Deterministic Mutation-Based Algorithm for Model Structure Selection in Discrete-Time System Identification

Md Fahmi ABD SAMAD, Hishamuddin JAMALUDDIN, Robiah AHMAD, Mohd. Shafiek YAACOB and Abul K. M. AZAD

Abstract- **System identification is a method of determining a mathematical relation between variables and terms of a process based on observed input-output data. Model structure selection is one of the important steps in a system identification process. Evolutionary computation (EC) is known to be an effective search and optimization method and in this paper EC is proposed as a model structure selection algorithm. Since EC, like genetic algorithm, relies on randomness and probabilities, it is cumbersome when constraints are present in the search. In this regard, EC requires the incorporation of additional evaluation functions, hence, additional computation time. A deterministic mutation-based algorithm is introduced to overcome this problem. Identification studies using NARX (Nonlinear AutoRegressive with eXogenous input) models employing simulated systems and real plant data are used to demonstrate that the algorithm is able to detect significant variables and terms faster and to select a simpler model structure than other well-known EC methods.**

***Index Terms*—Evolutionary computation; Model structure selection; Nonlinear AutoRegressive with eXogenous input model; System identification, Correlation test**

1. INTRODUCTION

System identification is a method of recognizing the characteristics of a system, thus producing a quantitative input-output relationship that explains or resembles the system’s dynamics [16]. The procedure of identification can be divided into four distinctive steps. The first step is the data acquisition. The second step is the model representation and structure selection that requires selection of the type of model followed by construction of the correct or optimal model structure. For a discrete-time difference equation, model structure selection includes the selection of degree of nonlinearity, maximum orders of lag for output, input, noise and time delay. Even when a model is found to be sufficiently able to represent a system at hand, it is always convenient for users to have a parsimonious model structure for the system. This can be achieved by penalizing less feasible solutions by comparison of the number of variables and terms through penalty functions [12]. The third step in system identification is the parameter estimation where the values of parameters based on the selected model structure are estimated. The fourth step is the model validity tests intended to verify or validate the representation of the final model selected.

The model structure selection stage requires a robust method that is able to search among model structure alternatives the global optimal model structure. Such a characteristic is found in EC [9]. EC, termed since 1991, groups four types of optimization methods i.e. genetic algorithm (GA), evolutionary programming (EP), evolution strategies (ES) and genetic programming. Their differences, as traced back from their original versions, are in representation, selection and genetic operators. The application of EC extends in the planning, design, identification, control and classification in engineering analysis and development [3, 22] and some examples of its usage in system identification, particularly in model structure selection stage, are also available [1, 2, 17, 20].

An analytical method in model structure selection is found in the extension of least square method, known as orthogonal least square (OLS), which evaluates the significance of a variable based on error reduction ratio (ERR) [21]. The OLS method requires the selection of threshold values of ERR to distinguish significance. The main disadvantage is that the change in the values of ERR for a given model is usually gradual, causing difficulty in justifying the level of significance and thus deciding the number of variables to be included in the final model. The use of a genetic algorithm has been shown to be better than the OLS method [1, 2].

However, the performance of EC is highly dependent on its algorithm parameters. Researches are wide in search of the optimal setting of the parameters since it affects the performance in terms of computation speed, exploration and exploitation of search solutions, especially when constraints are present [1, 8, 18]. Poor selection of these parameters may cause GA to either converge prematurely or too slowly. As an addition, the objective function (*OF*) in GA needs to be carefully redefined when parsimony of the model becomes a requirement in model structure selection.

The main aim of this paper is to propose a model structure selection algorithm, named deterministic mutation algorithm (DMA). It differs from other EC methods where its genetic operation is deterministic. In [15], deterministic mutation is based on a quasi-Newton optimization method for the problem investigated whereby, in this paper, the operator is ruled by its past performances to enable quicker detection of parsimonious models. It is an original adaptation of EC with no crossover and the characteristic of a forward search. Forward search evaluates variables one at a time and has been used such as

Received on Aug. 16, 2010; Revised on June 10, 2011. M. F. Abd Samad is with Faculty of Mechanical Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Malacca, Malaysia (e-mail: mdfahmi@utem.edu.my); H. Jamaluddin, R. Ahmad and M. S. Yaacob are with Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johore, Malaysia (e-mail: hishamj@fkm.utm.my, robiah@fkm.utm.my and shafiek@fkm.utm.my); A. K. M. AZAD is with College of Engineering and Engineering Technology, Northern Illinois University, Illinois 60115, USA (e-mail: azad@ceet.niu.edu).

in direct search, stepwise forward inclusion method and with orthogonal least square-error reduction ratio estimation method [6, 21]. In these methods, parsimony has a fixed level that, like predictive accuracy, is based on a prespecified significance level, and only one final model is produced for the validation stage. There is also a risk of validity, as the methods leaves out the parameter estimation stage before the selection is made. In this paper, a population-based optimization method is designed to overcome the above weaknesses, and also to require less time and an easier setting when compared to conventional algorithms.

This paper is organized in accordance with the steps of system identification as follows. Section 2 explains the model structure selection problem, with emphasis on the NARX model, and parameter estimation methods. Section 3 explains the application of EC, where emphasis is given to GA followed by an explanation of the proposed algorithm, DMA. Section 4 explains the model validation methods while section 5 is the simulation study and discussion of several algorithms. Section 6 concludes the paper.

2. Model Structure Selection and Parameter Estimation

2.1 Model Structure Selection

Various models are available when determining a suitable model representation for a system. These models are differentiated by either linear or nonlinear models where the latter provide much richer possibilities in describing systems and have better flexibility when inferring from a finite data set [16].

The NARX model is a common model structure representation for a nonlinear discrete-time system, which is also a generalization of the linear difference equation. It is written as:

$$y(t) = F_*^l[y(t-1), y(t-2), ...y(t-n_y), u(t-d), \\ u(t-d-1), ..., u(t-d-n_u+1)] + e(t)$$

where $F_*^l[\cdot]$ is a nonlinear function; $y(t)$, $u(t)$ and $e(t)$ are output, input and noise, respectively at time t; n_y and n_u are the maximum orders of lag for output and input, respectively; d is the time delay and l is the degree of non-linearity.

The NARX model can be converted into a linear-in-the-parameter model so that simple parameter estimation methods can be applied. An example of such a model is the linear regression model and is written as:

$$y(t) = \boldsymbol{\vartheta}^T(t)\ \boldsymbol{\Phi} + e(t),\ n_y \le t \le N$$

where $\boldsymbol{\theta}$ is the parameter vector, $\boldsymbol{\varphi}=[\varphi_1\ \ \varphi_2 \ldots \varphi_L]$ is the regressor vector, L is the maximum number of regressors adequate in describing the system's dynamics and N is the total number of data depending on the sampling frequency.

The regressors represent the variables and terms of the system and hence determine the size of the parameter vector. The number of possible regressors (L) in a NARX model increases with the degree of nonlinearity and orders of lag and is calculated as follows:

$L = M + 1$

where

$M = \sum_{i=1}^{l} n_i$, where l = degree of nonlinearity

and

$n_i = \dfrac{n_{i-1}(n_y + n_u + i - 1)}{i}$, where $n_o = 1$

Suppose that a system is known to have nonlinearity, l = 2; time delay, d = 1; maximum order of lag for input, n_u = 2 and maximum order of lag for output, n_y = 2. This makes L = 15, along with the inclusion of a constant term. In a linear-in-the-parameter form, the model is written as follows:

$$\begin{aligned} y(t) = {} & a_1 + a_2y(t-1) + a_3y(t-2) + a_4u(t-1) \\ & + a_5u(t-2) + a_6y^2(t-1) \\ & + a_7y(t-1)y(t-2) + a_8y(t-1)u(t-1) \\ & + a_9y(t-1)u(t-2) + a_{10}y^2(t-2) \\ & + a_{11}y(t-2)u(t-1) + a_{12}y(t-2)u(t-2) \\ & + a_{13}u^2(t-1) + a_{14}u(t-1)u(t-2) \\ & + a_{15}u^2(t-2) \end{aligned} \tag{1}$$

where a_i (i = 1, 2, 3, …, 15) are the parameters of the model with a_1 as a constant.

Since the decisions on the regressors are either inclusion or omission, simple binomial theorems apply. Therefore, the number of model choices from a fully expanded difference equation model is 2^L-1, where L is the number of possible regressors. Given L = 15, the search space consists of 32 767 points. The structure selection stage deals with the selection of an appropriate model for the system based on accuracy, parsimony, etc. An evaluation of each point or candidate model is exhaustive or computationally laborious.

2.2 Parameter Estimation

The least square method is an effective method of determining the parameters when a linear-in-the-parameter model is used [16]. The method was founded by a German mathematician Carl Friedrich Gauss in 1821 and has since been developed for various optimization problems [14]. Another well-known method is the maximum likelihood estimation introduced by R. A. Fisher [14] but when the model used in the evaluation is of linear-in-the-parameter type and the noise data is uncorrelated, the least square method is relatively simpler and shall be used [16].

3. Evolutionary Computation and Deterministic Mutation Algorithm in Model Structure Selection

The development of EC began in mid-1950s [9]. As mainstream EC representatives, evolutionary algorithms (GA, EP and ES), share the same metaphor as their working principle. That principle is the theory of biological evolution introduced by the famous Charles Darwin in 1859. The effectiveness of EC is related to its parameter control.

3.1 Genetic Algorithm

Among all methods in evolutionary algorithms, GA, introduced by Holland [19], is the most widely known and has wide application [4]. In the model structure selection, it begins its search for the optimal model structure by initializing a set of chromosomes in a population to represent candidate models. Chromosome consists of genes separated by different positions, also defined as locus in which each gene carries an allele i.e. information of the search point.

The typical coding method in GA is the binary representation. As each gene represents a variable or a term of a model in model structure selection, the allele is either 1 for presence or 0 for absence. As an illustration, based on the model given in Eq. (1) and certain regressor coding, the chromosome [110 100 001 000 100] may represent the following model:

$$\begin{aligned} y(t) &= a_1 + a_2 y(t-1) + a_4 u(t-1) \\ &\quad + a_9 y(t-1)u(t-2) + a_{13} u^2(t-1) \end{aligned}$$

The chromosomes are then evaluated using a specified *OF*, where this *OF* is converted into a fitness function. The main aim in this evaluation is to identify chromosomes of high fitness before the selection stage takes place. In the case of the least square method, *OF* is the minimization of the sum of squares of the differences between the model output and real data and is written as

$$OF = \sum_{i=1}^{N} \left(y_i(t) - \hat{y}_i(t) \right)^2 \qquad (2)$$

where $\hat{y}(t)$ is the predicted output and $y(t)$ is the actual output value, both at time *t*.

However, since GA does not have its own mechanism to differentiate parsimonious models from non-parsimonious ones, a penalty function can be integrated into its *OF*, written as follows:

$$OF = \sum_{i=1}^{N} \left(y_i(t) - \hat{y}_i(t) \right)^2 + \log(n) \qquad (3)$$

where n is the number of insignificant regressors plus 1 [7]. Insignificant regressors can be identified by regressors with their estimated parameter less than a specified penalty coefficient (*penalty*). The selection of *penalty* is made based on the knowledge that the number of regressors selected for a regression model is inversely related to the penalty coefficient (i.e. higher penalty coefficient selects a model with less regressors).

Here, low *OF* value is identified as high fitness and the opposite for high *OF* value. Conventional GA uses a proportional selection method for reproduction. The reproduction process is followed by genetic operations like crossover and mutation to produce chromosomes of new features known as offsprings. The processes are repeated until a stopping criterion like the maximum number of generations (*max_gen*) [19]. The best individual of the last population is identified as the optimal solution to the problem. The cycle of processes require a preselected setting of various algorithmic parameters such as population size (*pop_size*), crossover probability (p_c) and mutation probability (p_m). Attempts are continuously being made to identify the optimal set of parameters for use with GA. However, results do not always support each other when a change to a single parameter is made, i.e., the parameters are interdependent, or different optimization problems are attempted.

3.2 Modified Genetic Algorithm

The modified genetic algorithm (MGA) is a modification of the canonical GA, where it functions by the division of population into groups and each group undergoes different genetic operations [1]. For model structure selection, its representation and procedures are the same as GA. Its rationale of modification is in the allocation of different genetic operations to groups of individuals having different fitness in the population. By dividing the population into four groups of decreasing fitness values, these groups, respectively, undergo

1. Conservation following the elitist strategy
2. Mutation
3. Sequence of crossover and mutation
4. Replacement if a new chromosome is fitter

MGA has the potential for quicker convergence than simple GA but its optimality still depends on the parameters as described earlier [13].

3.3 Modelling Using Deterministic Mutation Algorithm

The fundamentals of GA lie in the theory of implicit parallelism [19]. It states that GA is able to globally search the optimal solution by concurrently evaluating multiple points when it evaluates a chromosome schema. The performance of GA as a robust search method is also related to the building block hypothesis in which the emphasis during the search is mentioned as toward the juxtaposition of short, low-order, high performance schemata [19]. Schemata can also be seen as hyperplanes. By knowing the optimal hyperplane, the search can be made more narrowed or easily converged.

DMA is developed based on the above notions. Since the introduction of GA, crossover has been a prominent genetic operator with the mutation as a ‘background operator’ used only to ensure that no important allele or value of gene in a chromosome during the search is left out. However, DMA uses mutation as its main operator, as has typically been the case in evolution strategies.

By beginning the search with a population of the most parsimonious model candidates, optimality of the final solution in DMA is cautiously tracked down by its accuracy and parsimony through generational mutation. These parsimonious models are found from chromosomes of few 1-valued genes. Hence, the initial population is made up of all chromosomes of the same bit string length with only one gene with allele 1. The chromosomes can be considered as order 1 chromosomes if the 0-valued genes are regarded as wildcard genes since these genes do not contribute directly to evaluation for now. The evaluation is carried out using the same description as in subsection 3.1 but according to Eq. (2), i.e., one without the use of a penalty function. The evaluation allows high fitness schema in the population to be identified and consequently the critical gene locus for mutation operation. The mutation operator applied does not require a probabilistic approach, as usually the case in GA, instead, is compulsory in each chromosome of each generation. By defining the chromosome length as L, $p_m = 1/L$. This deterministic property of the mutation operator is also described as the identification of the significant gene for mutation from the former generation via the best chromosome. The term 'deterministic' is used to emphasize that from a bit's perspective, the probability of mutation is either 1 for critical gene or 0 for non-critical ones.

The rule is applied to all chromosomes in the following generation except to the best chromosome, which is cut off from the population. The reason that the best chromosome is cut off is that, by that stage, better accuracy can only be achieved by more complex chromosomes, provided that no ill-conditioned estimation occurs. The mutation, hence, allows the size of the population to be reduced by 1 chromosome each generation. Meanwhile, evaluation results, most importantly of the best chromosome in each generation, can be recorded in a bookkeeping manner. The algorithm is stopped when *pop_size* reaches 1, where all levels of complexity (or parsimony) have been evaluated. The algorithm allows a collection of solutions and accuracy values in increasing model structure complexity with respect to generation.

The steps of DMA are given as follows:

Step 1: Initialize population with an identity matrix of size $L \times L$ where L is the number of possible regressors and length of a bit string such that each row represents 1 chromosome. Let generation number, $t = 1$.

Step 2: Evaluate each chromosome in population based on the objective function.

Step 3: Identify and let the best chromosome be C_t. Identify and let the locus of the critical gene in C_t of the current generation be L_t. The locus is identified by the position of bit 1 in C_t where the same position is occupied by bit 0 in other chromosomes.

Step 4: Record and remove the best chromosome C_t from the population while flipping the gene of other chromosomes at L_t from 0 to 1. This process is called deterministic mutation. Let $t = t + 1$.

Step 5: Repeat Steps 2 - 4 until $t = L$ where all genes have allele 1.

To illustrate its operation, assume a problem where $L = 4$. The initial population at generation $t = 1$ looks as follows:

Generation number Population Objective function

$$t = 1 \qquad \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \qquad \begin{matrix}OF_{11}\\ OF_{12}\\ OF_{13}\\ OF_{14}\end{matrix}$$

Next to each row is OF for each chromosome where each row in the population represents a chromosome. Here, OF_{pq} is OF of the q-th chromosome at p-th generation. Assume a minimization problem and $OF_{12} < OF_{1i}$ (i = 1, 3 and 4). To identify the locus of the critical gene, the position of bit 1 in C_1 = [0 1 0 0] is located. Counting from left to right, $L_1 = 2$. Since C_1 is now a potential solution, its structure is recorded. It is removed from the population to avoid evaluation redundancy, while deterministic mutation is applied to other chromosomes yielding:

$$t = 2 \qquad \begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 1 & 1 & 0\\ 0 & 1 & 0 & 1\end{bmatrix} \qquad \begin{matrix}OF_{21}\\ OF_{22}\\ OF_{23}\end{matrix}$$

Assume $OF_{23} < OF_{2j}$ (j = 1 and 2). This gives C_2 = [0 1 0 1] and since it is the only chromosome with bit 1 at position 4, $L_2 = 4$. The population of the next generation then looks as follows:

$$t = 3 \qquad \begin{bmatrix}1 & 1 & 0 & 1\\ 0 & 1 & 1 & 1\end{bmatrix} \qquad \begin{matrix}OF_{31}\\ OF_{32}\end{matrix}$$

The process continues until there is only one chromosome ($t = L$), where in this case, C_4 = [1 1 1 1]. The issue of when to stop the evolution can also be addressed in other ways, like dropping percentage of OF value or number of regressors. Users may decide to stop the evolution when the changes of OF value are insignificant to a certain percentage or when the number of regressors reaches certain quantity. Users may also select their model from selected potentials based on validity results. The algorithm provides a collection of the best chromosome in every generation, C_t (t = 1, 2, …, L) and their OF values. Notice that each chromosome in the collection has a different parsimony. For a minimization problem, it is explained in section 5 that $OF_p < OF_t$ for every $t < p$.

4. Model Validation

The model validity tests are used to verify whether the model selected fits the system data adequately. Before the tests are carried out, performance indicators of solutions are also calculated. The selected solution is validated using methods like correlation tests [5]. Model validation can also be made by cross-validation utilizing either the *k*-step-ahead prediction or the model predicted output function. In cross-validation, the error is calculated from the difference between the true output and predicted output for each data point as follows:

$$\text{error}(t) = \left| y(t) - \hat{y}(t) \right|$$

where $\hat{y}(t)$ is obtained by using the least square estimation method. In order to allow comparison to other optimization functions, the error is also normalized to the output's range of values for the same type of input. The value is given in a percentage form calculated as

$$\text{percentage of error}(t) = \frac{\text{error}(t)}{y_{\max} - y_{\min}} \times 100\%$$

where $y_{\max}$ and $y_{\min}$ are the maximum and minimum values of true output throughout the set of data, respectively.

4.1 Performance Indicators

In simulation study, the best model identified from a simple genetic algorithm or SGA (as described in subsection 3.1), MGA (subsection 3.2) and DMA are compared by using the following performance indicators:

1. Error index

In SGA and MGA, the best chromosome is identified in the last generation based on *OF* value. The error index (*EI*) of the chromosome is calculated as:

$$EI = \sqrt{\frac{\sum \left(y(t) - \hat{y}(t) \right)^2}{\sum y^2(t)}}$$

where $y(t)$ is system output and $\hat{y}(t)$ is predicted output.

2. Number of regressors

The number of regressors of the best chromosome is simply counted to demonstrate the level of parsimony, just as *EI* represents model accuracy.

3. Computation time

The computation time is used to demonstrate the efficiency of each algorithm. For SGA and MGA, a specific *pop_size* and *max_gen* is given. Since DMA allows the specification of its algorithm parameters to be based on the length of a chromosome, which is identified from the number of possible regressors, its computation time can be used to specify the parameters for SGA and MGA.

4.2 Correlation Tests

The correlation tests involve the determination of correlation between the system's properties. Linear models only require the detections of autocorrelation of residuals ($\phi_{\varepsilon\varepsilon}$) and correlation of residuals and system's inputs ($\phi_{u\varepsilon}$). Since nonlinear models contain polynomials of variables, the models require a more expanded number of tests [5]

$$\phi_{\varepsilon\varepsilon}(\tau) = E[\varepsilon(t-\tau)\varepsilon(t)] = \delta(\tau),\ \tau = 0 \qquad (4a)$$

$$\phi_{u\varepsilon}(\tau) = E[u(t-\tau)\varepsilon(t)] = 0,\ \forall\tau \qquad (4b)$$

$$\phi_{u^{2'}\varepsilon}(\tau) = E\left[\left(u^2(t-\tau) - \overline{u^2}\right)\varepsilon(t)\right] = 0,\ \forall\tau \qquad (4c)$$

$$\phi_{u^{2'}\varepsilon^2}(\tau) = E\left[\left(u^2(t-\tau) - \overline{u^2}\right)\varepsilon^2(t)\right] = 0,\ \forall\tau \qquad (4d)$$

$$\phi_{\varepsilon(\varepsilon u)}(\tau) = E[\varepsilon(t)\varepsilon(t-1-\tau)u(t-1-\tau)] = 0 \quad \tau \geq 0 \qquad (4e)$$

where $E[\bullet]$ is the expectation operator. The residual, $\varepsilon(t)$ is calculated by

$$\varepsilon(t) = y(t) - \hat{y}(t)$$

where $\hat{y}(t)$ is the predicted output.

The commonly accepted bandwidth for the model's fit to the system is approximately $\pm 1.96 / \sqrt{N}$, based on 95% confidence interval, where N is the number of data points [5].

5. Simulation Study

The simulation study is conducted to compare the performance of SGA, MGA and DMA in obtaining parsimonious models for system identification problems. The data acquisition is made by simulating several NARX models of different nonlinearities and using a real plant data. The simulated models are used to enable direct comparison of the correct solution to the solution found by the algorithms. This is possible by knowing the sequence of regressor selection in the computer program. The application of the algorithms is then extended to a real plant problem where the correct structure is unknown.

All simulated models are causal systems of different specifications where the outputs are defined by past inputs and outputs. The inputs are white signals, $u(t)$, generated from a uniform distribution function in the interval [-1, 1] and white noise $e(t)$ generated from a uniform distribution function in the interval [-0.01, 0.01]. Five hundred data points are generated from each simulation. The real plant data is gas furnace data suitable for *a posteriori* modelling [7].

The settings of SGA and MGA are made by selecting 0.6 for crossover probability and 0.01 for mutation probability. The value for crossover probability is taken from De Jong's genetic algorithm, claimed as optimum for both online and offline applications [10] and also recognized as the benchmark for parameter control study using meta-level GA [11]. The value for mutation probability is also claimed to be suitable for both online and offline applications in the meta-level GA study. Depending on the computation time needed by DMA, *pop_size* and *max_gen* for SGA and MGA are set equally at a specific value. This equality is used to ensure adequacy of both variations and manipulations of chromosomes. The penalty coefficient is selected slightly lower than or equal to the absolute value of the smallest parameter either in the simulation model or the model identified from DMA to ease the search for optimal penalty coefficient yet expected to satisfy the same level of parsimony as the model in DMA. The evaluation of chromosomes in SGA and MGA is made according to the *OF* described in Eq. (3). The performance of all the algorithms is compared via the performance indicators; Error index (*EI*), number of regressors and computation time where the computation time is taken by using a 1.70 GHz processor.

5.1 Simulated Models

Simulated Model 1 has an input lag, $n_u = 2$; time delay, $d = 1$; output lag, $n_y = 1$; nonlinearity, $l = 3$ and therefore has a search space of $2^{20} - 1 = 1\,048\,575$ points. The model is defined as

$$\begin{aligned} y(t) &= 0.5y(t-1) + 0.3u(t-2) + 0.3y(t-1)u(t-1) \\ &\quad + 0.5u^3(t-1) + e(t) \end{aligned}$$

The correct chromosome for this simulated model is [010 101 000 000 000 010 00]. To achieve approximately the same or more computation time than used by DMA, *pop_size* and *max_gen* for SGA and MGA are set equally at 20. Penalty is given to regressors with their parameter < 0.1.

Simulated Model 2 has an input lag, $n_u = 3$; time delay, $d = 1$; output lag, $n_y = 2$; nonlinearity, $l = 2$ and a search space of $2^{21} - 1 = 2\,097\,151$ points. The model is defined as

$$\begin{aligned} y(t) &= 0.5y(t-1) + 0.4u(t-1) - 0.2u(t-3) \\ &\quad + 0.3y(t-1)u(t-1) + 0.1y^2(t-2) \\ &\quad + 0.4u^2(t-3) + e(t) \end{aligned}$$

In this simulation, the correct chromosome is [010 101 001 001 000 000 001]. The values of *pop_size* and *max_gen* in SGA and MGA are set to 20 whilst *penalty* is set to 0.1.

Simulated Model 3 has a fairly larger number of correct and possible regressors. Its specifications are input lag, $n_u = 2$; time delay, $d = 1$, output lag, $n_y = 2$, nonlinearity, $l = 3$, therefore, a search space of $2^{35} - 1 = 34\,359\,738\,367$ points. The model is defined as

$$\begin{aligned} y(t) &= 0.5 + 0.4u(t-1) + 0.2y(t-1)u(t-1) \\ &\quad - 0.2y(t-1)u(t-2) + 0.3y(t-2)u(t-2) \\ &\quad + 0.2y^2(t-1)y(t-2) + 0.7y^2(t-1)u(t-2) \\ &\quad - 0.2y(t-1)u^2(t-1) - 0.8y^3(t-2) \\ &\quad - 0.4u^3(t-1) + e(t) \end{aligned}$$

where the number of correct regressors is 10 and the correct chromosome is [100 100 011 001 000 010 100 010 010 000 001 000]. The *pop_size* and *max_gen* for SGA and MGA are both set to 30 whilst *penalty* is 0.15.

Shown in Figs. 1-3 are the plots of *OF* and *EI* values against the number of generations for simulated Models 1-3, respectively, using SGA, MGA and DMA. Only the selected evolution period is shown because, in the beginning of evolution, the values of *OF* and *EI* using DMA are very high. Since MGA maintains its elitism, its *OF* values keep decreasing. DMA shows better convergence as the number of generations, hence the number of regressors, increases. For simulated Model 1, based on the records of DMA, the best chromosome in a term balanced by accuracy and parsimony is at generation 4, which gives the correct structure [010 101 000 000 000 010 00]. For simulated Model 2, the selected chromosome is from generation 6 that has the correct structure [010 101 001 001 000 000 001]. For simulated Model 3, the correct chromosome is selected by DMA in generation 10. Table 1 shows the performance measures of the three algorithms. Tests using correlation tests to the selected chromosomes from DMA for the simulated models give all results either within the confidence bands or containing only slight correlations at a few lag values.

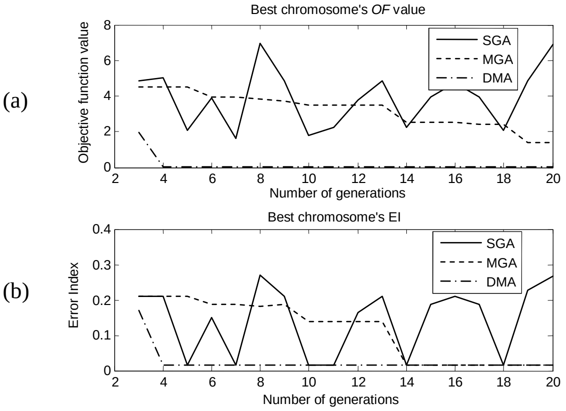


Fig. 1 (a) *OF* value versus the number of generations (b) *EI* versus the number of generations by using different algorithms for simulated Model 1

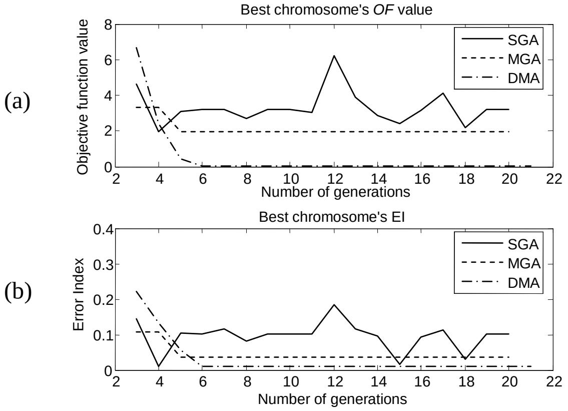


Fig. 2 (a) *OF* value versus the number of generations (b) *EI* versus the number of generations by using different algorithms for simulated Model 2

To further verify that the model obtained is input-independent, cross-validation is carried out. An additional 100 output data are generated from Model 3 and superimposed to the output values of the DMA selected model using a different input function. Instead of random white signal, the input $u(t)$ in these additional data are given by

$$u(t) = 0.5\big(\sin(0.3t) + \cos(0.4t)\big)$$

where t is sampling time = 501, 502, …600.

Throughout the superimposition, the maximum error is, however, found in the estimation set calculated as 0.0120. Its percentage of error is 0.96%. In the test set, the maximum error and percentage of error are 0.0099 and 0.89%, respectively.

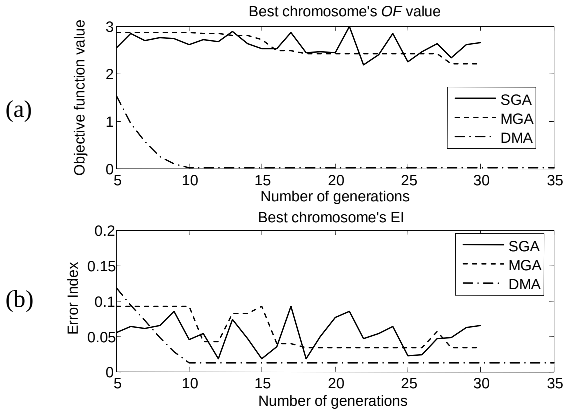


Fig. 3 (a) *OF* value versus the number of generations (b) *EI* versus the number of generations by using different algorithms for simulated Model 3

Table 1: Performance measures of SGA, MGA and DMA for simulated models

Simulated Models	**Performance Indicators**	**SGA**	**MGA**	**DMA**
1	Number of regressors	13	7	4
	Error index	0.2684	0.0162	0.0162
	Computation time (seconds)	1.625	1.421	1.156
2	Number of regressors	12	11	6
	Error index	0.1027	0.0308	0.0113
	Computation time (seconds)	1.719	1.453	1.187
3	Number of regressors	19	19	10
	Error index	0.0652	0.0336	0.0123
	Computation time (seconds)	2.984	3.234	1.890

5.2 Gas Furnace Data

This input-output data are from an actual process plant [7]. The data consist of a discrete stochastic input series of gas feed rates in cubic feet per minute and output series of carbon dioxide concentrations in outlet gas. There are 296 pairs of input-output data sampled at an interval of 9 seconds.

For real data such as this, the specifications of the model are determined first. Therefore, before SGA and MGA are tested, several variations of nonlinearity degrees, output lag order and input lag order of a NARX model are tested using DMA. The models from chromosomes with the most 1-valued genes are analyzed using the extended correlation tests in Eqs. (4a)-(4e). More regressors usually give a better model fit, but the least square method becomes ill-conditioned when the number of parameters is too large. In order to avoid this overfitting situation, only well-conditioned chromosomes of various specifications are compared. Based on the correlation tests, the most suitable variation for the gas furnace data is found to be l = 3, n_y = 7, and n_u = 2 where the total number of possible regressors is 220. This gives more than 10^{66} possible models to choose from.

DMA is then applied again to find a parsimonious model. As in the procedure in subsection 4.1, *pop_size* and *max_gen* for DMA are set to 220. The *pop_size* and *max_gen* for SGA and MGA are limited to 110 due to the more complex processes. The *penalty* for SGA and MGA is set to 0.000035, based on the smallest parameter value of the solution selected from DMA.

Fig. 4 shows the *OF* and *EI* values in SGA, MGA and DMA for the selected evolution period because of the same reason as in subection 5.1. MGA shows lower *OF* and *EI* values. The solution of DMA at generation 11 is selected as its optimal solution since solutions in latter generations, as seen, do not give significant improvement in *OF* or *EI* values. The *OF* and *EI* values for some solutions using DMA in interval of 5 generations can be inferred from Table 2. The results of performance measures are given in Table 3.

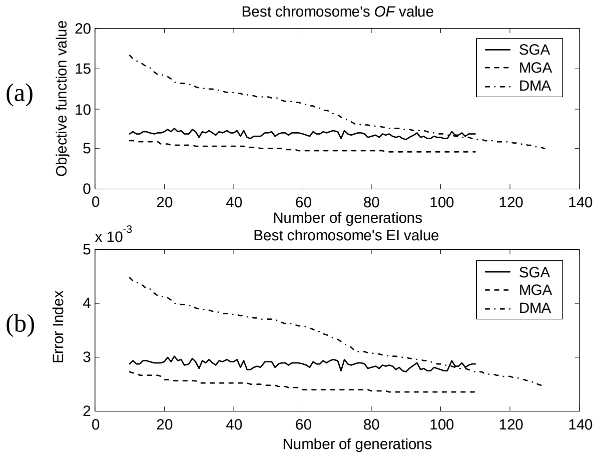


Fig. 4 (a) *OF* value versus the number of generations (b) *EI* versus the number of generations by using different algorithms for gas furnace data

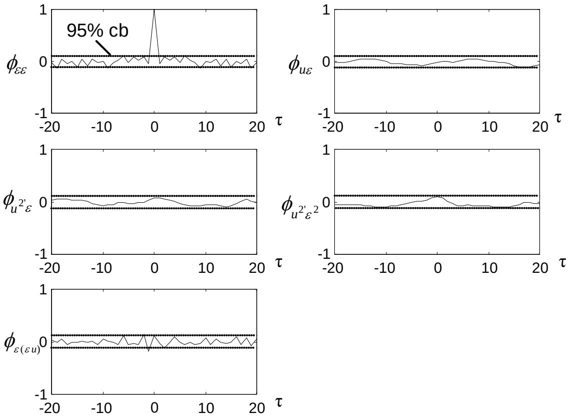
Table 2: Objective function and error index of best chromosomes in 5 generation intervals found in DMA for gas furnace data

Number of generations (number of regressors)	**Objective function**	**Error index**
5	22.3637	5.1889
10	16.6209	4.4733
15	15.2241	4.2812
20	14.1296	4.1245
25	13.1514	3.9791
30	12.6240	3.8985

Table 3: Performance measures of SGA, MGA and DMA for gas furnace data

	SGA	**MGA**	**DMA**
Number of regressors	113	139	11
Error Index	0.0029	0.0023	0.0044
Computation time (seconds)	1546.032	2063.906	1092.625

Although the *EI* in DMA is higher, this is due to the parsimony concern in the study. If the selected chromosome has been at the 108th and 129th generation, the *EI* given by chromosomes in DMA is 0.0028 and 0.0025, respectively. These are about the same as found in SGA and MGA. It is clear that DMA has advantages over computational need and parsimonious model search. Fig. 5 gives the correlation results of the selected solution from DMA where only slight correlations are found. Correlation tests to model with 58 regressors in DMA give all results within confidence bands. Fig. 6 shows the superimposition of actual and predicted outputs. The maximum error is 1.4643, while the percentage of error is 9.83%.



95% cb: 95% confidence band

Fig. 5 Correlation results of selected solution in DMA for gas furnace data

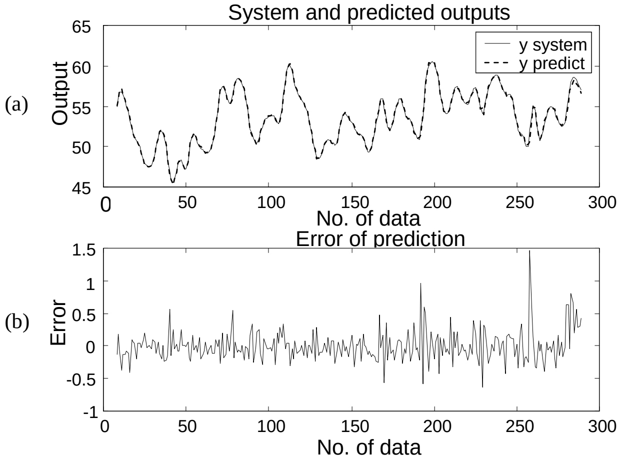


Fig. 6 (a) System and predicted outputs (b) Error of prediction using selected model from DMA for gas furnace data

5.3 Discussion

Based on the study of simulated models and real plant data, DMA is able to demonstrate a convincing and consistent capability in model structure selection. Since its solutions can be retrieved in an ascending quantity of gene 1 in a chromosome, careful trade-off between parsimony and accuracy is possible.

The usage of a penalty function in SGA and MGA requires a selection of penalty coefficient. Inappropriate penalty coefficient can be divergent in effective solution selection as the detection of significant variables and terms become ambiguous. SGA and MGA also requires the initialization of a random population in the procedures. This becomes a deficiency factor in the algorithm when non-parsimonious chromosomes begin to dominate the population. This dominance contributes to premature convergence, as parsimonious chromosomes are unable to be formed. SGA and MGA also utilizes probabilities that make the outcome of a simulation trial always differ from another, constituting a lack of consistency.

6. Conclusion

DMA functions in an organized deterministic pattern. It selects the most critical terms and variables in its advance of iterations with consideration of parameter estimation and without a rigid dependence of a fixed significance threshold. The simulation study shows that DMA is much simpler in the sense that it does not require too many genetic operations, parameter controls and penalty mechanism but is able to find good solutions quicker, with more efficient constraint-compliance and, additionally, is more accurate in some cases than SGA and MGA.

Acknowledgement

The authors would like to acknowledge all the support of Universiti Teknikal Malaysia Melaka and Universiti Teknologi Malaysia throughout this research especially for the UTeM-SLAB sponsorship.

References

[1] Ahmad, R., Jamaluddin, H. and Hussain, M. A., "Model structure selection for a discrete-time non-linear system using genetic algorithm", *Proceedings of the Institution of Mechanical Engineers - Part I: Journal of Systems and Control Engineering*, Vol. 218, No. 2, 2004, pp. 85-98.

[2] Ahmad, R., Jamaluddin, H. and Hussain, M. A., "Selection of a model structure in system identification using memetic algorithm", *Proceedings of the Second International Conference on Artificial Intelligence in Engineering & Technology*, Vol. 2, Kota Kinabalu, Sabah, Malaysia, Aug 3-5, 2004, pp. 714-720.

[3] Alves da Silva, A. P. and Abrão, P. J. "Applications of evolutionary computation in electric power systems", *Proceedings of the 2002 Congress on Evolutionary Computation*, Vol. 2, Honolulu, Hawaii, USA, May 12-17, 2002, pp. 1057-1062.

[4] Aytug, H., Khouja, M. and Vergara, F. E., "Use of genetic algorithms to solve production and operations management problems: a review", *International Journal of Production Research,* Vol. 41, No. 17, 2003, pp. 3955-4009.

[5] Billings, S. A. and Voon, W. S. F., "Correlation based model validity tests for non-linear models", *International Journal of Control*, Vol. 44, No. 1, 1986, pp. 235-244.

[6] Billings, S. A. and Wei, H. L. "A new class of wavelet networks for nonlinear system identification", *IEEE Transactions on Neural Networks,* Vol. 16, No. 4, 2005, pp. 862-874.

[7] Box, G. E. P., Jenkins, G. M. and Reinsel, G. C., *Time Series Analysis Forecasting and Control,* 3rd ed., Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[8] Eiben, A. E. and Smit, S. K., "Parameter tuning for configuring and analyzing evolutionary algorithms", *Swarm and Evolutionary Computation,* Vol. 1, 2011, pp. 19-31.

[9] Eiben, A. E. and Smith, J. E., *Introduction to Evolutionary Computing*, Springer-Verlag, Berlin, 2003.

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

[11] Grefenstette, J. J., "Optimization of control parameters for genetic algorithms", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 16, No. 1, 1986, pp. 122-128.

[12] Hong, X., Mitchell, R. J., Chen, S., Harris, C. J., Li, K. and Irwin, G.W., "Model selection approaches for non-linear system identification: a review", *International Journal of Systems Science*, Vol 39, No 10, 2008, pp. 925-946.

[13] Jamaluddin, H., Abd Samad, M. F., Ahmad, R. and Yaacob, M. S., "Optimum grouping in a modified genetic algorithm for discrete-time non-linear system identification", *Proceedings of the Institution of Mechanical Engineers - Part I: Journal of Systems and Control Engineering*, Vol. 221, No. 7, 2007, pp. 975-989.

[14] Jenkins, G. M. and Watts, D. G., *Spectral Analysis and Its Applications*, Emerson Adams Pr Inc, 1998.

[15] Kanarachos, A., Koulocheris, D. and Vrazopoulos, H., "Evolutionary algorithms with deterministic mutation operators used for the optimization of the trajectory of a four-bar mechanism", *Mathematics and Computers in Simulation*, Vol. 63, No. 6, 2003, pp. 483-492.

[16] Ljung, L., *System Identification: Theory for the User*, 2nd ed., Prentice Hall PTR, Upper Saddle River, New Jersey, 1999.

[17] Rodríguez-Vázquez, K., Fonseca, C. M. and Fleming, P. J., "Identifying the structure of nonlinear dynamic systems using multiobjective genetic programming", *IEEE Transaction on Systems, Man and Cybernetics – Part A: Systems and Humans*, Vol. 34, No. 4, 2004, pp. 531-545.

[18] Shilane, D., Martikainen, J., Dudoit, S. and Ovaska, S. J., "A general framework for statistical performance comparison of evolutionary computation algorithms", *Information Sciences*, Vol. 178, No. 14, 2008, pp. 2870-2879.

[19] Sivanandam, S. N. and Deepa, S. N., *Introduction to Genetic Algorithms*, Springer-Verlag, Berlin, 2008.

[20] Subudhi, B. and Jena, D., "A differential evolution based neural network approach to nonlinear system identification", *Applied Soft Computing*, Vol. 11, No. 1, 2011, pp. 861-871.

[21] Wei, H. L., Billings, S. A. and Liu, J., "Term and variable selection for non-linear system identification", *International Journal of Control*, Vol. 77, No. 1, 2004, pp. 86-110.

[22] Yao, X. and Xu, Y., "Recent advances in evolutionary computation", *Journal of Computer Science and Technology*, Vol. 21, No. 1, 2006, pp. 1-18.

Md Fahmi Abd Samad received his Bachelor in Mechanical Engineering, Masters in Engineering Management and Ph.D. in Mechanical Engineering from Universiti Teknologi Malaysia, Johore, Malaysia in 2000, 2002 and 2009, respectively. Currently, he works as a Senior Lecturer in Faculty of Mechanical Engineering, Universiti Teknikal Malaysia Melaka, and is a Graduate Member of Institution of Engineers, Malaysia. His interests are in system identification and evolutionary computation.

Hishamuddin Jamaluddin received his B.Sc., M.Sc. and Ph.D. from the Department of Control Engineering, Sheffield University, United Kingdom in 1982, 1985 and 1991, respectively. He is currently a Professor in the Department of Applied Mechanics, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia. His research interests include non-linear system modelling, system identification, neural networks, adaptive fuzzy models, genetic algorithm, neuro-fuzzy and active force control. His profile can be reached at http://www.fkm.utm.my/~hishamj.

Robiah Ahmad obtained her B.Sc. in Electrical Engineering from University of Evansville, Indiana, USA in 1988, M.Sc. in Information Technology for Manufacture from University of Warwick, United Kingdom and Ph.D. in Mechanical Engineering from Universiti Teknologi Malaysia. She is currently a Senior Lecturer in Department of Applied Mechanics, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia. Her research interests are system identification, evolutionary computation, mechatronics and instrumentation and control.

Mohd Shafiek Yaacob obtained the Bachelor of Science in Mechanical Engineering degree from Texas Tech University, USA in 1985 and the Master of Mechanical Engineering degree form Rice University, USA in 1987. He later obtained the Doctor of Philosophy degree from Universiti Teknologi Malaysia in 2004. He is currently an Associate Professor at the Faculty of Mechanical Engineering, Universiti Teknologi Malaysia. His areas of interest include control of engineering systems, mathematical modelling, system identification and simulation of systems. He is also, currently, involved in the study and simulation of artificial human type motions.

Abul K. M. Azad obtained his B.Sc. and M.Sc. in Applied Physics and Electronics from University of Dhaka, Bangladesh in 1986 and 1988, respectively. He graduated his Ph.D. in Control and Systems Engineering from University of Sheffield, UK in 1994. He currently serves as a Professor in Northern Illinois University. He is in leadership role with professional bodies as well as managing editorial reponsibilities. He is a reviewer for national and international funding agencies, and is a program evaluator for the Accreditation Board for Engineering and Technology (ABET). His profile can be accessed from http://www.niu.edu/~azad.