

The saturation of population fitness as a stopping criterion in genetic algorithm

Foo Fong Yeng¹, Soo Kum Yoke², Azrina Suhaimi³

^{1,3}Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Johor, Malaysia

²Academy of Language Studies, Universiti Teknologi MARA Cawangan Negeri Sembilan, Malaysia

Article Info

Article history:

Received Jul 1, 2018

Revised Apr 24, 2019

Accepted May 6, 2019

Keywords:

Artificial intelligence

Genetic algorithm

Machine learning

Optimization

Stopping criterion

ABSTRACT

Genetic Algorithm is an algorithm imitating the natural evolution process in solving optimization problems. All feasible (candidate) solutions would be encoded into chromosomes and undergo the execution of genetic operators in evolution. The evolution itself is a process searching for optimum solution. The searching would stop when a stopping criterion is met. Then, the fittest chromosome of last generation is declared as the optimum solution. However, this optimum solution might be a local optimum or a global optimum solution. Hence, an appropriate stopping criterion is important such that the search is not ended before a global optimum solution is found. In this paper, *saturation of population fitness* is proposed as a stopping criterion for ending the search. The proposed stopping criteria was compared with conventional stopping criterion, *fittest chromosomes repetition*, under various parameters setting. The results show that the performance of proposed stopping criterion is superior as compared to the conventional stopping criterion.

Copyright © 2019 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Foo Fong Yeng,

Faculty of Computer and Mathematical Sciences,

Universiti Teknologi MARA Cawangan Johor, Kampus Pasir Gudang,

Jalan Purnama, Bandar Seri Alam, 81750, Masai, Johor, Malaysia.

Email: fofo931@uitm.edu.my; fongyengf@gmail.com

1. INTRODUCTION

In 1975, John Holland, inspired by the Darwin Evolution Theorem [1, 2], introduced an algorithm [3, 4] mimicking the process of genetic inheritance in evolution [4-6]. It was named as Genetic Algorithm (GA). Due to its adaptable competences, the algorithm applications in research areas are gigantic [7] where they are not only found in pure sciences such as engineering [8-10] but also in social sciences such as operation managements [11-14].

GA has been categorised under the family of meta-heuristic algorithms [15, 16] such as Tabu Search and Artificial Neural Network. Meta-heuristic algorithms are always used for solving combinatorial problems or hard optimization problems [17] since they can provide good solutions at reasonable computational cost [18, 19]. However, they may not be able to guarantee the optimality of solution due to their stochastic nature [20].

Slightly different from some meta-heuristic algorithms that improve a single solution, the genetic algorithm handles a group of feasible solutions simultaneously [21]. These feasible solutions are encoded into chromosomes [22] and placed into an environment analogue of natural evolution where they need to survive, adapt, and propagate their genetics to the future generations [21]. The evolution of these chromosomes is the process of searching the optimum solution. The evolution takes many generations to converge to perfectly adapted chromosome (global optimum solution) [21]. Hence, timing for ending the evolution/searching process is crucial. Otherwise, premature convergence could happen in the evolution.

A premature convergence should be avoided as it may led to the acquisition of a local optimum solution instead of global optimum solution [20]. The choice of the stopping criterion would determine if the global optimum solution could be found before its searching process is terminated.

In this paper, we investigated the stopping criterion of genetic algorithm as the time of searching the solution space is one of the very important factors to find the global optimum solution. We proposed the density or saturation of population fitness as the new stopping criterion which served as a measurement key to end the searching process. The proposed stopping criterion was compared to a conventional stopping criterion which the searching process will be stopped when there is no improvement of fittest chromosome for some successive generations. The numerical results show that the proposed stopping criterion has better performance as compared to the conventional stopping criterion.

This paper is organized as follow. Section 2 gives the description of the genetic algorithm including solution encoding / encrypting, evolution and the stopping criteria that has been used. Section 3 details hybrid algorithms for two tested models, one with conventional stopping criterion, one with the proposed stopping criterion. Section 4 reports on experimental results with different parameters/genetic drift setting. In section 5, the findings are concluded and some recommendations are given for future study.

2. THE VITAE OF GENETIC ALGORITHM

In an artificial evolution system as shown in Figure 1, genetic algorithm search begins by generating a population of randomly generated candidate solutions. These candidate solutions are encoded into chromosomes [7] and brought into the evolution that is constructed by genetics operators. Each of the chromosomes is assigned with a fitness function [20] that serves as a fitness index.

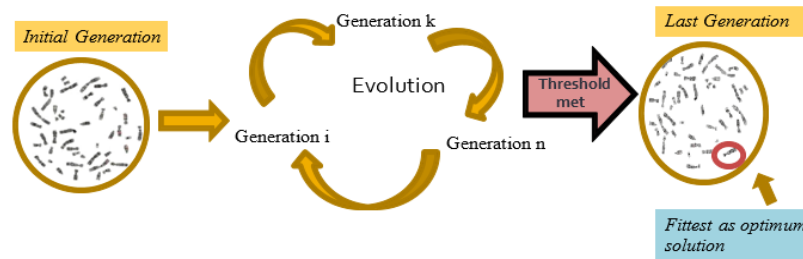


Figure 1. Artificial evolution system in genetic algorithm

The chromosomes from the same generation would have to compete with one another. A selection pressure that is biased to privilege on the fittest chromosomes is enforced in the system [5]. This act is to ensure that only those with dominant traits of optimisation would have higher possibilities to be selected for passing the genes [21]. Through the execution of genetic operators on these selected chromosomes, a new population / generation of chromosomes is formed [20]. The evolution/searching process will continue until a stopping criterion / threshold is met and the fittest in the last generation will be identified as an optimum solution [23]. However, the identified optimum solution could be a local optimum solution instead of a global optimum solution if the searching process ended too early.

Basically, the algorithm search involves two different spaces, one is coding space, and the other is solution space [24]. The implementation of genetic operators on coded solutions, namely chromosomes, works in the coding space. Nonetheless, the evaluation and selection of chromosomes are employed in the solution space which is the space for an actual solution [24].

2.1. Solution encryption

Each generation is constituted by a population of size M candidate solutions. These candidate solutions are feasible solutions [25] which will be encrypted into chromosomes C_{ij} constructed by gene g_{ijk} . They could be in the form of binary string [17], real number string [8] or matrix [26] depending on the types of optimization problem. The (1) is the solution representation form for j th chromosome of k genes in i th generation. For the purpose of this research, binary string constructed by gene $g_{ijk} = \{0,1\}$ was chosen.

$$C_{ij} = (g_{ijk}, \dots, g_{ij3}, g_{ij2}, g_{ij1}, g_{ij0}) \quad (1)$$

2.2. Selection, mating and mutation phases in the evolution

In the artificial evolution system, all chromosomes of each generation must go through the evolution [27] that comprised of a selection, mating and mutation phases as shown in Figure 2. In the selection phase, the selection pressure is enforced in a probabilistic manner into each generation by using selection operators such as Roulette Wheel [12], Tournament [28], and Elitism [9]. In this paper, Roulette Wheel and Elitism Selection Operators were selected to be imposed in the study.

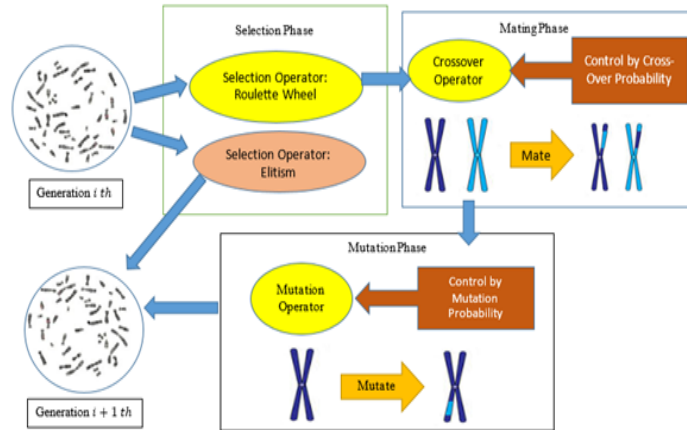


Figure 2. Evolution

As the name of Roulette Wheel suggests, each chromosome was given a slot that is proportional to its fitness in an imaginary roulette wheel. The size of each slot determines the probability of a chromosome being selected for the next phase. This means that the fittest would have precedence in breeding than those that were not well-adapted [17]. The selection pressure of Roulette Wheel operator was enforced into the chromosomes pool with some probability of crossover (crossover rate, CR).

Two chromosomes that had been selected by Roulette Wheel were treated as the parent chromosomes (PC_1, PC_2) for breeding in the mating phase. In this phase, the crossover operator exchanged and spliced the segmentations of both parent chromosomes at random point (r) [25] to form new chromosomes called offsprings (O_1, O_2) (2). These offsprings will carry the exchanged genetic information which inherited from their parent chromosomes [7]. In this study, the operation was repeated until a population of $M-2$ offspring was formed.

$$\left. \begin{aligned}
 PC_1 &= (g_{ijk}, \dots, g_{ijr+1}, g_{ijr}, \dots, g_{ij2}, g_{ij1}, g_{ij0}) \\
 PC_2 &= (g'_{ijk}, \dots, g'_{ijr+1}, g'_{ijr}, \dots, g'_{ij2}, g'_{ij1}, g'_{ij0}) \\
 O_1 &= (g_{ijk}, \dots, g_{ijr+1}, g'_{ijr}, \dots, g'_{ij2}, g'_{ij1}, g'_{ij0}) \\
 O_2 &= (g'_{ijk}, \dots, g'_{ijr+1}, g_{ijr}, \dots, g_{ij2}, g_{ij1}, g_{ij0})
 \end{aligned} \right\} \quad (2)$$

Next, the offsprings were brought into the mutation phase that consisted of mutation operator with some probability of mutation (mutation rate, MR). The probability of mutation is always a lower probability than the probability of crossover [7]. The mutation operator might alter the gene of the chromosomes at random position (R) [9] (3) with the intention of varying the genetic [22] and hence further expanding the solution space search [20]. The offsprings would then be regarded as new chromosomes for the coming generation. The mutation operators might serve as a tool to reduce the risk of premature convergence.

$$\left. \begin{aligned}
 O &= (g_{ijk}, \dots, g_{ijR}, g_{ij2}, g_{ij1}, g_{ij0}) \\
 g_{ijR} &= \begin{cases} 1 & \text{if } g_{ijR}=0 \\ 0 & \text{if } g_{ijR}=1 \end{cases}
 \end{aligned} \right\} \quad (3)$$

Simultaneously, Elitism operator was used to make sure that the elite chromosomes (fittest chromosomes of each generation) would not be disrupted by the execution of the crossover and mutation operators [28]. Two elites would be replicated directly as new chromosomes for the coming generation. This was a preservation act for reducing the probability of losing the fittest gene from the chromosome pool and decreasing the time of convergence to an optimum solution.

2.3. Stopping criteria

Lastly, a judgement to stop the evolution is important. The decision of having an appropriate threshold/stopping criterion greatly affects the capability of the algorithm [26]. Generally, there are two stopping criteria that have been widely used:

- a) Maximum generations or maximum CPU time allowed [26].
- b) No improvement of fittest chromosome for successive generations (*fittest chromosome repetition*) [14].

The first criterion (a) would end the evolution if the algorithm had met the pre-defined maximum number of generations or CPU time. However, to determine the appropriate time or maximum number of generations is a perplexing puzzle. Defining a very huge number of maximum generations or CPU time would lead to unpractical computational time while a small number might cause the algorithm has not enough time to reach a global optimum solution. The size of the population would sometimes influence the duration required for convergence [28].

In this aspect, the second criterion (b) might seem a better choice since the algorithm would stop if there is no improvement of fittest chromosome for few generations successively (*fittest chromosome repetition*). Again, it is another paradox where the algorithm user needs to determine the appropriate number of successive repetition for dismissing the search. The successive repetition of fittest chromosome is greatly influenced by the size and complexity of the research problem [26].

In this paper, a stopping criterion that measures the saturation of population fitness (F) of M chromosomes was proposed. The proposed stopping criterion aimed to work as a threshold to stop searching when the fitness deviation of the population is small (4) (when $\delta \rightarrow 0$) and further enhance the competency of algorithm in finding global optimum solution.

$$[(1/M) \sum (F_{ij}-F)^2] < \delta, \delta \rightarrow 0 \tag{4}$$

3. EXPERIMENTAL MODEL ALGORITHM

To test the proposed stopping criterion, two models with different stopping criteria were developed. The first model, named as Normal stopping criterion (Nsc) model, was designed with an algorithm that ends the searching process when *fittest chromosome repetitions* have reached the plateau, prefix upper bound. The second model, named as the Saturation stopping criterion (Ssc) model, was created with the proposed stopping criterion. The searching was stopped when the generation saturated with the fittest chromosomes. The algorithms of the two models are illustrated in Figure 3.

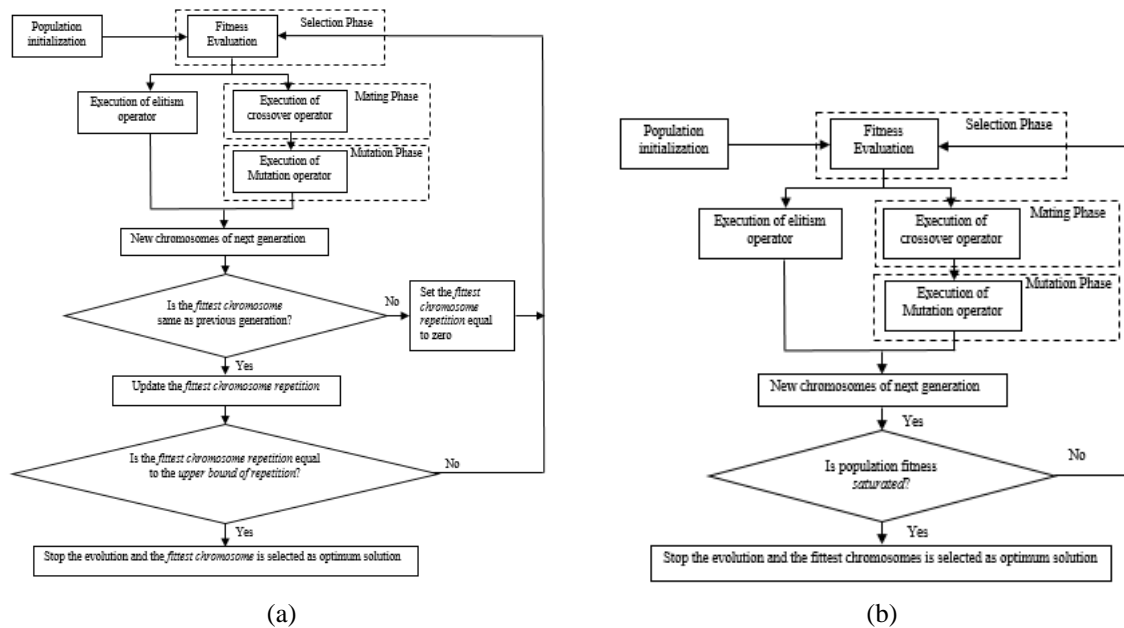


Figure 3. Algorithms for (a) Nsc Model; (b) Ssc Model

These two models are hybrid with a forecasting equation. This forecasting equation consists of one adjusted parameter α that range between zero and 1. The choice of the parameter α would influence the accuracy of forecasting simulation. Hence, the two models were assigned to identify a global optimum solution α such that the accurate forecasting simulation could be produced. Numerical tests were used to test the efficiency of the two models.

The parameter α would be encoded into chromosome in the form of (1) and the chromosome decipher of α is as (5). Each encoded parameter α_{ij} was assigned with a fitness function ($F_{ij}(\alpha_{ij})$), Mean Absolute Error (6). The $F_{ij}(\alpha_{ij})$ expressed the fitness function of j th chromosomes in i th generation. The chromosome that is able to minimize the errors of forecast was viewed as fittest. The forecasting equation was modified and hybrid into algorithms stated as (7). The h_{ijt} was the forecast data simulated by (7) and y_t represented the real data.

$$\alpha_{ij} = (\sum_{k=0}^K g_{ijk} \times 2^k) / 100, \quad K=0,1,2,\dots,k \tag{5}$$

$$\text{Minimize } F_{ij}(\alpha_{ij}) = (1/k) \sum |h_{ijt} - y_t| \quad \text{subject to } 0 < \alpha_{ij} < 1 \tag{6}$$

$$\left. \begin{aligned} h_{ijt} &= \alpha_{ij} (f_{ij1t} - f_{ij2t}) / (1 - \alpha_{ij}) + [2 f_{ij1t} - f_{ij2t}] \\ f_{ij1t} &= \alpha_{ij} y_t + (1 - \alpha_{ij}) f_{ij1t-1} \\ f_{ij2t} &= \alpha_{ij} f_{ij1t} + (1 - \alpha_{ij}) f_{ij2t-1} \end{aligned} \right\} \tag{7}$$

Generation, $i = 1, 2, \dots$ upper bound of generation

Chromosome, $j = 1, 2, \dots, M$ chromosome

Data Time, $t = 1, 2, \dots$ maximum number of time

In the selection phase, the selection pressure was affected by selection probability ($RP_{ij}(C_{ij})$) (8) of each chromosome. At the same time, two elite chromosomes (EC_w) would be replicated directly into the next generation (9). Two elites (EC_w) of generation i th were assigned as the first two new chromosomes ($C_{i+1,w}$) of the coming generation $i+1$ th. In the crossover and mutation phases, the possibilities of happening were controlled by crossover rate (CR) and mutation rate (MR).

$$\left. \begin{aligned} P_{ij}(C_{ij}) &= F_{ij} / \sum F_{ij} \\ CP_{ij}(C_{ij}) &= 1 - P(C_{ij}) \\ RP_{ij}(C_{ij}) &= CP_{ij} / \sum CP_{ij} \end{aligned} \right\} \tag{8}$$

$$\text{If } EC_w = C_{i,j}, \text{ then } C_{i+1,w} = EC_w, w = 1, 2, j = \text{two identified elite chromosomes} \tag{9}$$

4. RESULTS AND DISCUSSION

Both Nsc and Ssc models were tested for their capabilities in forecasting simulation by using a set of time series data range [1300, 1600]. The competence of models in avoiding premature convergence and finding the global optimum solution was used as an index for gauging a good model. A good model shall be able to produce a simulated result that is very close to actual value [28] under various circumstances (of different crossover rate and mutation rate).

In this research, both models carried out simulation experiment with 1000 trials to accumulate the statistical records of stopping criteria efficiency. The effectiveness of stopping criterion could be revealed by the probability of having a good forecast (consist of global optimum solution α) when the searching ended. The genetic drift/parameters of both experiment models are standardized as in Table 1.

Table 1. Genetic drift/parameters for experiment models

| Genetic drift/parameters |
|---|
| Population size, M = 20 chromosomes |
| Crossover rate, CR = [0.5, 0.9] with rate interval 0.1 |
| Mutation rate, MR = [0.01, 0.1] with rate interval 0.01 |

Figure 4 depicts the probabilities of Nsc model of achieving global optimum alpha α under different crossover rates (CR) and mutation rates (MR). The performances of Nsc model were above average when crossover rate was set in the range of [0.5, 0.7], the probabilities of getting good forecast were reported above 0.6 but far from 1. Under these CRs, the abilities of the model were improved when the MR was raised to the

range [0.08, 0.1], yet the probabilities were still slightly further from 1. The effectiveness of Nsc in obtaining accurate forecasting results was reported good when CR was set high (CR=[0.8,0.9]), the probabilities were averagely above 0.7 regardless of the MR.

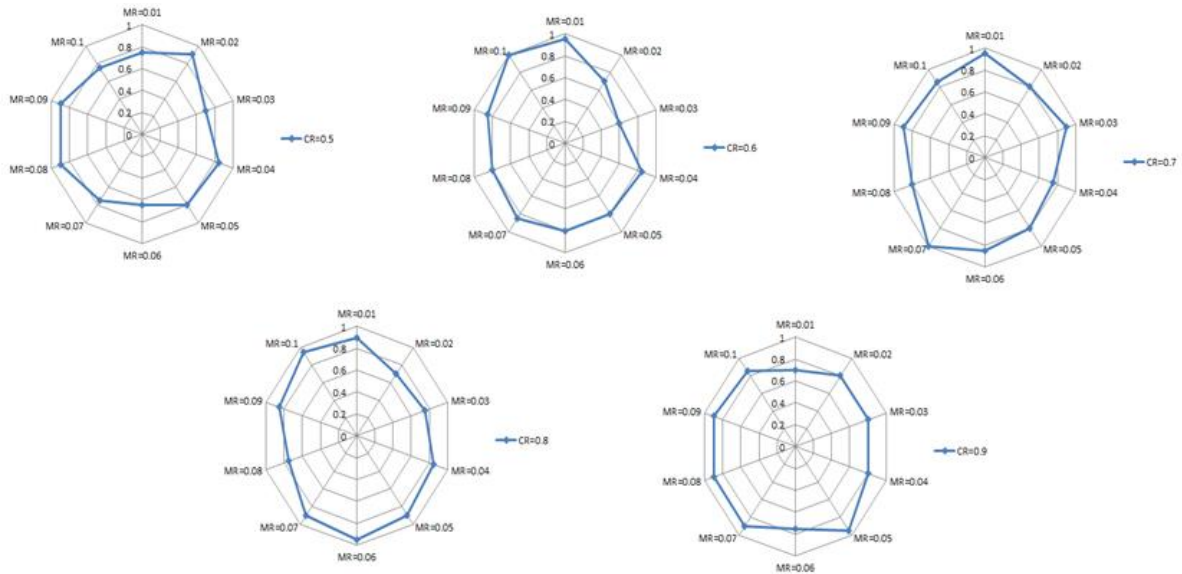


Figure 4. The probabilities of Nsc model of obtaining global optimum α under CR=[0.5,0.9] with interval rate 0.1 and MR=[0.01,0.1] with interval rate 0.01

The radar charts of Figure 5 reveal the capability of the Ssc model under the same genetic drift. The Ssc model seemed able to provide better results than the Nsc model when CR and MR rates were not high (CR=[0.5,0.7], MR=[0.01,0.06]). Under the same CR interval [0.5, 0.7], the competency of the model became apparent when MR was set above 0.07. The performances of the Ssc model were superior when both CR and MR were high (CR=[0.8, 0.9] and MR=[0.07, 0.1]), the probabilities of getting global optimum solution were approaching 1.

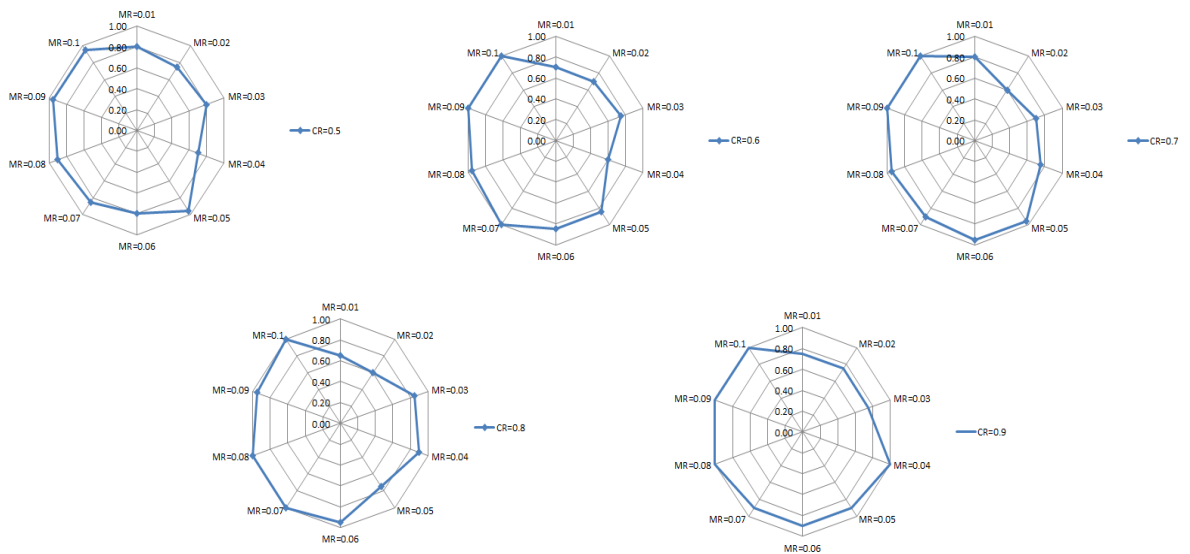


Figure 5. The probabilities of Ssc model of obtaining global optimum α under CR=[0.5,0.9] with interval rate 0.1 and MR=[0.01,0.1] with interval rate 0.01

Figure 6 show the comparison of Ssc and Nsc models performances under (a) $CR=[0.5, 0.9]$ with interval rate 0.1 and (b) under $MR=[0.01, 0.1]$ with interval rate 0.01. From both the graphical representations, the Ssc model has demonstrated tracking ability better than the Nsc model. The probabilities of the Ssc model in generating good simulation model were close to 1 especially when CRs were high as shown in Figure 6(a). In Figure 6(b), the Nsc model was better at the low mutation rate ($MR=[0.01, 0.02]$) but the prevailing characteristic of the Ssc model emerged when MR was greater than 0.03. The possibilities of the Ssc model having a good forecast were very close to 1 when MR was above 0.07.

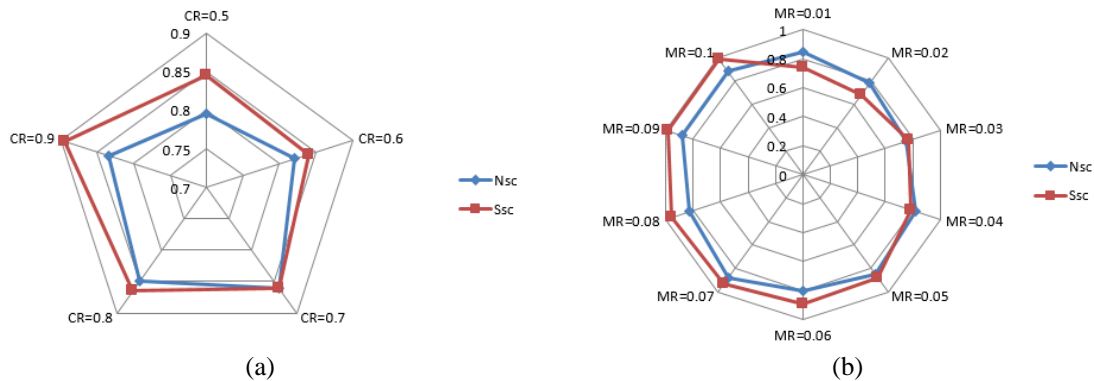


Figure 6. Comparison of Ssc and Nsc models performances (probabilities)
(a) under $CR=[0.5,0.9]$ with interval rate 0.1; (b) under $MR=[0.01,0.1]$ with interval rate 0.01

5. CONCLUSION AND RECOMMENDATION

Two research models, Nsc and Ssc models, with different approaches in stopping criteria that had been developed and were numerically tested with forecasting simulation. The Nsc model adopted the idea of setting the upper bound for the fittest chromosome repetition. If a chromosome was declared as the fittest of its generation and as being successively selected for M generations, then the searching process would be terminated and this chromosome was viewed as the best solution.

Nonetheless, the Ssc model was designed to apply the concept of fittest dominancy. It is a known fact that the fittest of each generation would have a higher probability in transmitting the genetic information. Gradually, the population would be dominated by the fittest's trait. If the density of saturation or degree of domination of the fittest had reached the pre-defined boundary (4), then the evolution was considered complete and the fittest of the last generation was commended as the optimum solution.

In the numerical test, it was found that the Ssc model was more capable, the probabilities of obtaining global optimum parameter α were found higher than Nsc model. The Nsc model showed a moderate performance with low values of MR and CR. Its competency was slightly improved after the MR and CR were raised. Generally, the Ssc model demonstrated a better skill in finding a good solution when MR and CR were not high. The capability of the Ssc model boosted when CR and MR had been set high. Hence, it can be concluded that the proposed stopping criterion has shown a great improvement in enhancing the algorithm ability in solving the optimization problem and reducing the risk of premature convergence. For future study, one may research on the influence of different genetic operators in evolution. The proposed stopping criterion may also be tested in other fields of optimization such as engineering.

REFERENCES

- [1] H. M. Pandey, et al., "A comparative review of approaches to prevent premature convergence in GA," *Applied Soft Computing*, vol. 24, pp. 1047-1077, Nov 2014.
- [2] J. Zhang, et al., "Optimization of Actuators in Smart Truss Based on Genetic Algorithms," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol/issue: 10(7), pp. 1615-1620, 2012.
- [3] M. Bolhasani and S. Azadi, "Parameter estimation of vehicle handling model using genetic algorithm," *Scientia Iranica*, vol/issue: 11(1&2), pp. 121-127, 2004.
- [4] K. Deb, "New Optimization Techniques in Engineering," in *Studies in Fuzziness and Soft Computing*, Springer, Berlin, Heidelberg, pp. 13-14, 2004.
- [5] J. Andre, et al., "An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization," *Advances in Engineering Software*, vol/issue: 32(1), pp. 49-60, 2001.

- [6] B. Benlahbib, et al., "Wind Farm Management using Artificial Intelligent Techniques," *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue: 7(3), pp. 1133, 2017.
- [7] A. F. J. Al-gburi, et al., "An heterogeneous population-based genetic algorithm for data clustering," *Indonesian Journal of Electrical Engineering and Informatics*, vol/issue: 5(3), pp. 275-284, 2017.
- [8] S. Jangjit and P. Laohachai, "Parameter Estimation of Three-Phase Induction Motor by Using Genetic Algorithm," *Journal of Electrical Engineering & Technology*, vol. 4, no. 3, pp. 360-364, 2009.
- [9] H. Saito and N. Tsunashima, "Superquadrics parameter estimation from shading image using genetic algorithm," in *Proceedings of IECON'94 - 20th Annual Conference of IEEE Industrial Electronics Control and Instrumentation*, vol. 2, pp. 978-983, 1994.
- [10] A. P. Unnikrishnan, et al., "License Plate Localization Using Genetic Algorithm including Color Feature Extraction," *Procedia Technology*, vol. 24, pp. 1445-1451, 2016.
- [11] F. Jolai, et al., "Genetic algorithm for bi-criteria single machine scheduling problem of minimizing maximum earliness and number of tardy jobs," *Applied Mathematics And Computation*, vol. 194, pp. 552-560, 2007.
- [12] S. H. R. Pasandideh, et al., "A genetic algorithm for vendor managed inventory control system of multi-product multi-constraint economic order quantity model," *Expert Systems with Applications*, vol/issue: 38(3), pp. 2708-2716, 2011.
- [13] R. Venkatesan and V. Kumar, "A genetic algorithms approach to growth phase forecasting of wireless subscribers," *International Journal of Forecasting*, vol/issue: 18(4), pp. 625-646, 2002.
- [14] B. C. Ervural, et al., "Model Estimation of ARMA Using Genetic Algorithms: A Case Study of Forecasting Natural Gas Consumption," *Procedia - Social and Behavioral Sciences*, vol. 235, pp. 537-545, Oct 2016.
- [15] C. Djellali and M. Adda, "A new predictive approach to variables selection through Genetic Algorithm and Fuzzy Adaptive Resonance Theory Using medical diagnosis as a case," *Procedia Computer Science*, vol. 109, pp. 448-457, 2017.
- [16] S. Kalantari and M. S. Abadeh, "GASA-JOSH: A Hybrid Evolutionary-Annealing Approach for Job-Shop Scheduling Problem," vol/issue: 2(2), pp. 132-140, 2013.
- [17] N. Dugan and Ş. Erkoç, "Genetic Algorithms in Application to the Geometry Optimization of Nanoparticles," *Algorithms*, vol/issue: 2(1), pp. 410-428, 2009.
- [18] C. R. R. and J. E. Beasley, "Introduction," in *Introduction in Modern Heuristic Techniques for Combinatorial Problems*, C. R. Reeves, Ed. New York: Halsted Press(an imprint of John Wiley & Sons, Inc), pp. 1-19, 1992.
- [19] N. Hassan, et al., "Number of iteration analysis for complex fss shape using GA for efficient ESG," *Bulletin of Electrical Engineering and Informatics*, vol/issue: 7(4), pp. 505-513, 2018.
- [20] V. N. Wijayaningrum and W. F. Mahmudy, "Fodder Composition Optimization using Modified Genetic Algorithm," *International Journal of Intelligent Engineering and Systems*, vol/issue: 9(3), pp. 1-8, 2016.
- [21] X. Yan, et al., "An Improved Genetic Algorithm and Its Application in Classification," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol/issue: 10(1), pp. 337-346, 2013.
- [22] S. Kalantari and M. S. Abadeh, "An Effective MultiPopulation Based Hybrid Genetic Algorithm for Job Shop Scheduling Problem," *Bulletin of Electrical Engineering and Informatics*, vol/issue: 2(1), pp. 59-64, 2013.
- [23] M. S. Hossain, et al., "An improved fitness function for automated cryptanalysis using genetic algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol/issue: 13(2), pp. 643, 2019.
- [24] G. Renner and A. Ekárt, "Genetic algorithms in computer aided design," *Computer-Aided Design*, vol/issue: 35(8), pp. 709-726, Jul 2003.
- [25] R. B. Bahaweres, et al., "Software Testing With Genetic Algorithm," no. September, pp. 19-21, 2017.
- [26] C. Oğuz and M. F. Ercan, "A Genetic Algorithm for Hybrid Flow-shop Scheduling with Multiprocessor Tasks," *Journal of Scheduling*, vol/issue: 8(4), pp. 323-351, Jul 2005.
- [27] F. F. Yeng, et al., "The Consequence of Elitism and Rabble Selection Operator in Genetic Algorithm," in *2014 IEEE Colloquium on Humanities, Science and Engineering*, pp. 469-474, 2014.
- [28] A. R. Melo, et al., "Blood Perfusion Parameter Estimation in Tumors by means of a Genetic Algorithm," *Procedia Computer Science*, vol. 108, pp. 1384-1393, 2017.