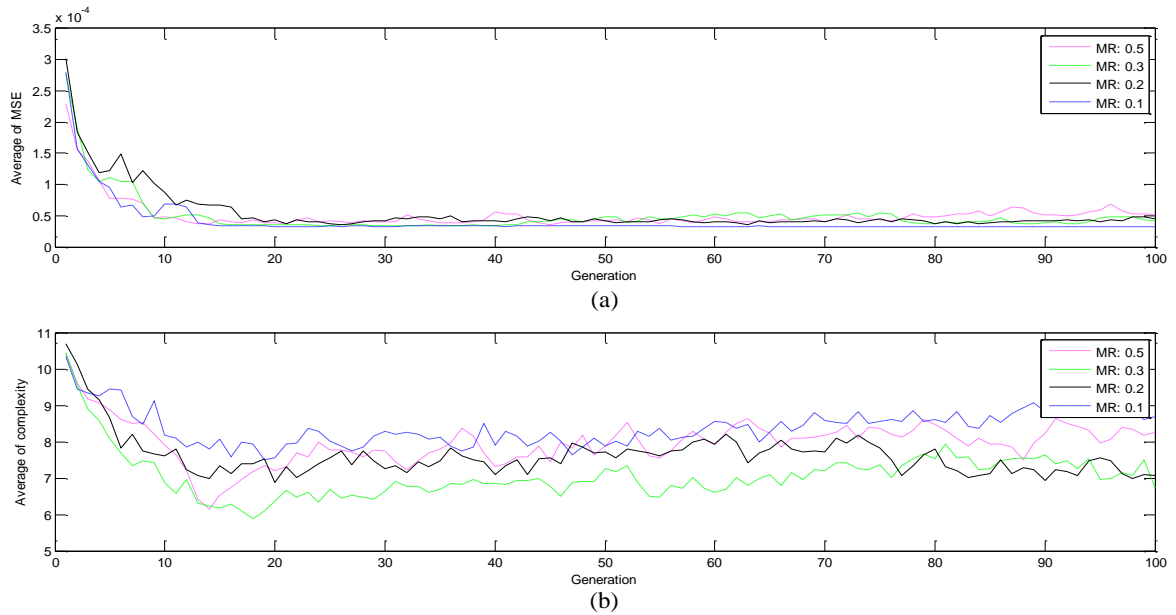


Figure 9 Convergence curve of varied MR for Model 2 with fixed CR = 0.7: (a) average of MSE (OFn1); (b) average of complexity (OFn2)

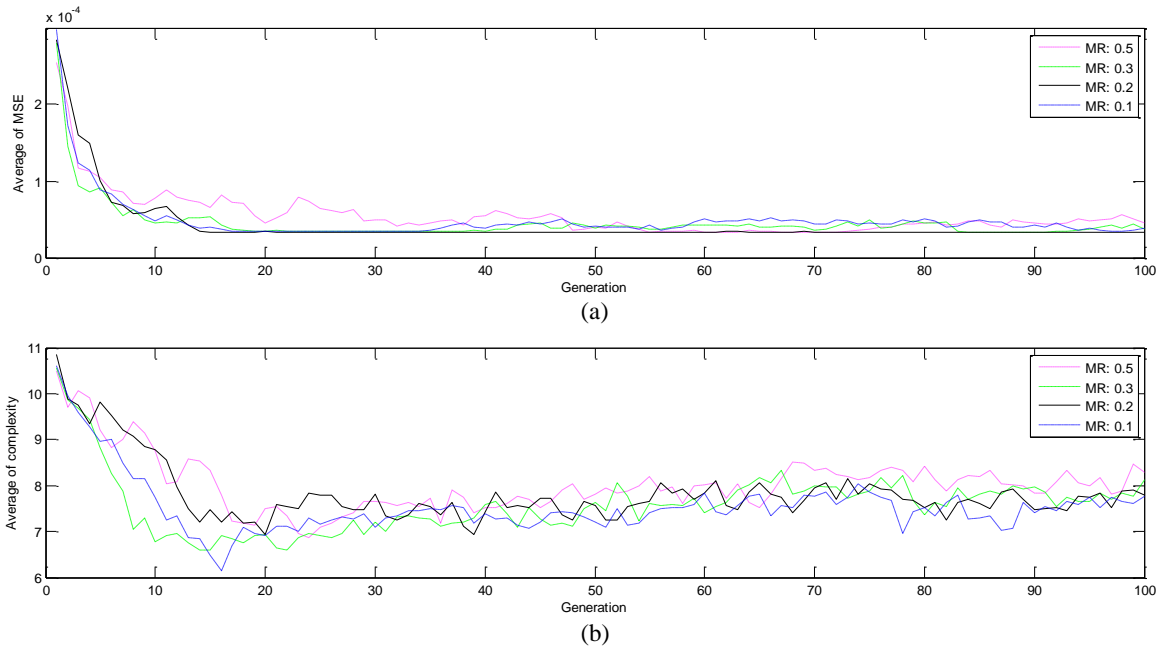


Figure 10 Convergence curve of varied MR for Model 2 with fixed CR = 0.6: (a) average of MSE (OFn1); (b) average of complexity (OFn2)

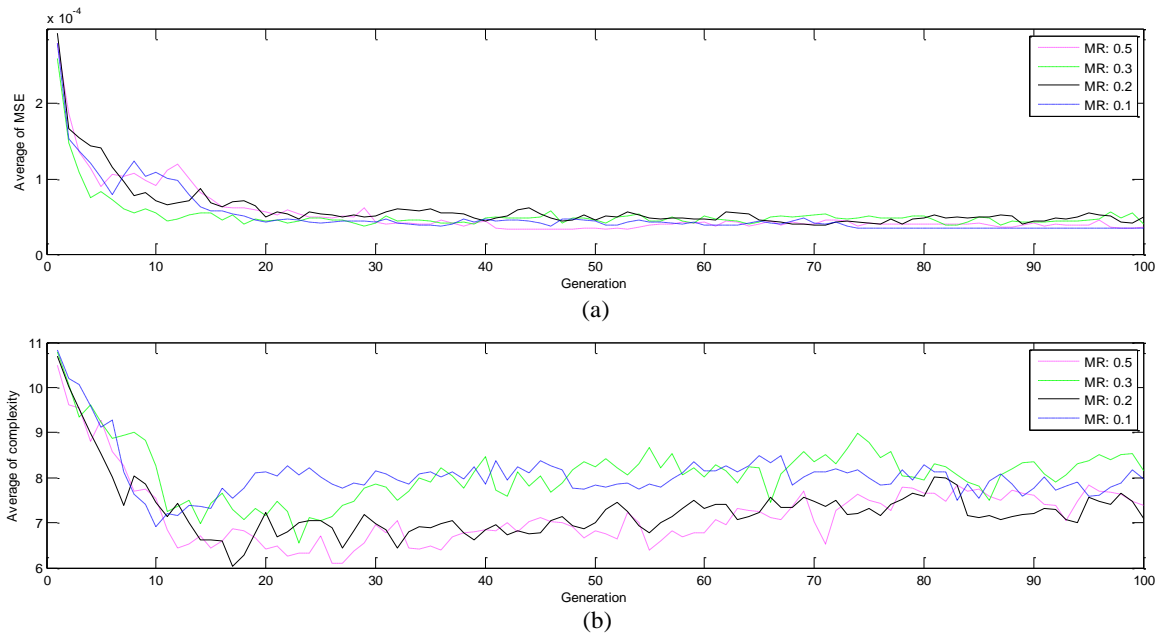


Figure 11 Convergence curve of varied MR for Model 2 with fixed CR = 0.5: (a) average of MSE (OFn1); (b) average of complexity (OFn2)

Table 2 Values of average of OFn1 and OFn2 for Model 1

MR	0.9		0.7		0.6		0.5	
	Average of OFn1	Average of OFn2	Average of OFn1	Average of OFn2	Average of OFn1	Average of OFn2	Average of OFn1	Average of OFn2
0.5	1.1×10^{-3}	15.69	6.0×10^{-4}	14.16	3.0×10^{-4}	16.07	3.0×10^{-4}	13.64
0.3	6.0×10^{-4}	9.45	2.0×10^{-4}	7.59	1.5×10^{-3}	7.08	7.0×10^{-4}	7.90
0.2	1.0×10^{-4}	8.57	7.0×10^{-4}	7.80	6.0×10^{-4}	7.59	7.0×10^{-4}	8.32
0.1	1.0×10^{-4}	10.15	3.0×10^{-4}	8.74	5.0×10^{-4}	7.69	1.0×10^{-4}	8.03

Table 3 Values of average of OFn1 and OFn2 for Model 2

CR	0.9		0.7		0.6		0.5	
	Average of OFn1	Average of OFn2	Average of OFn1	Average of OFn2	Average of OFn1	Average of OFn2	Average of OFn1	Average of OFn2
0.5	1.0×10^{-4}	6.88	5.2×10^{-5}	8.26	4.5×10^{-5}	8.28	3.5×10^{-5}	7.39
0.3	4.1×10^{-5}	5.12	4.2×10^{-5}	6.71	3.8×10^{-5}	8.12	4.0×10^{-5}	8.15
0.2	4.4×10^{-5}	7.61	4.5×10^{-5}	7.08	3.3×10^{-5}	7.79	4.9×10^{-5}	7.10
0.1	4.4×10^{-5}	9.32	3.2×10^{-5}	8.70	3.8×10^{-5}	7.76	3.4×10^{-5}	7.97

In the investigation of the convergence of the algorithm, each combination of different values of the parameters gave different curves of convergence. Considering both Model 1 and Model 2, the values of MR of 0.3 and 0.2 at each value of CR showed the advantages compared to the values of MR of 0.5 and 0.1, as shown in Figures 4-11. The advantages were the curve of convergence that was smooth and fast convergence. Moreover, they were more likely converged to the lower values of the objective functions under consideration. Thus, these can be assured by looking at Tables 2 and 3. The values of average OFn1 that were listed showed that there was no much difference for each combination of MR and CR. Meanwhile, most of the values of average OFn2 had shown lower value at 0.3 and 0.2 of MR for each value of CR, except CR = 0.6 for Model 2.

The value of crossover rate, CR, kept larger, which was more than 0.5, particularly to solve the larger search space of problem. The larger CR was required for huge-dimensional problems that was influenced for the number of individuals to be mutated in generating the offspring population [34]. From this simulation, a generalization cannot be made due to the different results that had been obtained. Therefore, these results indicated that the acceptable range for MR was [0.3, 0.2], while CR was [0.9, 0.6], respectively. However, the user had to decide the parameter values that had been preferred to be used, which were fixed in the initialization stage, while the algorithm was executed. In this study, based on the acceptable range that was stated earlier, the parameter values that were decided to use had been MR = 0.3 and CR = 0.7, respectively. These values had been confirmed to be used in executing MOODE in order to find an optimal model structure for dynamic system modelling.

5.0 CONCLUSION

A new application of MOODE was described for model structure selection in system identification problem. The study had focused on selecting the compact model structure that best represented the measured input-output data. It had shown that the proposed algorithm was capable in fulfilling the objectives of the study: to produce a good and an adequate model with minimal number of term and good predictive accuracy. The main study in this

paper had been on perturbation parameter tuning of the MOODE and its application to dynamic system modelling, which has remained as an open issue. The authors had decided to use the values of 0.7 and 0.3 in CR and MR, respectively. However, the proper selection for DE operators depends on the problem faced by a user. MOODE was performed better in minimizing both objective functions, i.e. MSE and complexity of the model structure with rightly chosen perturbation parameters.

Acknowledgement

The authors would like to acknowledge School of Manufacturing Engineering, Universiti Malaysia Perlis (UniMAP), FRGS (9003-00457), and Universiti Teknologi Malaysia (UTM) in supporting the research work.

References

- [1] L. Ljung. 1999. System Identification Theory for the User, Prentice-Hall Inc, New Jersey.
- [2] X. Hong, R. J. Mitchell, S. Chen, C. J. Harris, K. Li, G. W. Irwin. 2008. Model Selection Approaches for Non-Linear System Identification: A Review. *International Journal of Systems Science*. 39: 925-946.
- [3] L.A. Aguirre, On the Structure of Nonlinear Polynomial Models: Higher Order Correlation Functions, Spectra, and Term Clusters, *IEEE Transactions on Circuits and Systems--I: Fundamental Theory and Applications*. 44 (1997): 450-453.
- [4] I. Lind, L. Ljung. 2008. Regressor and Structure Selection in NARX Models using a Structured ANOVA Approach. *Automatica*. 44: 383-395.
- [5] R. Ahmad, H. Jamaluddin, M. A. Hussain. 2004. Model Structure Selection for a Discrete-time Non-linear System using Genetic Algorithm. *Journal of Systems and Control Engineering, Part I. Proc. Instn. Mech. Engrs*. 218: 85-98.
- [6] S. A. Billings, H.L. Wei. 2008. An Adaptive Orthogonal Search Algorithm for Model Subset Selection and Non-linear System Identification. *International Journal of Control*. 81: 714-724.
- [7] K. Rodriguez-Vazquez, C. M. Fonseca, P. J. Fleming. 2004. Identifying the Structure of Nonlinear Dynamic Systems Using Multiobjective Genetic Programming. *IEEE Transactions on Systems, Man, and Cybernetics--Part A: Systems and Humans*. 34: 531-545.
- [8] S. Chen, S. A. Billings, W. Luo. 1989. Orthogonal Least Squares Methods and Their Application to Non-linear System Identification. *Int. J. Control*. 50: 1873-1896.
- [9] M. Z. Zakaria, M. S. Saad, H. Jamaluddin, R. Ahmad. 2014. Dynamic System Modeling of Flexible Beam System using Multi-Objective Optimization Differential Evolution Algorithm. *World Virtual Conference on Advanced Research in Mechanical and Materials Engineering*. Trans Tech Publications Ltd.

- [10] K. Z. Mao, S. A. Billings. 1997. Algorithms for Minimal Model Structure Detection in Nonlinear Dynamic System Identification. *International Journal of Control*. 68: 311-330.
- [11] R. M. Storn, K. V. Price. 1997. Differential Evolution - A Simple and Efficient Heuristics for Global Optimization over Continuous Spaces. *Journal of Global Optimization*. 11: 341-359.
- [12] V. Feoktistov, *Differential Evolution: In Search of Solutions*. Springer Science+Business Media, New York, 2006.
- [13] J.A. Adeyemo, F.A.O. Otieno. 2009. Multi-Objective Differential Evolution Algorithm for Solving Engineering Problems. *Journal of Applied Sciences*. 9: 3652-3661.
- [14] B. V. Babu, P. G. Chakole, J. H. Syed Mubeen. 2005. Multiobjective Differential Evolution (MODE) for Optimization of Adiabatic Styrene Reactor. *Chemical Engineering Science*. 60: 4822-4837.
- [15] H.-Y. Fan, J. Lampinen, Y. Levy. 2006. An Easy-to-implement Differential Evolution Approach for Multi-Objective Optimization. *Engineering Computations: International Journal for Computer-Aided Engineering and Software*. 23: 124-138.
- [16] C. Peng, H. Sun, J. Guo, (2010) Multi-Objective Optimal PMU Placement using a Non-dominated Sorting Differential Evolution Algorithm, *International Journal of Electrical Power & Energy Systems*. 32: 886-892.
- [17] H. A. Abbas, R. Sarker. 2002. The Pareto Differential Evolution Algorithm. *International Journal Artificial Intelligence Tools*. 11: 531-552.
- [18] E. Zitzler, L. Thiele. 1999. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, *IEEE Transactions on Evolutionary Computation*. 3: 257-271.
- [19] E. Zitzler, Deb, K., and Thiele, L. 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*. 8: 173-195.
- [20] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 6: 182-197.
- [21] T. Krink, S. Paterlini. 2011. Multiobjective Optimization using Differential Evolution for Real-world Portfolio Optimization. *Computational Management Science*. 8: 157-179.
- [22] L. Ljung, T. Glad. 1994. Modeling of Dynamic Systems, Prentice-Hall, Inc, New Jersey.
- [23] S. Chen, S. A. Billings. 1989. Representations of Nonlinear Systems: The NARX Model. *International Journal of Control*. 49: 1013-1032.
- [24] R. M. Storn. 1996. On the Usage of Differential Evolution for Function Optimization, *Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS (1996) IEEE, Berkeley, CA, New York*. 519-523.
- [25] M. Z. Zakaria, H. Jamaluddin, R. Ahmad, S.M.R. Loghmanian. 2012. Comparison between Multi-Objective and Single-Objective Optimization for the Modeling of Dynamic Systems, *Journal of Systems and Control Engineering, Part I. Proc. Instn. Mech. Engrs*. 226: 994-1005.
- [26] R. Johansson. 1993. *System Modeling and Identification*, Prentice-Hall International Inc, New Jersey
- [27] K. Deb. 2001. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Ltd, Chicester, England.
- [28] S. M. R. Loghmanian, H. Jamaluddin, R. Ahmad, R. Yusof, M. Khalid. 2012. Structure Optimization of Neural Network for Dynamic System Modeling using Multi-Objective Genetic Algorithm, *Neural Computing and Applications*. 21: 1281-1295.
- [29] A. E. Eiben, R. Hinterding, Z. Michalewicz. 1999. Parameter control in evolutionary algorithms, *Evolutionary Computation. IEEE Transactions on*. 3: 124-141.
- [30] A. E. Eiben, S. K. Smit. 2011. Parameter Tuning for Configuring and Analyzing Evolutionary Algorithms, *Swarm and Evolutionary Computation*. 1: 19-31.
- [31] T. Back, D. B. Fogel, Z. Michalewicz. 2000. *Evolutionary Computation 1: Basic Algorithms and Operators*, Institute of Physics Publishing, Bristol and Philadelphia.
- [32] M. Z. Zakaria, H. Jamaluddin, R. Ahmad, A. H. Muhaimin. 2011. Effects of Genetic Algorithm Parameters on Multiobjective Optimization Algorithm Applied to System Identification Problem. *Modeling, Simulation and Applied Optimization (ICMSAO), 2011 4th International Conference on, Kuala Lumpur*. 1-5.
- [33] G.-C. Luh, C.-Y. Wu. 1999. Nonlinear System Identification using Genetic Algorithm. *Journal of Systems and Control Engineering, Part I. Proc. Instn. Mech. Engrs*. 213: 105-118.
- [34] D. Zaharie. 2009. Influence of Crossover on the Behavior of Differential Evolution Algorithms. *Applied Soft Computing*. 9: 1126-1138.