

Partial Evolutionary ANN for Output Prediction of a Grid-Connected Photovoltaic System

S.I Sulaiman, T.K Abdul Rahman, and I. Musirin

Abstract—This paper presents the Evolutionary Programming (EP) based technique to optimize the architecture and training parameters of a one-hidden layer backpropagation Artificial Neural Network (ANN) model for the prediction of total AC power output from a grid connected photovoltaic system. A partial Evolutionary Programming-ANN (EPANN) model has been developed for the prediction. It utilizes solar radiation, wind speed and ambient temperature as its inputs while the output is the total AC power produced from the grid connected PV system. EP is used to optimize the regression performance of the ANN model by determining the optimum values for the number of nodes in the hidden layer as well as the optimal momentum rate and learning rate for training. The results obtained from the EPANN have been compared with the results from a classical ANN with similar input and output settings. It is observed that the prediction of total AC power output from a grid connected PV system could be accelerated and simplified using the partial evolutionary ANN model.

Index Terms—Artificial neural network (ANN), Correlation coefficient (R), Evolutionary programming-ANN (EPANN), and Photovoltaic (PV).

I. INTRODUCTION

In the area of photovoltaic (PV) system, the major concern among the users is the unpredictability of the energy output from the system. Thus, many researchers have conducted different type of studies to overcome this problem. One of the prediction techniques used is the artificial neural network (ANN). A three-layer feedforward ANN was used to predict the energy output of a grid connected PV system by knowing the solar radiation, module temperature and clearness index [1]. Similarly, the output of a PV module was also predicted using the same architecture but with different types of inputs and outputs [2]. The ANN utilizes solar radiation, ambient temperature and module temperature as its inputs whereas voltage and current are used as outputs. Although these studies have produced many important discoveries in the prediction of PV system outputs using ANN, the design of ANN models using specific sets of design constraints relies so much on past experience with same applications and is subjected to trial and error processes [3]. For instance, in a

grid connected PV system, a study has been conducted to predict the total AC power output of a grid-PV system using manually-designed ANN [4]. However, this manual design of ANN is time consuming and vulnerable to inaccuracy issues due to the tedious trial and error process experienced by the ANN designers. Due to this limitation, evolution process has been introduced to provide faster training of the ANN [5].

Evolution of ANN can be achieved by evolving the connection weights, architectures or learning rules of the ANN [3]. The evolution of the connection weights is done by online training of the ANN connections using a preset architecture while the evolution of ANN architectures allows automatic adjustment of ANN topologies for different learning assignments. In the evolution of connection weights, weights are modified to learn a particular function. In contrast, the evolution of architectures describes the evolution of topological parameters such as number of nodes and number of hidden layers. On the other hand, the evolution of learning rules involves the adaptive process of searching the best learning rules [5].

In general, there are many types of evolution methods available for evolving ANN. One of the most popular methods is Evolutionary Programming (EP). This algorithm is an optimization tool based on natural evolution [6]. In this study, Evolutionary Programming (EP) is selected to perform the evolution of ANN. It was first introduced to simulate artificial intelligence through the evolution of finite state machines [7]. In addition, the usage of Gaussian mutation and self-adaptation has been proved to become a determining factor in improving the performance of EP [8].

Apart from that, due to the complexity in designing ANN models, simulated evolution has been introduced to automatically design the ANN [9]. For instance, a study to find the optimum ANN hidden layer size using EP has been successfully performed [10]. Likewise, an algorithm to determine the ANN hidden layer size and weight coefficients has been developed [11].

Therefore, besides predicting the total AC power output from grid-PV system, this paper also demonstrates the design of a hybrid prediction model using Evolutionary Programming-Artificial Neural Network (EPANN). Using this simulated evolution technique, the design of ANN models in this study is made faster compared to the conventional manual design. Nevertheless, the simulated evolution of ANN utilized in this work is considered as semi-automatic due to the fact that prior information on the

Manuscript received September 16, 2008.

S. I. Sulaiman is with the Universiti Teknologi MARA Malaysia, Shah Alam, 40450 Malaysia (phone: +603-55436031; fax: +603-55435077; e-mail: shahril@salam.uitm.edu.my).

T. K. Abdul Rahman, is with the Universiti Teknologi MARA Malaysia, Shah Alam, 40450 Malaysia (phone: +603-55435051; fax: +603-55435077; e-mail: khawa@salam.uitm.edu.my).

I. Musirin is with the Universiti Teknologi MARA Malaysia, Shah Alam, 40450 Malaysia (phone: +603-55435044; fax: +603-55435077; e-mail: ismailbm@salam.uitm.edu.my).

prediction of PV system output using ANN is used to simplify the search space during evolution process.

II. TESTED PREDICTION MODEL

In the last few decades, ANN has been used extensively in various fields of power engineering for solving many complex problems. In basic computational model of ANN, a node in an ANN collects input signals from other nodes and merges them. Relevant computation is performed before the result is mapped to an output node [12]. In this study, a multi-layer feedforward neural network has been employed for investigation. A three-layer feedforward ANN with one hidden layer has been used to satisfactorily predict the total AC power output from a grid connected PV system either through the classical model or the evolutionary model.

Data collection involves the collection of four data types including solar radiation, SR (in kW/m^2) falling on horizontal plane, wind speed, WS (in m/s), ambient temperature, AT (in $^\circ\text{C}$) and total AC power output (in kW). All data have been obtained from a 42kW_p grid connected PV system mounted on the roof of Quadrangle Building, University of New South Wales, Australia. The data patterns obtained are based on 15-minute interval. In this study, 1000 data patterns have been selected for the training process.

A three-variate ANN model that uses SR , WS and AT as its input and total AC power as its output has been employed. This work is valid as the performance of PV modules is also influenced by ambient temperature [13]. Apart from that, the wind speed would also influence the ambient temperature and hence affecting the output of the grid connected PV system. The proposed ANN model is illustrated in Fig. 1.

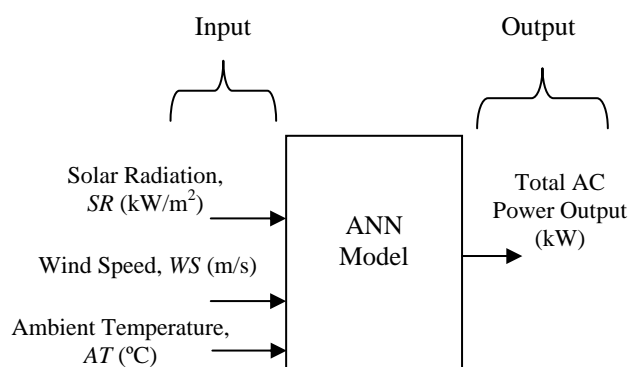


Fig. 1 ANN model for the prediction of power output from grid connected PV System

Before the EPANN model was developed, a classical ANN model has been developed to act as a benchmark for the EPANN model. Apart from that, both models are written using Matlab R2006a. The performance of the EPANN was then compared to the performance of classical ANN in evaluating the effectiveness of the EPANN model.

III. PROPOSED ANN MODEL

After determining the type, inputs and output of the ANN, suitable ANN parameters for evolution process has to be identified since there are a few methods to evolve the ANN. The first method involves the evolution of connection

weights during ANN training [14]-[15]. This method emulates the conventional technique of adjusting weights of ANN to learn an unknown function. The second method entails the evolution of ANN architecture [16]. Unlike the traditional method that requires the number of nodes of hidden layers to be determined heuristically by ANN designers, this more advanced method allows the size of nodes to expand or shrink automatically using simulated evolution. Besides that, the evolutionary training of ANN can also be achieved through the evolution of transfer functions [17].

Although there is an argument on the method that has the highest level of importance in ANN evolution process, the type of method to be used in ANN evolution actually depends on the amount and quality of information obtained about the proposed ANN prediction model. For instance, if more previous information on the learning algorithms is available compared to the information of ANN architecture, the evolution of learning algorithms should be implemented first before progressing towards the evolution of ANN architecture. Thus, the region of evolution is more restricted based on the prior information obtained. As a result, the convergence of the search becomes faster. Nevertheless, this preset rule of decision is invalid if there is a specific attention to the type of learning algorithm or the size of architecture to be investigated in a study.

In this work, several predetermined ANN characteristics and parameters have been selected before training is performed. Firstly, the Levenberg-Marquardt algorithm has been chosen as the learning algorithm for the ANN as it has been proved useful in many prediction studies. Secondly, the evolution of weights was not conducted as the back-propagation method could satisfactorily perform the optimal computation of the set of weights in ANN. Thirdly, the transfer function configuration is set to be [sigmoid, purely linear] after preliminary investigation. Apart from that, the number of epochs is set to be very large (1000 epochs) in order to allow accurate convergence of the ANN. The mean-square-error training goal is chosen to be sufficiently small (10^{-3}) to ensure satisfactory prediction accuracy. As these settings have been fixed at the beginning of training process, the evolutionary process is considered partial. Nevertheless, this partial EPANN is expected to simplify and speed up the design of ANN.

IV. PROPOSED EPANN MODEL

After the ANN parameters and characteristics have been identified, the evolution technique has been specified for the EPANN model. In this study, the EP is chosen to perform the evolution process. EP is a heuristic technique used to perform a semi-random search in optimizing an objective function. It comprises several main processes namely generation of random population, fitness evaluation, mutation, combination, selection and convergence test. In this study, the number of nodes in hidden layer, momentum rate, and learning rate, α of the ANN are allowed to evolve to reach their optimum values using EP. The best solution is achieved by maximizing the correlation coefficient, R of the prediction during ANN training. Higher R would imply a higher

accuracy of the prediction model. The flowchart of the proposed algorithm is illustrated in Fig. 2.

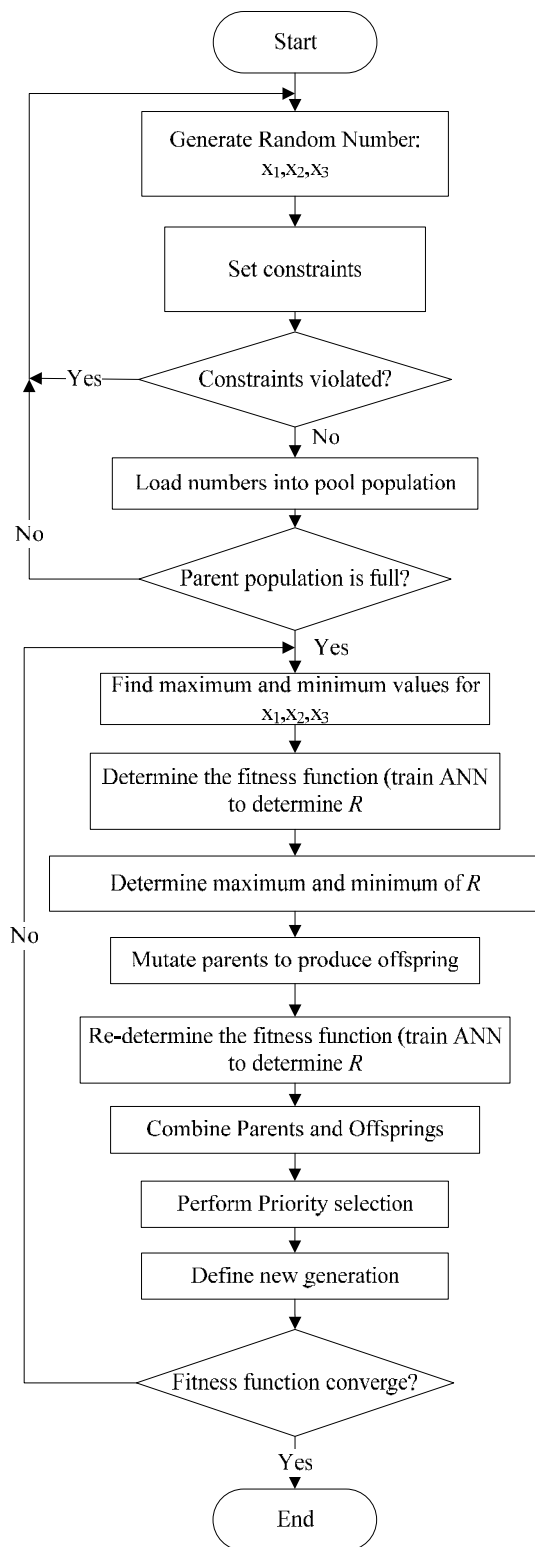


Fig. 2 Flowchart for the evolution of ANN using EP

The program begins with the initialization of three random numbers that represent the number of nodes in the hidden layer (x_1), learning rate (x_2) and momentum rate (x_3). These numbers are generated randomly based on normal distribution function. Although the computation time is not a major concern in this study, the search space of this program is confined by the results from previous efforts in predicting PV systems output using ANN. The randomly generated

values for x_1 are specified to be within a range from positive integer 1 to 100. The upper limit for these numbers is set to be very large (100) so that the problem of trapping at a local optima can be avoided. On the other hand, x_2 and x_3 are chosen to have random values between 0 and 1 as these numbers denote the learning rate and momentum rate respectively. In this case, the range is set to be as large as possible because insufficient prior information from previous studies is obtained regarding the suitable values for learning rate and momentum rate in predicting the output of a PV system. After a set of random numbers consisting x_1 , x_2 and x_3 have been generated, constraints of the search are determined. In this study, the constraints signify the range of correlation coefficient, R for the training process. In addition, R also signifies the fitness value of the evolutionary ANN. The R values for the constraints are set to be from 0.99 to 1.00 so that only good prediction results are transcribed into the population. Next, the preliminary fitness calculation is conducted to examine any constraint violation. This procedure is done by training the ANN using the set of random numbers generated previously. After the training process, if the set of random numbers generated produces R value outside the range defined by the predetermined constraints, the random number generation process is repeated. Otherwise, the set of random numbers are loaded into a pool population. This set of random numbers is called as the parent. This parent will undergo a mutation process to breed an offspring at a later stage. After a parent is successfully created, the random number generation is repeated until the pool population is completely filled. The maximum number of parents in the population is set to be 20 based on the previous work [18]. This value is repeatedly found to be sufficient for an EP to converge. Once the pool population is filled, the evolution process continues with the calculation of the minimum and maximum values for each variable obtained in the fully occupied pool population. These values will be then used for the mutation process in the later stage. Next, the first fitness function calculation is performed by training the ANN repeatedly using the 20 sets of random numbers obtained from the parent population. Thus, twenty R values are produced from the twenty sets of parents. Later, maximum and minimum values of R from these results are determined for mutation process. Then, the mutation process is started by mutating each parent using Gaussian mutation. This process generates 20 sets of random numbers and each set is called as an offspring. Using the sets of random numbers generated from each offspring, the second fitness function calculation is performed by training the ANN. Subsequently, both parents and offspring are combined to form a new population of 40 sets of random numbers with each of them having their own fitness value. After the combination process, these parents and offspring undergo the selection process using priority selection strategy. In this strategy, the population is arranged in descending order according to the individual fitness value of each parent or offspring. Thus, the set of random numbers with highest R value will be on the top while the set of random numbers with lowest R value will be ranked at the bottom. The top 20 sets of random numbers are later transcribed as the new generation of sets of random numbers.

Next, convergence test is performed to decide whether the search for the best set of random numbers should be continued or stopped. In this study, the stopping criteria are defined by the following equations.

$$R_{\max} - R_{\min} \leq 0.001 \quad (1)$$

$$x_{1,\max} - x_{1,\min} = 0 \quad (2)$$

The first equation describes that the difference between the maximum fitness value and the minimum fitness value in the new generation should be less than 0.001 for the search to stop. However, the second equation should also be complied for the whole process to stop. The second equation is added to the set of constraint equations because the final number of nodes recommended by the search should be a single value. If these stopping criteria are not met, the search for optimum set of random numbers is continued. Nonetheless, the new search is performed by repeating the mutation process based on the statistical values obtained from the new generation of random numbers.

V. RESULTS AND DISCUSSION

After successful training of the classical ANN and evolutionary ANN, the performance of EPANN is compared with the performance of the classically designed ANN to investigate the effect of using EP for evolving the ANN. The results for the prediction of total AC power output of the grid-PV system are illustrated in Table I.

TABLE I: TRAINING PARAMETERS AND RESULTS FOR CLASSICAL ANN AND EPANN

Parameters / Results	Classical ANN (without EP)	EPANN
Number of training patterns	1000	1000
Number of nodes in hidden layer, $x1$	3	2
Learning rate, $x2$	0.1000	0.0444
Momentum rate, $x3$	0.1000	0.0556
Training algorithm	trainlm	trainlm
Average correlation coefficient, R	0.9928	0.9932
Type of transfer function	logsig-purelin	logsig-purelin
Training duration	numerous days	3.19 hours

In Table I, the fitness value obtained using classical ANN is 0.9928 whereas the fitness value obtained using EPANN is 0.9932. Hence, the EPANN actually produces higher R value compared to the classical ANN model. The improvement of R is approximately 0.04%. Although the improvement of R is small, this result shows that the R value could be optimized using EPANN. Apart from that, the number of nodes in the

hidden layer using classical ANN and EPANN is found to be 3 nodes and 2 nodes respectively. In addition, the optimum learning rate and momentum rate using EPANN are discovered to be 0.0444 and 0.0556 respectively while the optimum learning rate and momentum rate using classical ANN are discovered to be 0.1000 and 0.1000 respectively. The evolutionary model generates more accurate values for the learning rate and momentum rate compared to the classical ANN. This difference is also caused by the different search range utilized by both techniques. In classical ANN, the search of training parameters was implemented using a smaller search scale compared to the EPANN which uses a broader search range preset by the ANN designer. This characteristic also indicates better capability of the EPANN over the classical ANN. Besides that, as the classical ANN was designed manually through heuristic technique, the training duration of classical ANN is much longer compared to the EPANN which was designed automatically using EP with several preset constraints. The EPANN model was completed within 3.19 hours while the classical ANN was successfully developed after a few days of manual training efforts. In short, these results show that the EPANN model is able to select the optimal values for the number of nodes in hidden layer, learning rate and momentum rate so that the regression performance of the ANN could be optimized.

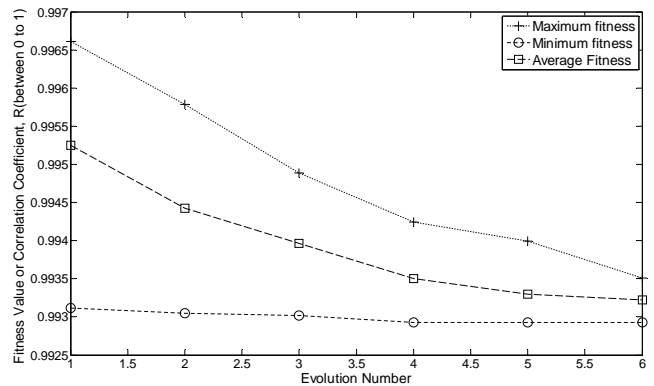


Fig. 3 Plot of Different Types of Fitness Values at Different Evolution Number

The detailed performance of the EPANN is illustrated in Fig. 3 to Fig. 5. In Fig. 3, it can be observed that the fitness value converge at the sixth evolution, i.e the difference between the maximum fitness and minimum fitness is less or equal to 0.001. As the result does not exhibit an absolute convergence, i.e the difference between maximum fitness and minimum fitness is zero, average fitness is introduced to illustrate more presentable value of the fitness.

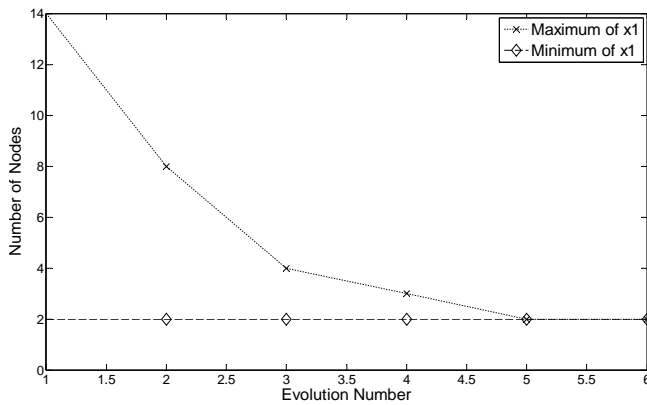


Fig. 4 Plot of Number of Nodes at Different Evolution Number

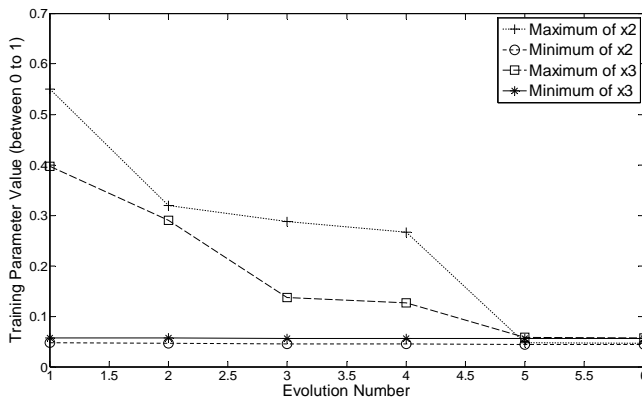


Fig. 5 Plot of Different Types of Training Parameter Values at Different Evolution Number

Unlike the fitness value which converges at the sixth evolution, the number of nodes, x_1 has actually converged earlier at the fifth evolution as shown in Fig. 4. However, the number of nodes, x_1 produces an absolute convergence as required by the stopping criteria. Likewise, the learning rate and momentum rate have also converged at the fifth evolution as illustrated in Fig. 5. Although no hard-limiting stopping criteria were imposed on the learning rate and momentum rate, the graph shows nearly absolute convergence of the learning rate and momentum rate.

VI. CONCLUSION

The EPANN is found to be useful for simplifying and accelerating the search of the optimized size of hidden layer as well as the optimized values for learning rate and momentum rate in ANN training. Nevertheless, due to the random nature of ANN process, the fitness of the EPANN i.e. the correlation coefficient of the prediction may not show complete convergence. Therefore, average fitness has been introduced as a measure of the ANN performance. Apart from that, the EPANN has also shown a superior performance in terms of correlation coefficient and number of nodes compared to the classical ANN. Besides having higher correlation coefficient, this lower number of nodes obtained in EPANN demonstrates that the prediction model can be realized using smaller architecture with different sets of learning rate and momentum rate. In addition, lower learning rate and lower momentum rate produced by EPANN minimize the training effort for the prediction model. In short, this paper has demonstrated that the prediction of total AC

power output of a grid-PV system could be optimized and accelerated using EPANN.

ACKNOWLEDGMENT

The authors would like to thank the School of Photovoltaic Engineering, University of New South Wales, Australia for making the training data and testing data available for study.

REFERENCES

- [1] I. Ashraf and A. Chandra, "Artificial neural network based models for forecasting electricity generation of grid connected solar PV power plant", *Int. Journal of Global Energy Issues*, vol. 21, no. 1/2, pp. 119-130, 2004.
- [2] M. Balzani and A. Reatti, "Neural network based model of a PV array for the optimum performance of PV system", in *Proc. PhD Research in Microelectronics and Electronics Conf.*, vol. 2, 2005, pp. 123-126.
- [3] X. Yao, "Evolving artificial neural networks", in *Proc. Of the IEEE*, vol. 87, no. 9, 1999.
- [4] S.I. Sulaiman, T.K. Abdul Rahman, and I. Musirin, "ANN-based technique with embedded data filtering capability for predicting total AC power from grid-connected photovoltaic system", in *Proc. 2nd International Power Engineering and Optimization Conference (PEOCO2008)*, 2008, pp. 272-277.
- [5] X. Yao and Y. Liu, "Towards designing artificial neural networks by evolution", *Applied Mathematics and Computation*, vol. 91, pp. 83-90, 1998.
- [6] D.B. Fogel, "An introduction to simulated evolutionary optimization", *IEEE Transactions on Neural Networks*, vol. 5, pp. 3-14, 1994.
- [7] L.J. Fogel, "Autonomous automata", *Industrial Research*, vol. 4, no. 1, pp.14-19, 1962.
- [8] T. Back and H.-P. Schwefel, "Evolutionary computation: an overview", in *Proc. IEEE International Conference on Evolutionary Computation (ICEC '96)*, 1996, pp. 20-29.
- [9] D.B. Fogel, L.J. Fogel, and V.W. Porto, "Evolutionary programming for training neural networks", in *Proc. International Joint Conference on Neural Networks*, vol. 1, pp. 601-605, 1990.
- [10] M. Sarkar and B. Yegnanarayana, "Feedforward neural networks configuration using evolutionary programming", in *Proc. International Conference on Neural Networks*, vol. 1, 1997, pp. 438-443.
- [11] K. Peng, S.S. Ge, and C. Wen, "An algorithm to determine neural network hidden layer size and weight coefficients", in *Proc. 15th IEEE International Symposium on Intelligent Control (ISIC 2000)*, 2000, pp. 261-266.
- [12] B. Yegnanarayana, *Artificial Neural Networks*, New Delhi: Prentice Hall of India, 2006, ch. 1.
- [13] S.R. Wenhams, M.A. Green, and M.E. Watt. *Applied Photovoltaics*. Centre for Photovoltaic Devices and Systems, Sydney: The University of New South Wales, 1995, ch. 1.
- [14] W.M. Jenkins, "Neural network weight training by mutation", *Computers & Structures*, vol. 84, pp. 2107-2112, 2006.
- [15] M. Annunziato, I. Bertini, A. Pannicelli and S. Pizzuti, "Evolutionary feed-forward neural networks for traffic prediction", in *Proc. International Congress on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN 2003)*, 2003, pp. 1-8.
- [16] J. Fang and Y. Xi, "Neural network design based on evolutionary programming", *Artificial Intelligence in Engineering*, vol. 11, pp. 155-161, 1997.
- [17] M.F. Augusteijn and T.P. Harrington, "Evolving transfer functions for artificial neural networks", *Neural Computing and Applications*, vol. 13, pp. 38-46, 2004.
- [18] I. Musirin and T.K. Abdul Rahman, "Evolutionary programming based optimization technique for maximum loadability estimation in electric power system", in *Proc. National Power and Energy Conference (PECon)*, 2003, pp. 205-210.

Shahril Irwan Sulaiman became a Member (M) of **IEEE** in 2008. He was born in Kelantan, Malaysia. He received his B.Eng in Electrical & Electronics Engineering from Universiti Tenaga Nasional (UNITEN), Malaysia in 2002 and M.Eng.Sc in Photovoltaic Engineering from University of New South Wales, Australia in 2003. Since 2004, he has been a lecturer in Department of Electronics, Faculty of Electrical Engineering, Universiti Teknologi MARA (UiTM), Malaysia. His research interests are in the areas of photovoltaics (PV), renewable energy technologies and artificial



intelligence. He is currently working towards his PhD degree at Universiti Teknologi MARA (UiTM), Malaysia.

E.E. (Hons) and PhD on Loughborough Technology and Malaysia on 1996 Professor at the Faculty and currently the Dean Engineering, Universiti Malaysia. She has technical papers and has been also supervising post-graduate students. Her research interest includes voltage profile studies, artificial neural network, Evolutionary Computation such as Evolutionary Programming and Genetic Algorithm, Artificial Immune System (AIS), Economic Dispatch and Loss Minimization. She is currently the secretary for the Power and Energy Society (PES) Chapter, IEEE Malaysia Section.



Titik Khawa Abdul Rahman received BSc 1988 from University of University of Malaya, respectively. She is a of Electrical Engineering of Faculty of Electrical Teknologi MARA written more than 80



Dr. Ismail Musirin obtained Diploma of Electrical Power Engineering in 1987, Bachelor of Electrical Engineering (Hons) in 1990; both from Universiti Teknologi Malaysia, MSc in Pulsed Power Technology in 1992 from University of Strathclyde, United Kingdom and PhD in Electrical Engineering from Universiti Teknologi MARA (UiTM), Malaysia in 2005. He is currently an Associate Professor and Chair, Centre of Electrical Power Engineering Studies (CEPES),

Faculty of Electrical Engineering, UiTM. He has published 2 books and more than 100 technical papers in the international and national, conferences and journals. He is reviewer for IEEE, IET and WSEAS journals and conferences. His research interest includes power system stability, optimization techniques, distributed generator and artificial intelligence.