Praise Worthy Prize

International Review of Mechanical Engineering (I.RE.M.E.), Vol. 15, N. 2
ISSN 1970 - 8734 *February 2021*

Comparison of Crossover in Genetic Algorithm for Discrete-Time System Identification

Farah Ayiesya Zainuddin[1], Md Fahmi Abd Samad[1, 2]

Abstract – *System identification is a process where a mathematical model is derived in order to explain dynamical behaviour of a system. One of its step is model structure selection and it is crucial that, in this step, an adequate model i.e. a model with a good balance between parsimony and accuracy of the model is selected in approximating the system. Genetic algorithm (GA), a method known for optimisation, is used for selecting a model structure. GA is known to be able to reduce much computational burden. This paper investigates the effect of different types of crossover, namely, single-point, double-point, multiple-point and uniform crossover, within GA in producing an optimum model structure for system identification. This was carried out using a computational software on a number of simulated data. As a conclusion, using Akaike Information Criterion as objective function, single point crossover produces the model with the best balance in most of the tests.*

Keywords: *Genetic Algorithm, System Identification, Model Structure Selection, Crossover, Discrete-Time Model, ARX, NARX*

Nomenclature

ε	Model residual
a	Model parameter
AIC	Akaike Information Criterion
ARX	AutoRegressive with eXogenous variable
d.c.	Constant level
e	Noise variable
EI	Error Index
GA	Genetic Algorithm
k	Number of model parameter
L	Maximum number of possible terms
l	Model nonlinearity
lchrom	Chromosome length
N	Number of observation data
NARX	Nonlinear AutoRegressive with eXogenous variable
n_k	Time lag
n_u	Maximum order of input lag
n_y	Maximum order of output lag
OF	Objective Function
p_c	Crossover probability
p_m	Mutation probability
RSS	Residual Sum of Square
t	Time instant
u	Input variable
y	Output variable
$\hat{y}$	k-step-ahead predicted output variable

I. Introduction

System Identification (SI) is a method of determining a mathematical relation between variables and terms of a process based on observed input-output data with the aim to enable better control of a system [1]. Modelling of system can be divided into continuous-time and discrete-time modelling. Many real-world systems, for example, in the fields of mechanics, electricity, chemistry, economics, biology and ecology are dynamic systems. In these systems, the response of the system depends not only on input but also on its internal state. Although these systems warrant a continuous-time model rather than a discrete-time model, it is practical that data acquisition is performed under the assumption that the variables or terms are interconnected by instants of time.

Variables refer to different components of a system such as inputs, outputs and disturbances whereas terms refer to different dimensions or variables transformation [2]. System identification is an essential work for control engineering. According to an identified system model, a controller can be designed to meet the required specification by means of different control methods.

Once the system structure is known beforehand, the remaining problem is how to identify accurately the corresponding parameters [3]. There is an abundance of engineering areas where system identification have been applied, including for prediction of wind turbine power output [4], modelling of flexible beam structure [5] and modelling of hydropower inverter system [6]. Overall, SI is done through four main steps involving data acquisition, selection of the model structure, parameter estimation, and model validation [7]. An optimal model is normally described as having adequate predictive accuracy in the response to the system, yet parsimonious in structure. A parsimonious model structure is preferred, since system analysis and control is made easier with

 https://doi.org/10.15866/ireme.v15i2.19726

fewer variables and/or terms [2]. Artificial intelligence methods, such as GA, are innovative ways of performing SI. GA is based on Darwin's theory of natural evolution.

The genetic algorithm can work efficiently and deliver better results in the area of optimisation and search problems [8]-[10]. GA imitates evolutionary processes.

By applying the principle of survival of the fittest, it conforms to the metaphor of natural biological evolution. In the SI context, it starts with the genetic encoding into chromosomes of potential input-output relationship from the representations of a system. Therefore, a chromosome is a string of code representing a model solution. Each string position is referred to as gene. In a binary-represented GA, the variables and terms of a discrete-time system are represented by the chromosome genes as bit 1 for existence and bit 0 for omission. The encoding is followed by selection process. The selection process refers to the process of selecting chromosomes to go through genetic operations by a set of chromosomes (called population). These genetic operators are crossover and mutation. The crossover operator operates two chromosomes by exchanging parts of their structure.

There are many varieties of crossover type e.g. single-point crossover, double-point crossover, multiple-point crossover and uniform crossover [11]. Chang [3] makes a new use of the GA in the identification of systems to solve optimally off-line PID controllers. Samad & Nasir [12] uses GA to identify a discrete-time system based on a novel information criterion. These researchers recorded successful usage of GA in SI. Some researchers introduced new crossover types to solve various optimisation problems with good quality solutions [10], [13]-[16]. Yet some researchers still use traditional crossover to deal with their target problems [9], [16]-[18]. The purpose of this paper is to clarify the performances of traditional crossovers known in genetic algorithm, specifically for use in binary representation of model structure selection problem in discrete-time system identification.

The next sections are as follows. Section II explains about methodology. Section III explains the results.

Section IV provides discussion and lastly Section V concludes the paper along with recommendation of future works.

II. Methodology

The following subsections explain the problem representation in greater detail, genetic operators that are used in the study, simulated models and characteristics, evaluation of chromosome and, lastly, indicators in comparing the performances of the crossovers.

II.1. Problem Representation

The representation stage refers to the mapping of problem components into an individual representation.

This can be achieved either by using the real values of solutions also known as the phenotype or by using different representative values such as binary ones also known as the genotype [19]. While different genetic operators apply to different types of representations, the transformation between binary and real valued vectors is relatively easy [20]. In addition to binary coding, Gray coding, an alternative representation of {0, 1} cardinality, was also used [21], [22]. Upon completion of the representation the population is initialized. Most population initialization is done randomly and very few researchers have used heuristic information because it depends on the problem area [23]. Luh and Rizzoni [24] and Luh and Wu [25], who used binary numbers for model structure selection, recommended a representation method where a solution is represented by a fixed number of allowable substrings. Each substring is a term which is based on a decoding scheme. However, the number of regressors for the model is fixed in this method and thus neither further accuracy achievement is enabled by more inclusion of regressors or parsimony search. An alternative method is to assume that each component of a representation represents a variable or a term of a model in which the allele or representation is either 1 for present or 0 for omission [26]-[29]. As an example, by supposing that a system represented as a nonlinear autoregressive with exogenous variable (NARX) model has a nonlinearity, $l = 2$, maximum order of lag for input, $n_u = 2$, maximum order of lag for output, $n_y = 2$ and time lag, $n_k = 1$, the number of possible terms in the model, L, is found to be 15 (refer [30] for calculation). The output, $y(t)$ for the system is represented by the following linear-in-the-parameter equation:

$$\begin{aligned} y(t) = {} & a_1 + a_2 y(t-1) + a_3 y(t-2) + a_4 u(t-1) \\ & + a_5 u(t-2) + a_6 y^2(t-1) + a_7 y(t-1) y(t-2) \\ & + a_8 y(t-1) u(t-1) + a_9 y(t-1) u(t-2) \\ & + a_{10} y^2(t-2) + a_{11} y(t-2) u(t-1) \\ & + a_{12} y(t-2) u(t-2) + a_{13} u^2(t-1) \\ & + a_{14} u(t-1) u(t-2) + a_{15} u^2(t-2) \end{aligned} \tag{1}$$

where a_i is the parameter value of regressor i ($1 \le i \le L$) with a_1 referred as the parameter for the system constant level also called d.c. level. Based on the above information, a binary chromosome representation of length $lchrom = 15$ is generated. The search space with this representation is $2^{lchrom}-1$, meaning there are 32767 possible models to choose from. The chromosome [110 100 001 000 100] represents the following model, based on the model given in Equation (1) and certain regressor sequence coding:

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

II.2. Genetic Operators

GA can be considered as a multi-directional search method to solve problems as inheritance throughout an

evolution includes three evolutionary operations: reproduction, crossover, and mutation. This means it has more likelihood of escape from a local minimum.

Traditional gradient method however searches for solution in the search space only from a single direction [3]. All the variables of interest must be encoded as binary digits (genes) in the traditional binary-coded GA, and a collection of binary digits further form a string (chromosomes). There are several notable traditional crossover types [31]-[32]:

1. Single-point: Parent fragmentation and then combination of the parents at a crossover point to create the offspring;
2. Double-point: Use two crossover points where the chromosome sections are swapped between points;
3. Multiple-point: The number of crossover points is selected randomly;
4. Uniform: Provides the uniformity of combining the bits of both parents.

II.3. Simulated Models

The study began with simulation of models to produce four sets of single-input-single-output data. These simulated models were of the type autoregressive with exogenous variable (ARX) and NARX. The models were denoted as Model 1, Model 2, Model 3 and Model 4 and each model was assumed to have d.c. level (constant).

The models are listed below, written as linear regression models, together with the number of correct regressors, the search specification and number of possible models for ease of comparison:

Simulated Model 1:

$$\begin{aligned} y(t) &= 0.5y(t-1) - 0.2y(t-4) + 0.5y(t-8) \\ &\quad +0.6u(t-2) - 0.2u(t-9) + e(t) \end{aligned} \tag{3}$$

Number of correct regressor = 5 out of 20, search space specification: l=1, n_y=8, n_u=8, n_k=2, number of possible model = 1048575.

Simulated Model 2:

$$\begin{aligned} y(t) &= 0.4y(t-1) + 0.4u(t-1) - 0.6u(t-3) \\ &\quad -0.7y(t-1)u(t-1) - 0.2y^2(t-2) + \\ &\quad +0.2u^2(t-3) + e(t) \end{aligned} \tag{4}$$

Number of correct regressor = 6 out of 20, search space specification: l=2, n_y=2, n_u=3, n_k=1, number of possible model = 1048575.

Simulated Model 3:

$$\begin{aligned} y(t) &= 0.4y(t-3) + 0.3u(t-1) + \\ &\quad +0.7y(t-2)u(t-1) + \\ &\quad +0.1y(t-3)u(t-2) + \\ &\quad -0.5u(t-1)u(t-2) + \\ &\quad -0.4u(t-1)u(t-3) + e(t) \end{aligned} \tag{5}$$

Number of correct regressor = 6 out of 27, search space specification: l=2, n_y=3, n_u=3, n_k=1, number of possible model = 134 217 727.

Simulated Model 4:

$$\begin{aligned} y(t) &= 0.8u(t-1) + 0.5y(t-1)y(t-2) \\ &\quad +0.1y(t-1)u^2(t-1) + \\ &\quad -0.1y(t-2)u(t-1)u(t-2) + \\ &\quad -0.4u^2(t-1)u(t-2) + \\ &\quad +0.2u(t-1)u^2(t-2) + e(t) \end{aligned} \tag{6}$$

Number of correct regressor = 6 out of 34, search space specification: l=3, n_y=2, n_u=2, n_k=1, number of possible model = 17 179 869 183.

Five hundred data points were generated from all models. The input $u(t)$ was generated from a random uniform distribution in the interval [-1, 1] to represent white signal, while noise $e(t)$ was generated from a random uniform distribution [-0.01, 0.01] to represent white noise. The specification of the algorithm was fixed for all models where the population size is set to 200, the maximum generation was 100, the mutation probability, p_m = 0.01 and the crossover probability, p_c = 0.6. This paper used roulette-wheel selection. The elitist strategy was also used so that the best chromosome based on evaluation was always preserved. In GA search, the model structure was first identified and then followed by parameter estimation. The parameter estimation method used was the least squares method.

II.4. Evaluation

In assigning fitness to the chromosomes for selection to proceed, Akaike Information Criterion (AIC) was used [33], [34]. The highest fitness was assigned to the model that minimizes the criterion the most, here called Objective Function (OF), while 0 fitness for the highest OF. This information criteria is widely used for selecting model structures. AIC is composed as:

$$\mathrm{AIC} = N \ln \frac{\mathrm{RSS}}{N} + 2k \tag{7}$$

where N is the number of observations, RSS (the Residual Sum of Square) is the maximised value of the likelihood function for the estimated model and k is the number of parameters in the model. Maximised value of the likelihood function for the estimated model (RSS) is defined in the formula below:

$$\mathrm{RSS} = \sum_{t=k}^{N} \varepsilon^2(t) = \sum_{t=k}^{N} \left(y(t) - \hat{y}(t) \right)^2 \tag{8}$$

where $\varepsilon(t)$ is the residual, $\hat{y}(t)$ and $y(t)$ are the k-step-ahead predicted output and actual output value at time t, respectively, and N is the number of data. The identification simulation run were made 15 times for

each crossover in order to come up with an average performance of crossover.

II.5. Performance Indicator

Two indicators of performance were used, i.e. Objective Function (OF) and Error Index (EI), when comparing the crossover performance. The best chromosome's OF value was the OF value of the elitist.

The OF calculation was as given in equation (7). The error index referred to the square root of the sum of squared error of the elitist divided by the sum of actual output squared. Its calculation is as follows:

$$\mathrm{EI} = \sqrt{\frac{\sum\left(y(t)-\hat{y}(t)\right)^2}{\sum y^2(t)}} \tag{9}$$

where y is the actual output value and $\hat{y}$ is the k-step ahead predicted output obtained from least-squares estimation. While OF emphasized accuracy of prediction and parsimony of the model, simultaneously, EI only measured the accuracy of the model [35].

III. Results

Figures 1 to 4 show the simulation results made using GA with AIC as objective function for all models. The results are plotted graphs of four evolutions consisting of three data for each evolution.

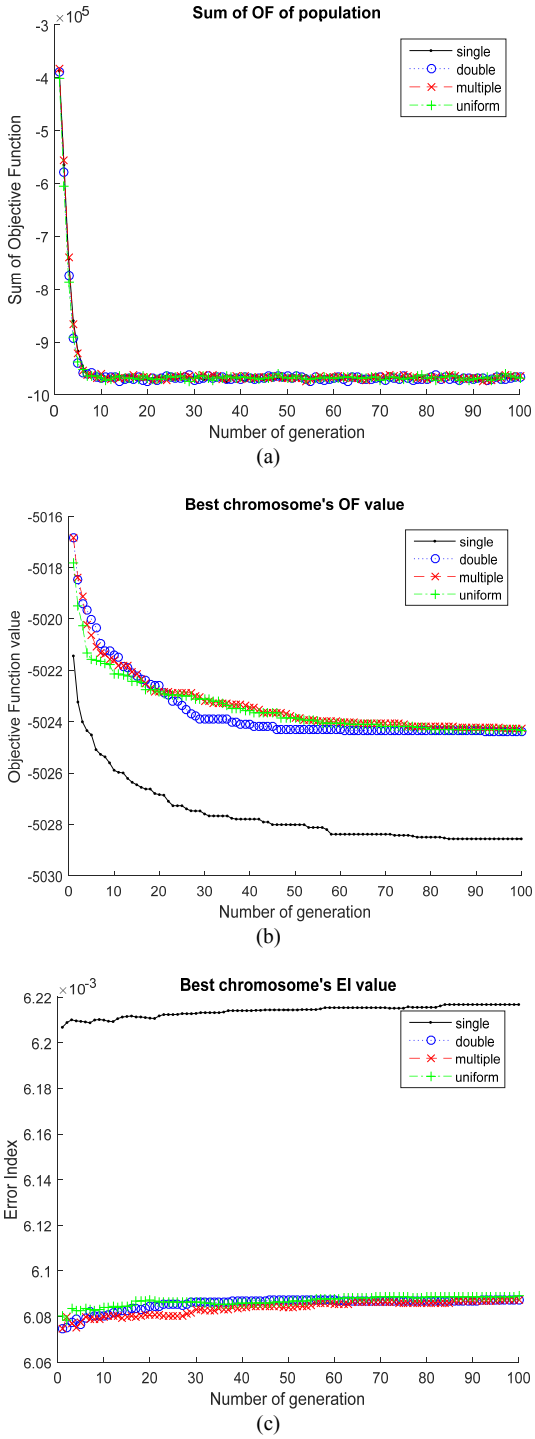
It should be noted that all the graphs are the average result from 15 runs. The results are in term of the sum of OF in population, the best chromosome's OF value and the best chromosome's EI value.

III.1. Simulated Model 1

The results for simulated model 1 are presented in Figs. 1(a), 1(b) and 1(c). Fig. 1(a) displays almost the same trend in graph pattern for all 4 crossover types, indicating similar speed of convergence. Fig. 1(b) shows that single-point produces model with the lowest OF compared to the other crossover types. However, when seen with Figure 1(c) that shows data of the best chromosome's EI value, the model from single-point crossover has higher EI value. Fig. 1(c) also shows that model with the lowest EI come from the multiple-point crossover.

The binary representation of the original model was [100 100 010 001 000 000 10]. The lowest OF model had the binary [100 110 010 001 100 010 10]. Eight regressors were selected. Other than all correct regressors, the model included $y(t\text{-}5)$, $u(t\text{-}3)$ and $u(t\text{-}7)$.

The lowest EI model had the binary [100 100 010 101 000 001 10]. It included $y(t\text{-}10)$ and $u(t\text{-}8)$. Clearly, this indicates that the latter model included regressors that contribute much to the accuracy but its balance between accuracy and parsimony was not as good as the former.

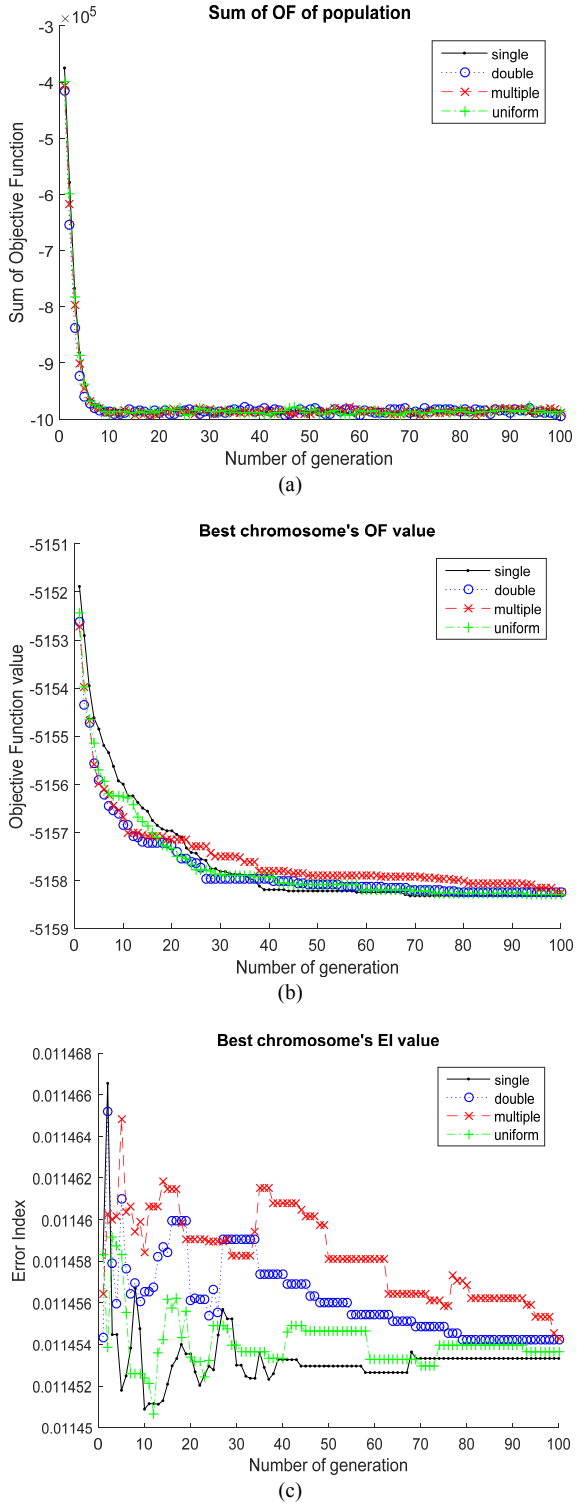


Figs. 1. (a) Sum of OF for simulated model 1, (b) Best chromosome's OF value for simulated model 1, (c) Best chromosome's EI value for simulated model 1

III.2. Simulated Model 2

Figs. 2(a), 2(b) and 2(c) show the results for simulated model 2. Fig. 2(a) also shows nearly the same trend in graph pattern for all 4 crossover types. Figs. 2(b) and 2(c) indicate that single-point crossover produced the model with the lowest OF and EI compared to the other crossover types. It was still unable to select the same

model as the original model as, other than the original regressors, it included $y^2(t\text{-}1)$ and $y(t\text{-}1)y(t\text{-}2)$, totalling to 8 regressors. Even though one may say that the single-point addressed the required balance between parsimony and accuracy, its achievement was only seen somewhere close to the 40th generation.



Figs. 2. (a) Sum of OF for simulated model 2, (b) Best chromosome's OF value for simulated model 2, (c) Best chromosome's EI value for simulated model 2

III.3. Simulated Model 3

Figs. 3(a), 3(b) and 3(c) show the results for simulated model 3. Fig. 3(a) shows almost identical graph pattern for all 4 crossover types. From the naked eyes, it seemed that the graph pattern was the same for all types of crossover in Fig. 3(b).

However, when zooming in the graph, Fig. 3(b) shows that double-point crossover produced a model with the lowest OF compared to the other crossover types. Fig. 3(c) shows that the model with the lowest EI came from multiple-point crossover.

Examining the models, the models with the lowest OF had 7 regressors. Other than the correct regressors, it included $u^2(t\text{-}1)$. The model with lowest EI achieved such a low value, simply because it included $y(t\text{-}1)y(t\text{-}3)$, $y^2(t\text{-}3)$, $u^2(t\text{-}2)$ and $u^2(t\text{-}1)$. So, other than the correct regressors, the inclusion made up 10 regressors – a non-parsimonious one.

III.4. Simulated Model 4

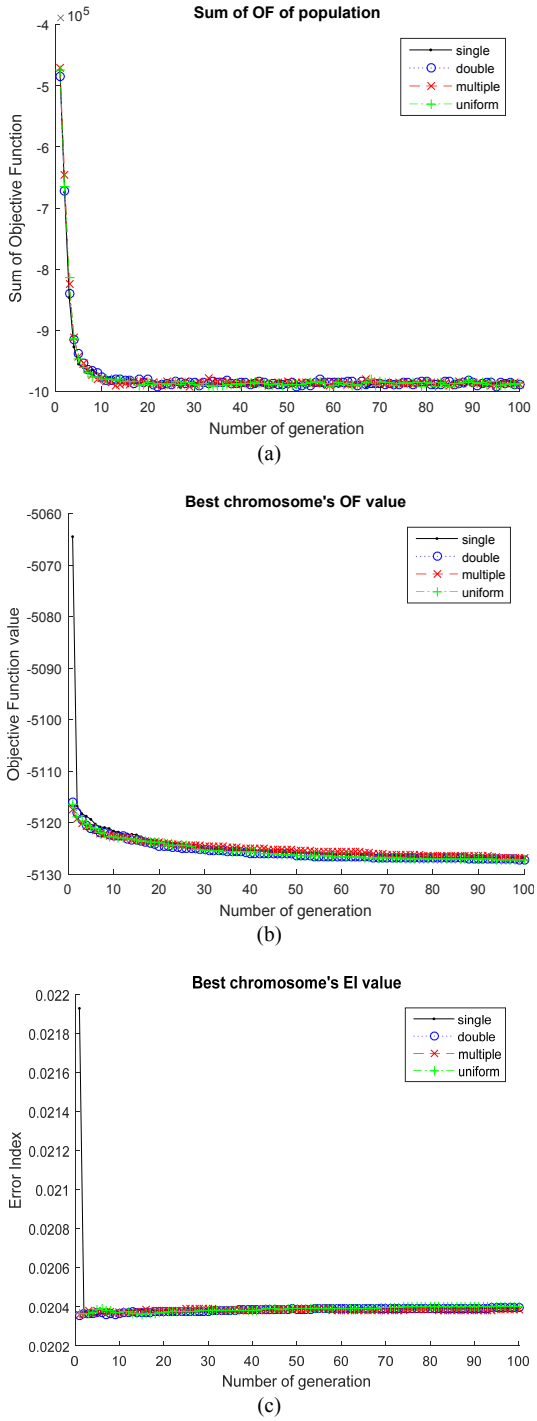
Figs. 4(a), 4(b) and 4(c) show the results for simulated model 4. Fig. 4(a) shows almost identical graph pattern for all 4 crossover types. Fig. 4(b) shows that single-point produced the model with the lowest OF compared to the other crossover types. Fig. 4(c) shows that the model with the lowest EI came from the multiple-point crossover.

Multiple-point crossover model had good EI values but the OF values were not as good as single-point and double-point. This was due to the fact that it selected 14 regressors. On the other hand, the model from the single-point crossover had 8 regressors. Other than the correct regressors, it included $y^2(t\text{-}1)$ and $y(t\text{-}1)y^2(t\text{-}2)$.

IV. Discussion

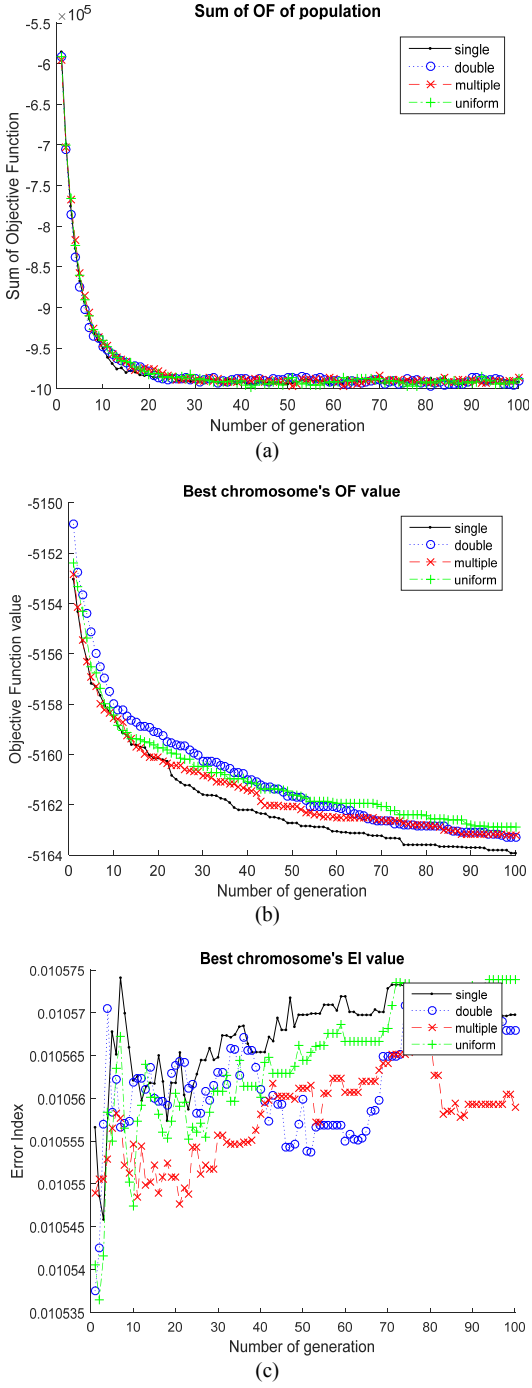
For all simulated models, the graphs of population sum of OF were quite the same, concerning respective crossover types. Although the single-point crossover performed well for simulated model 1 and 4, it is noted that when tried in simulated model 2 and 3, the single-point crossover was a bit "late" in finding a model with low OF value.

Yet, single-point crossover was still able in finding a model with lower or equally low OF as the others as evolution continues. This presents an area for the argument of when to stop the evolution. Stopping the evolution too early will definitely presents a different outcome of the study. Furthermore, the graphs of sum of OF value (for simulated model 2 and 3), indicates that the population in single-point crossover was already low, generally, but proper mating had not been achieved to allow the production of lower OF model. In other words, the population was already filled with models of low OF but unsuitable mating caused it to be slow in getting lower OF models. Proper mating between chromosomes is an additional way, aside from crossover and mutation, in enabling better exploration of search space.



Figs. 3. (a) Sum of OF for simulated model 3, (b) Best chromosome's OF value for simulated model 3, (c) Best chromosome's EI value for simulated model 3

High OF model will usually give low EI and vice versa. This is because a non-parsimonious model (containing many regressors) provide better accuracy (hence, low EI) whilst being highly penalized (hence, high OF). This is seen in simulated model 1 but not in simulated model 2, possibly because the selected model in single-point crossover was already a parsimonious model. Another interesting note was of uniform crossover in simulated model 4. It had high OF, yet also high EI compared to the others. Table I summarizes the findings.



Figs. 4. (a) Sum of OF for simulated model 4, (b) Best chromosome's OF value for simulated model 4, (c) Best chromosome's EI value for simulated model 4

TABLE I
OVERALL PERFORMANCE OF CROSSOVER

Simulated Model	Crossover with the best performance		
	Sum of OF	Lowest OF	Lowest EI
Model 1	Equally the same	Single	Multiple[a]
Model 2	Equally the same	Single[b]	Single
Model 3	Equally the same	Double	Multiple
Model 4	Equally the same	Single	Multiple

[a] Multiple is slightly more accurate
[b] Single is slightly better than the others

V. Conclusion

GA is an efficient approach to computing which is effectively used in various issues. The output depends on the encoding scheme and the choice of genetic operators, in particular the operators for selection, crossover and mutation. Before applying the operators to solve a new problem, it is essential to overview the search space and to understand its modality. This study aimed at identifying the performance of different types of crossover within the limitation of using binary representation for absence and presence of regressors in discrete-time system identification. In this study, the single-point crossover was the preferred choice over the others as it was able to produce a model that was adequately accurate and parsimonious. The multiple-point crossover was the type that produced most of the accurate models, but by being over-parameterized.

Generally, the reason that crossover did not perform well was perhaps due to unsuitable type and/or parameter values when setting up GA. This includes, among others, the crossover probability, mutation probability and number of maximum generation. More rigorous analysis could be made to identify such weaknesses and thus improve crossover capability. Future research of the study may be directed towards developing and incorporating a mating technique in GA so that more efficient and shorter processing time is needed.

Acknowledgements

The authors would like to thank Ministry of Higher Education Malaysia and Universiti Teknikal Malaysia Melaka (UTeM) for their financial support through FRGS/1/2018/TK03/UTEM/02/13 grant, technical advice and suggestion.

References

[1] V. Duong, A. R. Stubberud, System identification by genetic algorithm. *Proc. IEEE Aerospace Conference*, paper 270, March 2002, pp. 2331–2337.

[2] M. F. Abd Samad, A. R. M. Nasir, Performance of parameter-magnitude based information criterion in identification of linear discrete-time model, *Journal of Fundamental and Applied Sciences*, Vol. 10, n. 3S, pp. 345-354, 2018.

[3] W. D. Chang, Nonlinear system identification and control using a real-coded genetic algorithm, *Applied Mathematical Modelling*, Vol. 31, n. 3, pp. 541-550, 2007.

[4] Qatamin, R., Mohamed, O., Abu Elhaija, W., Prediction of Power Output of Wind Turbines Using System Identification Techniques, (2020) *International Review on Modelling and Simulations (IREMOS)*, 13 (1), pp. 43-51. doi: https://doi.org/10.15866/iremos.v13i1.17713

[5] Ting, R., Mat Darus, I., Sahlan, S., Ab Talib, M., Optimized Modeling of Flexible Beam Structure with Pole-Zero Estimation, (2019) *International Review of Mechanical Engineering (IREME)*, 13 (3), pp. 148-161. doi: https://doi.org/10.15866/ireme.v13i3.15922

[6] Vacharapanich, V., Chenvidhya, D., Pratinthong, N., Kirtikara, K., Identification Estimated with Bat Search Algorithms for Modeling of Inverter System, (2018) *International Review of Electrical Engineering (IREE)*, 13 (2), pp. 80-88. doi:https://doi.org/10.15866/iree.v13i2.14650

[7] M. F. Abd Samad, H. Jamaluddin, R. Ahmad, M. Shafiek, Deterministic Mutation-Based Algorithm for Model Structure Selection in Discrete-Time System Identification, *International Journal of Intelligent Control and Systems*, Vol. 16, n. 3, pp. 182-190, 2011.

[8] A. Mazahery, M. O. Shabani, Assistance of novel artificial intelligence in optimization of aluminum matrix nanocomposite by genetic algorithm, *Metallurgical and Materials Transactions A*, Vol. 43, n. 13, pp. 5279-5285, 2012.

[9] A. Bala, and A. K. Sharma, A comparative study of modified crossover operators, *Third International Conference on Image Information Processing (ICIIP)*, December 2015, pp. 281-284.

[10] B. Koohestani, A crossover operator for improving the efficiency of permutation-based genetic algorithms, *Expert Systems with Applications*, Vol. 151, 113381, 2020.

[11] Abd Samad, M., Evolutionary Computation in System Identification: Review and Recommendations, (2014) *International Review of Automatic Control (IREACO)*, 7 (2), pp. 208-216.

[12] M. F. Abd Samad, A.M. Nasir, Discrete-time system identification based on novel information criterion using genetic algorithm, *Journal of Fundamental and Applied Sciences*, Vol. 9, n. 7S, pp. 584-599, 2017.

[13] P. D. Thanh, H. T. T. Binh, B. T. Lam, New mechanism of combination crossover operators in genetic algorithm for solving the traveling salesman problem, In V. H. Nguyen, A. C. Le, V. N. Huynh (Eds.), *Knowledge and Systems Engineering*, Advances in Intelligent Systems and Computing, Vol. 326 (Springer, 2015, 367-379).

[14] Y. F. Jin, Z. Y. Yin, S. L. Shen, D. M. Zhang, A new hybrid real-coded genetic algorithm and its application to parameters identification of soils, *Inverse Problems in Science and Engineering*, Vol. 25, n. 9, pp. 1343-1366, 2017.

[15] G. Singh, N. Gupta, M. Khosravy, New crossover operators for real coded genetic algorithm (RCGA), *International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, November 2015, pp. 135-140.

[16] S. G. Varun Kumar, R. Panneerselvam, A study of crossover operators for genetic algorithms to solve VRP and its variants and new sinusoidal motion crossover operator, *International Journal of Computational Intelligence Research*, Vol. 13, n. 7, pp. 1717-1733, 2017.

[17] Y. Chen, M. Elliot, D. Smith, The application of genetic algorithms to data synthesis: a comparison of three crossover method, *International Conference on Privacy in Statistical Databases*, September 2018, 160-171.

[18] A. H. Wright, Genetic algorithms for real parameter optimization, In G. J. E. Rawlins (Ed.), *Foundations of genetic algorithms*, vol. 1, 1991, 205-218.

[19] T. Bäck, D. B. Fogel, Glossary, In T. Bäck, D.B. Fogel, Z. Michalewicz, (Eds.), *Evolutionary Computation 1: Basic Algorithms and Operators* (CRC Press, 2000, pp. xxi-xxxvii).

[20] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs* (3rd, revised and extended edition, Springer-Verlag Berlin Heidelberg, 1996).

[21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley Publishing Company, 1989).

[22] A. Czarn, C. MacNish, K. Vijayan, B. Turlach, Statistical exploratory analysis of genetic algorithms: the importance of interaction, *Proc. Congress on Evolutionary Computation*, vol. 2, June 2004, pp. 2288-2295.

[23] H. Aytug, M. Khouja, F.E. Vergara, Use of genetic algorithms to solve production and operations management problems: a review, *International Journal of Production Research*, Vol. 41, n. 17, pp. 3955-4009, 2003.

[24] G. C. Luh, G. Rizzoni, Nonlinear system identification using genetic algorithms with application to feedforward control design, *Proc. of 1998 American Control Conference (ACC)*, (IEEE Cat. No. 98CH36207), vol. 4, June 1998, pp. 2371-2375.

[25] G. C. Luh, C.Y. Wu, Non-linear system identification using genetic algorithms, *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, Vol. 213, n. 2, pp. 105-118, 1999.

[26] R. Ahmad, H. Jamaluddin, M. A. Hussain, 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, 2004, pp. 3-5, 2004.
[27] R. Ahmad, H. Jamaluddin, M. A. Hussain, Model structure selection for a discrete-time non-linear system using a genetic algorithm, *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, Vol. 218, n. 2, pp. 85-98, 2004.
[28] T. G.Yen, C. C. Kang, W. J. Wang, A genetic based fuzzy-neural networks design for system identification, *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1, pp. 672-678, 2005.
[29] A. Sakaguchi, T. Yamamoto, A study on system identification using GA and GMDH network, *29th Annual Conference of the IEEE Industrial Electronics Society* (IEEE Cat. No. 03CH37468), vol. 3, 2003, pp. 2387-2392.
[30] M. Korenberg, S. A. Billings, Y. P. Liu, P. J. McIlroy, Orthogonal parameter estimation algorithm for non-linear stochastic system, *International Journal of Control*, Vol. 48, n. 1, pp. 193-210, 1988.
[31] F. A. Zainuddin, M. F. Abd Samad, A Review of Crossover Methods and Problem Representation of Genetic Algorithm in Recent Engineering Applications, *International Journal of Advanced Science and Technology*, Vol. 29, pp. 759-769, 2020.
[32] A. J. Umbarkar, P.D. Sheth, Crossover operators in genetic algorithms: a review, *ICTACT Journal on Soft Computing*, Vol. 6, n. 1, pp. 1083-1092.
[33] H. Akaike, A New Look at The Statistical Model Identification, *IEEE Transactions on Automatic Control*, Vol. 19, n. 6, pp. 716-723, 1974.
[34] H. Akaike, Information Theory and An Extension of The Maximum Likelihood Principle, *Proceedings of the 2nd International Symposium on Information Theory, Supplement to Problems of Control and Information Theory*, 1972, pp. 267-281.
[35] M. F. Abd Samad, H. Jamaluddin, R. Ahmad, M. S. Yaacob, A. K. M. Azad, Effect of penalty function parameter in objective function of system identification, *International Journal of Automotive and Mechanical Engineering*, Vol. 7, pp.940-954, 2013.

Authors' information

[1]Faculty of Mechanical Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka.

[2]Center for Advanced Computing, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka.

Farah Ayiesya Zainuddin has earned Bachelor of Mechanical Engineering (Thermal-Fluids) and Master of Science in Mechanical Engineering from Universiti Teknikal Malaysia Melaka. Her research areas include genetic algorithm and aeronautics. She is currently pursuing a PhD degree in mechanical engineering.

Md Fahmi Abd Samad has earned his PhD degree in mechanical engineering from Universiti Teknologi Malaysia and currently serving as an associate professor in Faculty of Mechanical Engineering, Universiti Teknikal Malaysia Melaka. His research interests are in system identification, control engineering, and evolutionary computation. He is also a Chartered Engineer.
E-mail: mdfahmi@utem.edu.my

