Evaluation of Fast Evolutionary Programming, Firefly Algorithm and Mutate-Cuckoo Search Algorithm In Single-Objective Optimization

Muhammad Zakyizzuddin Bin Rosselan, Shahril Irwan Bin Sulaiman, and Norhalida Binti Othman

***Abstract*— In this study proposes an evaluation of different computational intelligences, i.e Fast-Evolutionary Algorithm (FEP), Firefly Algorithm (FA) and Mutate-Cuckoo Search Algorithm (MCSA) for solving single-objective optimization problem. FEP and MCSA are based on the conventional Evolutionary Programming (EP) and Cuckoo Search Algorithm (CSA) with modifications and adjustment to boost up their search ability. In this paper, four different benchmark functions were used to compare the optimization performance of these three algorithms. The results showed that MCSA is better compare with FEP and FA in term of fitness value while FEP is fastest algorithm in term of computational time compare with other two algorithms.**

***Index Terms*-Fast-Evolutionary programming (FEP), Firefly algorithm (FA), Mutate-Cuckoo search algorithm (MCSA), Optimization, Test functions.**

I. INTRODUCTION

Recently, there are a lot of nature inspired meta-heuristic evolutionary algorithms have been developed for optimization problems. Most of these algorithms are nature inspired and work on the basis of random search in some suitable search region depending on the problem [1]. The main concerns of developing these meta-heuristic algorithms are to find the best solution for the problem in certain time. Even though, the solutions produce may be not the best for the required problem but still they stand valid since they do not require excessively long time to be solved [2]. Theoretically, meta-heuristic algorithms consist of two main characteristic which are intensification (exploitation) and diversification (exploration) [2][3].

This paper is about an evaluation of fast evolutionary programming, firefly algorithm and mutate-cuckoo search algorithm in single-objective optimization submitted on 21st September 2016. Accepted 15 November 2016.

Muhammad Zakyizzuddin Bin Rosselan finished his degree in Electrical Engineering at Universiti Teknologi MARA (UiTM), Faculty of Electrical Engineering, Universiti Teknologi MARA (UiTM) 40450, Shah Alam, Selangor. (email: zaky_rosselan@yahoo.com.my)

Shahril Irwan Bin Sulaiman finished his Ph.D. in Photovoltaiv (PV) energy system at Universiti Teknologi MARA (UiTM), Faculty of Electrical Engineering, Universiti Teknologi MARA (UiTM) 40450, Shah Alam, Selangor. (email: shahril_irwan2004@yahoo.com)

Norhalida Binti Othman finished her M.Eng.Sc in Power System Engineering at Universiti Teknologi Malaysia (UTM) Skudai, Faculty of Electrical Engineering *Universiti Teknologi MARA* 81750 Masai, Johor baharu,Malaysia (email : halida8142@johor.uitm.edu.my)

Intensification intends to select the best solution by searching around the current best solution, while diversification tries to improve the efficiency of exploring the search space so that the algorithm does not get stuck into local optimum [4].

Nowadays, there are a lot of these nature inspired algorithms developed by the previous researchers. For example, Ant Colony Optimization algorithm (ACO) was inspired form the behavior of ants in the wild, Particle Swarm Optimization algorithm (PSO) was evolved from the world of fish and bird, whereas the Bee Colony Optimization algorithm (BCO) was developed from the behavior of bees in their colony or habitat [5]. Most of these algorithms have been widely used for certain types of application such as optimization process. They are becoming useful as an alternative method to replace the existing conventional techniques in solving certain complicated problems in various areas.

This paper presents the evaluation of the Fast Evolutionary Programming (FEP), Firefly Algorithm (FA) and Mutate-Cuckoo Search Algorithm (MCSA) for single objective optimization (SOO).

These three algorithms will be evaluated using selected test functions which are De Jong's function, Rosenbrock's function, Axis parallel hyper-ellipsoid function and Rotated hyper-ellipsoid function [6]. Each algorithm used in this paper differs in the kind of their behavior and mutation technique that is used.

The rest of this paper first discusses the methodology of the project in Section 2, and then describes the findings and evaluation in in Section 3, while the conclusion of the project in Section 4.

II. THEORETICAL BACKGROUND

A. Fast-Evolutionary Programming (FEP)

The basic concept of the Fast-Evolutionary Programming (FEP) is based on the Conventional Evolutionary Programming (CEP). CEP was inspired from the biological evolution and has been proposed as an approach to artificial intelligence (AI). In CEP, there are two basic procedure involve which are each individual generates an offspring via mutation and the selection of the parents for the next generation are based on the better individuals from the parent and offspring populations (Pap. 40). Practically, CEP implemented the Gaussian distribution operator during mutation process. Meanwhile, FEP or also known as Cauchy mutation-based EP used the Cauchy distribution as a mutation operator in order to produce the offspring during the mutation process.

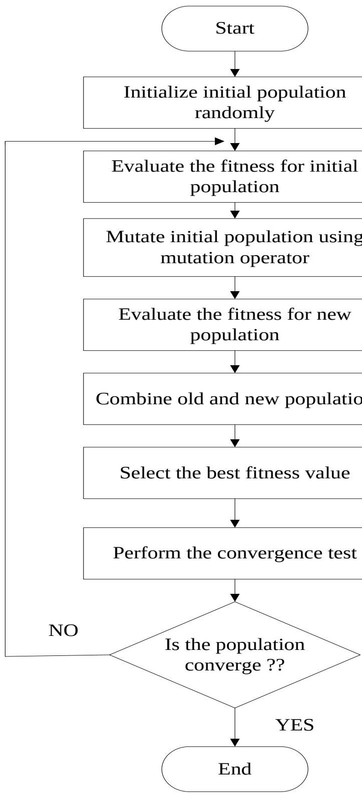


Fig 1. Stages of Fast-Evolutionary Programming (FEP).

Fig. 1 shows the flowchart of general procedure for FEP. There are five main components of this FEP which are initialization, fitness computation, mutation, combination and selection. First, during the initialization process, the initial population will be generated and this initial population called as the parent. Second, from the initial population generated, the fitness of this initial population will be evaluated using the selected function or equation. After that, the Cauchy distribution operator will be used to mutate the initial population and produce the new set of population that will be called as an offspring. Next, the combination process required the combination of the parent and the offspring in order to determine the better result that produce form the selection process.

The random number consist of zero mean and standard deviation to each vector of parent will be add to the parent (y_i) in order to produce a single offspring (y'_i). Equation (1) shows the general equation for Gaussian operator. For $N(0,\sigma^2_i)$, it represents a Gaussian random variable with mean 0 and standard deviation σi.

$$y'_i = y_i + N(0, \sigma^2_i), \text{ for } i = 1, 2.........n \quad (1)$$

Beside, for FEP using Cauchy operator, an offspring is created by the equation (2) shown below. Ci (0, 1) is define as

a Cauchy random variable with scale parameter k=1 centered at 0 that generate anew for each value of i.

$$y'_i = y_i + \sigma_i . C_i(0,1), \text{ for } i = 1,2.........n \quad (2)$$

B. *Firefly Algorithm (FA)*

Firefly algorithm (FA) has been invented by Xin-She Yang at Cambridge University during 2007 [5]. FA has been developed based on the social behavior of fireflies in the tropical summer sky [3][7]. Fireflies used the bioluminescence with different flashing pattern to communicate, search for pray and find mates. FA was developed based on three idealized rules which are firstly, all the fireflies will be attracted to other fireflies based on their sex and gender because all fireflies are unisex. Secondly, the less bright fireflies will move towards the brighter one because the attractiveness is proportional to the brightness. Lastly, analytical form or landscape of the objective function will affect the brightness of a firefly [8]. Fig. 2 shows the flowchart for general procedure of Firefly algorithm (FA).

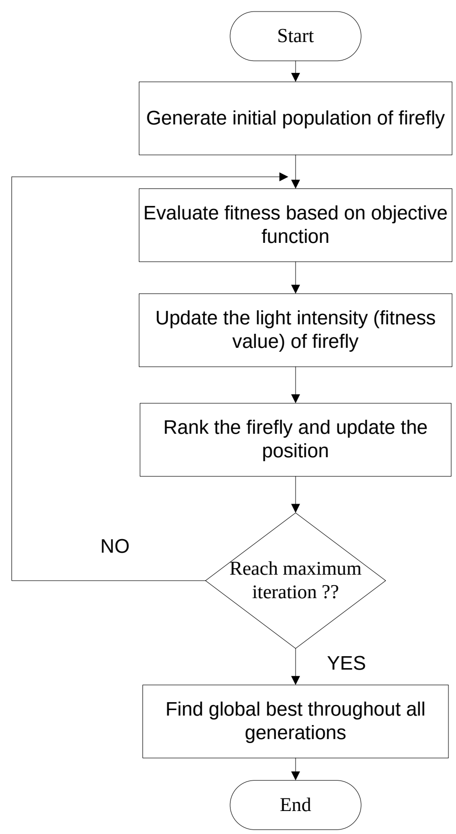


Fig 2. Stages of Firefly Algorithm (FA).

Based on Fig. 2, at the beginning of the FA process, the initial population will be produce and will be evaluate by using selected function or equation. Next, the light intensity I(r) of the fireflies will be updated. Basically, the light intensity depends on the inverse square law. However, in this research the inverse square law and absorption coefficient (γ) will be used simultaneously in order to avoid the singularity at certain part of the expression. After that, all the fireflies will be rank according to their objective whether for maximization or minimization process.

Last but not least, since the maximum iteration was reached, find the global best through all the generations. The reference for developing the new population which was selected from the best generations will be used to represent the best population produces from the previous generations.

C. *Mutate-Cuckoo Search Algorithm (MCSA)*

Cuckoo Search Algorithm (CSA) is one of the nature inspired optimization algorithm developed by Xin-She Yang in 2009 based on the behavior of the bird species that called cuckoo [4]. The CSA was developed by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of host bird [9]. This process involves two forms of cuckoo which are mature cuckoos and eggs. Mature cuckoos will lays their eggs in certain bird's nest and only the eggs that does not recognized by the host bird can survive meanwhile for the recognize eggs will be killed by the host bird.

Typically, CSA used Levy flight as the mutation operator. Levy flight can be thought of as a random walk where the step size has a Levy tailed probability distribution [4]. However, for Mutate-Cuckoo Search Algorithm (MCSA), Levy flight will be combine with the Cauchy distribution operator to perform the mutation process. The general procedure of MCSA shows in Fig. 3.

The process of MCSA has some additional process that makes it different from the conventional Cuckoo Search Algorithm (CSA) which is the addition of the Cauchy distribution operator during the mutation process. The rest of the processes are similar with the conventional CSA which is starting with the defining and initializing the parameters and the initial host nest population. Next, evaluate the fitness function and modify the host nest population by using Levy flight. After that, Cauchy distribution operator will do the mutation process. Then, evaluate again the new population by using the selected fitness function. Next, if the condition does not satisfy, move the cuckoo toward the best environment and repeat the process starting from the beginning. If the condition satisfies, choose current best and if the populations exceed maximum generation, the process will stop.

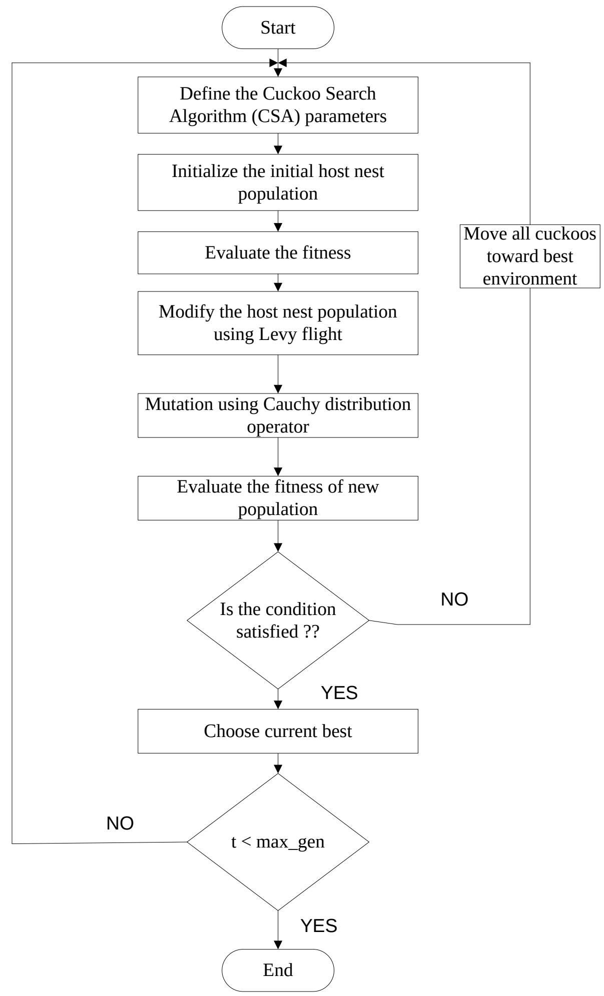


Fig 3. Stages of Mutate-Cuckoo Search Algorithm (MCSA).

III. METHODOLOGY

A. *Test Functions*

There are many benchmark test functions which are designed to test the performance of optimization algorithms [2]. In this paper, four test functions which are De Jong's function, Rosenbrock's function, axis parallel hyper-ellipsoid function and rotated hyper ellipsoid function have been used to determine the relative performance of the three different optimization techniques. The simplest test function is De Jong's and also known as sphere function and the function can be expressed using equation (3).

$$f_{DeJong's} = \sum_{i=1}^{n} {x_i}^2 \tag{3}$$

The second test function that will be used was axis parallel hyper-ellipsoid function and also known as a weighted sphere model. The general definition of the function can be expressed using equation (4).

$$f_{axisparallel-ellipsoid} = \sum_{i=1}^{n} (i.x_i) \tag{4}$$

The next test function is Schwefel's function. Another name of this function is rotated hyper-ellipsoid function and it is actually an extension of axis parallel hyper-ellipsoid function. The function has the following general expression as shown in (5).

$$f_{schwefel's} = \sum_{i=1}^{n} \sum_{j=1}^{n} {x_j}^2 \quad (5)$$

The last test function is Rosenbrock's valley function and also called as the banana function. The function has the following general expression as shown in (6).

$$f_{Rosenbrock's} = \sum_{i=1}^{n-1} [100(x_{i+1} - {x_i}^2)^2 + (1 - x_i)^2] \quad (6)$$

The performance of all the optimization algorithms used will be tested using this test functions. The performance of the algorithms will be measured based on the fitness solution and computation time.

IV. RESULTS AND DISCUSSION

The purpose of developing the Fast-Evolutionary Programming (FEP), Firefly Algorithm (FA) and Mutate-Cuckoo Search Algorithm (MCSA) are to make the comparison between the performance of these three algorithms based on their optimization accuracy and the computational time. The ranges between 20 to 100 populations were used in this paper and the populations were generated by using random numbers. The performance of each algorithm was showed in Fig. 4, 5 and 6 which have been tested using selected test function.

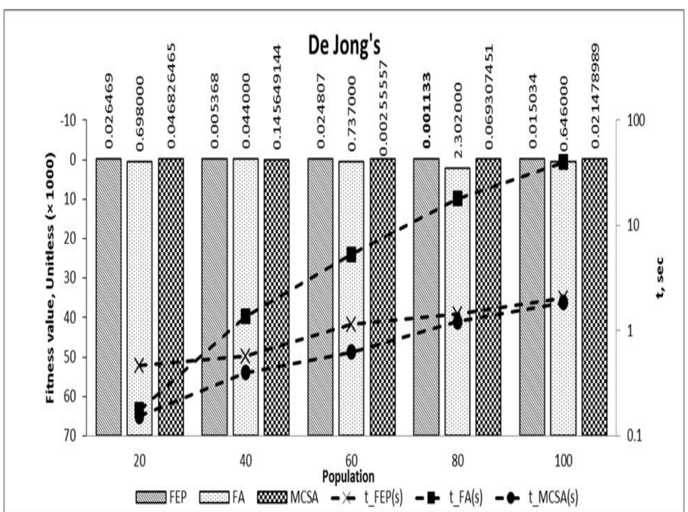


Fig 4. Comparison between FEP, FA and MCSA using De Jong's function

In Fig. 4, De Jong's function was used as a benchmark function to compare the performance between FEP, FA and MCSA. Based on the results, FEP during 80 numbers of populations gives the lowest fitness which is 0.001133. As showed in Fig. 4, MCSA gave the lowest computational time and FA gave the worst computational time.

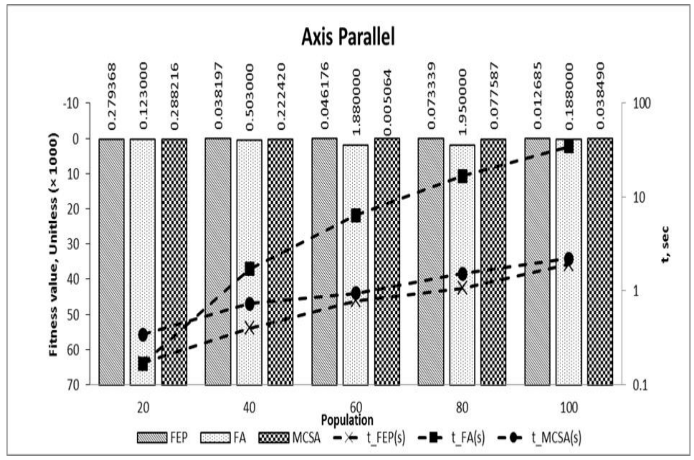


Fig 5. Comparison between FEP, FA and MCSA using Axis-parallel hyper-ellipsoid function.

From Fig. 5, MCSA during 60 populations shows the lowest fitness value which is 0.005064. Besides, by using this benchmark function, FEP gave the lowest computational time compare with FA and MCSA.

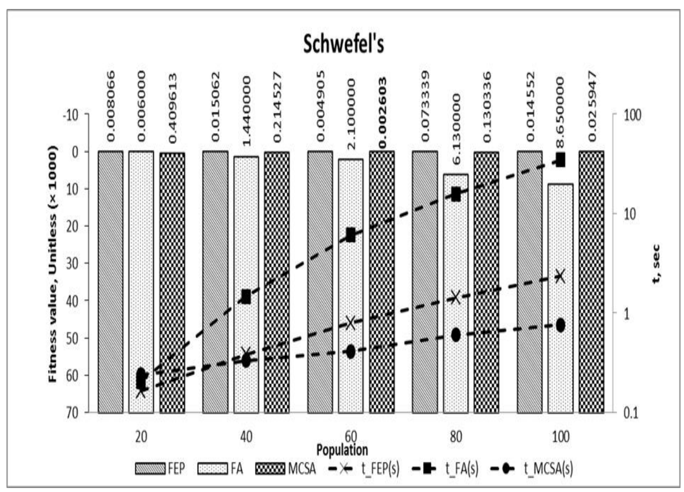


Fig 6. Comparison between FEP, FA and MCSA using Rotated hyper-ellipsoid function.

Rotated hyper-ellipsoid or Schwefel's function recorded the lowest fitness value is at 60 numbers of populations which is 0.002603. Besides, MCSA scores the lowest computational time compare to FA and FEP. Next, the increment of the duration for optimization process is directly proportional with the increment of the number of populations.

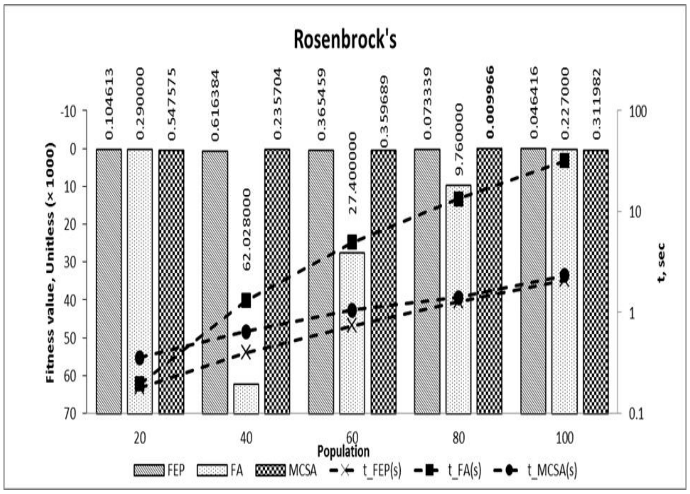


Fig 7. Comparison between FEP, FA and MCSA using Rosenbrock's function.

Result in Fig. 7 shows during 80 populations, MCSA gave the lowest fitness value which is 0.009966. Meanwhile, for computational time, FEP presents the lowest computational time as compare with FA and MCSA.

As a conclusion, in terms of fitness value, it can be said that MCSA gave the best and lowest computational time compare with FA and FEP since 3 from 4 of the benchmark function showed the lowest fitness value at MCSA. Meanwhile, for computational time, MCSA and FEP gave the lowest computational time as compared with FA.

V. CONCLUSION

In this paper, the performance of Fast-Evolutionary Programming (FEP), Firefly Algorithm (FA) and Mutate-Cuckoo Search Algorithm (MCSA) have been compared using selected test functions. 5 different numbers of populations used in this study starting with 20, 40, 60, 80 and 100. Results showed that MCSA provides the best optimal solution with better computational time when compared with FEP and FA in solving optimization problem described in De Jong's, Axis-parallel ellipsoid, Schwefel's and Rosenbrock's functions.

ACKNOWLEDGMENT

This work was supported in part by the Research Acculturation Grant Scheme, Ministry of Education (Ref: 600-RMI/RAGS 5/3 (6/2014)) and Universiti Teknologi Mara (UiTM) Malaysia.

REFERENCES

[1] H. Soneji and R. C. Sanghvi, "Towards the improvement of Cuckoo search algorithm," *Proc. 2012 World Congr. Inf. Commun. Technol. WICT 2012*, vol. 6, pp. 77–88, 2012.

[2] G. P. Singh and A. Singh, "Comparative Study of Krill Herd, Firefly and Cuckoo Search Algorithms for Unimodal and Multimodal Optimization," *Int. J. Intell. Syst. Appl.*, vol. 6, no. 3, pp. 35–49, 2014.

[3] A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, "Firefly algorithm with chaos," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 1, pp. 89–98, 2013.

[4] S. Roy and S. Sinha Chaudhuri, "Cuckoo Search Algorithm using Lèvy Flight: A Review," *Int. J. Mod. Educ. Comput. Sci.*, vol. 5, no. 12, pp. 10–15, 2013.

[5] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Second Edi. United Kingdom, 2010.

[6] N. I. A. Aziz, S. I. Sulaiman, I. Musirin, and S. Shaari, "Assessment of evolutionary programming models for single-objective optimization," in *Proceedings of the 2013 IEEE 7th International Power Engineering and Optimization Conference, PEOCO 2013*, 2013, pp. 304–308.

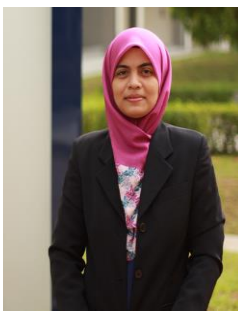
[7] A. Hossein, X. Yang, and A. Hossein, "Mixed variable structural optimization using Firefly Algorithm," *Comput. Struct.*, vol. 89, no. 23–24, pp. 2325–2336, 2011.

[8] X.-S. Yang, "Firefly Algorithms for Multimodal Optimization," pp. 1–10, 2010.

[9] E. Valian, S. Tavakoli, S. Mohanna, and A. Haghi, "Improved cuckoo search for reliability optimization problems," *Comput. Ind. Eng.*, vol. 64, no. 1, pp. 459–468, 2013.

Muhammad Zakyizzuddin Bin Rosselan graduated with a B.Eng in Electrical Engineering majoring in Power System (Universiti Teknologi Mara (UiTM), Malaysia). Currently, he is a Research Assistant and a master student at Faculty of Electrical Engineering (Universiti Teknologi Mara (UiTM), Malaysia). He is taking Msc.Eng.Sc in PV Engineering and his research concentrate in sizing and optimization of Grid-connected Photovoltaic (GCPV) system.

Shahril Irwan bin Sulaiman graduated with Ph.D. in photovoltaiv (PV) energy system UiTM, Malaysia), M.Eng.Sc. in PV Engineering (UNSW, Australia), B.Eng. in Electrical & Electronics Engineering (UNITEN, Malaysia), and Cert. PV system (PTM-ISPQ, Malaysia). He has many years technical experience in PV energy systems with various government agencies and industry. He is Master Trainer for competency training in design and installation of grid-connection and stand-alone PV system courses accredited by national and international organizations.He is also an Alternate Member of the Working Group on PV for development of Malaysian Standards. Currently he teaches at Universiti Teknologi Mara, Malaysia, as a Senior Lecturer.



Norhalida Binti Othman graduate with B.Eng (Hons) degree from Universiti Teknologi Mara (UiTM), Shah Alam, Malaysia and M.Eng.Sc in Power System Engineering at Universiti Teknologi Malaysia (UTM) Skudai. Her research interest is in optimization technique in Power system. She currently a lecturer in Universiti Teknologi Mara (UiTM) Pasir Gudang.