International Journal of Power Electronics and Drive System (IJPEDS)
Vol. 10, No. 1, March 2019, pp. 548~561
ISSN: 2088-8694, DOI: 10.11591/ijpeds.v10.i1.pp548-561

Critical evaluation of soft computing methods for maximum power point tracking algorithms of photovoltaic systems

Norazlan Hashim[1], Zainal Salam[2]
[1]Faculty of Electrical Engineering, Universiti Teknologi MARA, Malaysia
[2]School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, Malaysia

Article Info

Article history:

Received Jun 12, 2018
Revised Nov 29, 2018
Accepted Dec 16, 2018

Keywords:

CS
DE
EP
GA
MPPT
Partial shading
PSO
Soft computing (SC)

ABSTRACT

With the proliferation of numerous soft computing (SC)–based maximum power point tracking (MPPT) algorithms for photovoltaic (PV) systems, determining which algorithm performs better than others is becoming increasingly difficult. This is primarily due to the absence of standardized methods to benchmark their performances using consistent and systematic procedures. Moreover, the module technology, power ratings, and environmental conditions reported by numerous publications all differ. Based on these concerns, this paper presents a critical evaluation of the five most important and recent SC-based MPPTs, namely, genetic algorithm (GA), cuckoo search (CS), particle swarm optimization (PSO), differential evolution (DE), and evolutionary programming (EP). To perform a fair comparison, the initialization, selection, and stopping criteria for all methods are fixed in similar conditions. Thus, the performance is determined by its respective reproduction process. Simulation tests are performed using the MATLAB/SIMULINK environment. The performance of each algorithm is compared and evaluated based on its speed of convergence, accuracy, complexity, and success rate. The results indicate that EP appears to be the most promising and encouraging SC algorithm to be used in MPPT for a PV system under the multimodal partial shading condition.

Corresponding Author:

Norazlan Hashim,
Faculty of Electrical Engineering,
Universiti Teknologi MARA, 40450 Shah Alam, Malaysia.
Email: azlan4477@salam.uitm.edu.my / azlan4477@yahoo.com

1. INTRODUCTION

Maximum power point tracking (MPPT) is a control algorithm embedded inside a DC–DC power converter to extract the maximum power from a photovoltaic (PV) array. The objective is to ensure that the power to be extracted always matches the peak value of the power-voltage (P-V) characteristic curve under varying solar irradiation (G) and temperature (T). The idea of the tracking is to lock the converter operating voltage and current to the maximum power point (MPP) of the PV array. The MPP, when in a normal uniform irradiance condition, exhibits a unique peak of the P-V curve. The tracking has to be accomplished rapidly to ensure that the power is not lost during the changes in G and T. In addition, the MPPT must be able to correctly locate the MPP during the occurrence of partial shading—a condition in which a portion of the PV array is shaded while other parts remain uniformly irradiated. During partial shading, the P-V curve exhibits multiple peaks, thus transforming the problem from single modal to multimodal, that is, with multiple maxima points. Thus, the problem becomes much more complicated as the MPPT algorithm needs to continuously track the continuous variations of several peaks that changes with G and T.

Despite a large number of conventional MPPT algorithms published in literature [1]-[5], only several methods are widely implemented, namely, the perturb and observe (P&O), incremental conductance (IC), and hill climbing (HC) methods. These algorithms are based on checking the slope of the curve periodically to ensure that the peak is detected (when the slope is zero). Generally, it operates satisfactorily under the uniform irradiance condition, that is, when the P-V curve has a unique peak. However, during partial shading, the algorithm cannot locate the correct MPP because the problem has transformed to a multimodal, and it cannot differentiate between the local and global peaks. This is inevitable because the nature of these algorithms is based on the peak detection principle, that is, when it locates a perceived maximum point, it locks itself within the vicinity of that point. If the peak is local, substantial loss of PV power results. To address this problem, a soft computing (SC) MPPT is proposed. Since the SC algorithm searches for all the peaks over the entire P-V curve, finding the global MPP is very likely. The authors in [1]-[3], [5]-[16] have done extensive reviews on the application of SC for MPPT; these include fuzzy logic controller (FLC), artificial neural network (ANN), particle swarm optimization (PSO), genetic algorithm (GA), differential evolution (DE), ant colony optimization (ACO), Bayesian fusion (BF), cuckoo search (CS), and chaotic search (ChS).

With the proliferation of SC-based MPPT techniques (and their variations), determining which algorithm is more effective than others is difficult as no proper evaluation to criticall—primarily because of the fact that no two methods are compared fairly, nor are they verified independently. This is because in most published works, the module technology, experimental setup, power ratings, and environment conditions (particularly the variations in G and T) in which the PV system setup was subjected to are all different. In addition, the partial shading experiments that have been carried out are never unique. This raises questions on the legitimacy of the claims as different shading patterns result y assess their performances exists. The authors' claims on the superiority of their own techniques are unjustifiable in different MPPT efficiencies.

With regard to these concerns, this paper aims to provide a standardized procedure to critically evaluate the performances of various SC MPPT techniques. Three well-established methods are considered, namely, PSO, GA, and DE, along with two recently proposed algorithms, cuckoo search (CS) and evolutionary programming (EP). Although there exist several comparative studies among GA, DE, and PSO, they only offer general reviews without any evaluation on their respective performances. Each algorithm is assessed in terms of accuracy, speed, complexity, and success rate of convergence. Two statistical procedures—namely, the mean absolute error (MAE) and standard deviation (STD)—are used for benchmarking. In addition, the relative complexity of the algorithm is determined by measuring the average CPU time taken for each iteration. The proposed evaluation will assist the researchers and practitioners in selecting the best algorithm to design their MPPT applications.

2. OVERVIEW OF SC-BASED MPPT

Soft computing (SC) is a collection of flexible, adaptable, and intelligent problem-solving methods to exploit the tolerance for imprecision to achieve tractability, robustness, and low-cost solutions [17]. In general, SC can be classified into two broad categories, single-point and population-based search. For the former, the algorithm scans the solutions in the whole search space of the problem (in the case for the PV system, the search space is the entire P-V curve) using one agent at a time. On the other hand, for the population-based search type, the algorithm operates on several agents (in parallel) within the search space. The latter is unique because these simple agents cooperate and interact with one another to accomplish complex tasks. To date, the reported SC-based MPPT algorithms used to solve partial shading problems are shown in Figure 1. Some of the important features of these methods have been described briefly in the introduction. In this paper, only population-based algorithms will be discussed given their superiority in solving multimodal optimization problems.

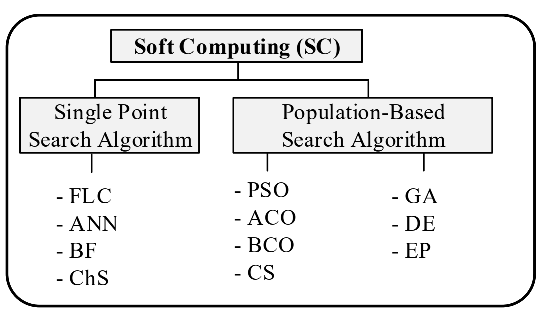

Figure 1. Reported SC-based MPPT algorithms used to solve partial shading problem

2.1. Generalized processes

The way in which the population-based SC optimizes the solution can be generalized into three major processes, namely, initialization, reproduction, and selection. In the initialization step, the initial parent for the population size with n candidates is generated. In reproduction, the offspring are created from the selected parents through a uniquely formulated equation, according to the SC type. Finally, the selection stage is the discriminatory process to choose the individuals to survive for the next generation. The reproduction and selection processes are repeated iteratively until a prespecified stopping criterion is met.

2.2. Initialization

In the context of MPPT, the term population is referred to the power converter's (normally DC–DC) duty cycle (D), while the optimal solution is the maximum PV array's output power (P_{PV}). A small population size leads to poor solutions; on the other hand, a large population increases the computation time. Hence, a trade-off is needed to achieve credible solutions with a reasonable number of iterations. Various methods to choose population size are recommended in [18, 19]; in this study, the population size was set to 5 because it was found (by trial and error) to produce the best results. In most cases, the values attached to the initial population ($D_{initial}$) are generated randomly. However, randomness produces different results on successive runs, even if those runs were initialized identically. To eliminate this uncertainty, the controlled initialization method is preferred. For n population, the initialization with uniformly distributed elements within the interval $0 \leq D \leq 1$ is given by the following:

$$D_{initial} = \begin{bmatrix} \frac{1}{n+1} & \frac{2}{n+1} & ..... & \frac{n}{n+1} \end{bmatrix} \times 1 \tag{1}$$

So at the beginning of the search, five different values of D with accordance to (1) are used to find the best value of P_{PV}.

2.3. Reproduction

Reproduction is the most crucial step as it differentiates the ability of the algorithm to produce the next population generation. The first selected population is called the parent ($D_{initial}$); the second and subsequent population (after going through the reproduction) is called the offspring (D_{New}). Swarm-based algorithms (PSO, ACO, and CS) are based on the social behavior of insects or animal. They utilize specific reproduction operators such as particle velocity (for PSO) and Lévy flight (for CS) to create D_{New}. On the other hand, evolutionary-based algorithms (EP, DE, and GA) generate D_{New} through natural genetics evolution. They use genetic operators such as crossover (also called recombination) and mutation. The crossover exchanges some parts of two individuals, while the mutation operator changes the value of the randomly chosen individual.

2.4. Selection

Selection is the process to discriminate (isolate) the best individuals for the next generation (D_{New}). It is based on the fulfillment of criteria set by the fitness function. The selection should be chosen such that it converges to the global optimum solution (i.e., P_{PV_Best}) without having to sacrifice too much convergence speed. There exist selection schemes proposed in literature; the most common are roulette wheel, tournament, ranking, and steady state selection. A comprehensive analysis of all these schemes has been reported in [20], [21]. In this work, the ranking selection scheme is chosen given its simplicity and, at the same time, yields good results. The basic equation for this scheme is given by the following:

$$D_{New} = \begin{cases} D_{New} & if\ f(D_{New}) \geq f(D_{Old}) \\ D_{Old} & else \end{cases} \tag{2}$$

2.5. Stopping criterion

The stopping criterion is the terminating condition that halts the algorithm. It occurs when one or more prescribed conditions are met. The most commonly used stopping criteria are the following:

a. Generation Number — A threshold value is set. The algorithm stops the iteration after carrying out a certain number of iterations.
b. Best Fitness Threshold — This stops the iteration when the maximum value of objective function (P_{PV_Best}) is less than the set value ($P_{PV\_Specified}$).
c. Population Convergence — This stops the iteration when the difference between the maximum and minimum values of all individuals (D_{New}) in the population is less than the prescribed tolerance.

d. Fitness Convergence — This stops the iteration when the difference between the maximum and minimum values of objective function (P_{PV}) for all individuals (D_{New}) is less than the prescribed tolerance.

In this study, the fitness convergence, that is, the PV power (P_{PV}), is chosen as the stopping criterion because it gives better results than the others. This tells the algorithm to stop searching for the optimum solution (P_{PV_Best}) when the fitness of all individuals are quite close to one another, that is, within the range of 1 W. Smaller tolerances result in greater simulation accuracy but, in general, lower convergence speed.

3. SELECTED SC ALGORITHMS

Each selected SC algorithm has its own reproduction operator parameter: crossover constant (CR) and mutation rate (F) for GA and DE, search step (α or β) for CS and EP, and acceleration constants (C_1 and C_2) for PSO. The main consequence of the operator is the step size; if the step size is large, the search is rapid, but the targeted global peak may be missed. On the other hand, if the step size is too small, the search would be very long; most probably, the irradiance has changed to a new value before the global peak is successfully tracked. In practice, trial-and-error tuning determines the parameters' values that yield the best optimized results (P_{PV_Best}). The optimization is performed before the execution of the algorithm, and these values are fixed throughout the run. However, choosing the right parameter is often time-consuming; the normal procedure is to set the parameter value and then observe the results. Furthermore, because random functions exist in the reproduction formula, the search result of each method varies at each iteration. To address this, the simulation was run with 100 trials, and the results are averaged. The best values of the reproduction parameters for each algorithm are tabulated in Tables 1-5. For consistency, each SC algorithm is implemented based on the proposed benchmark methodology as discussed in the previous section.

3.1. Genetic algorithm (GA)

GA is an optimization algorithm inspired by natural genetic evolution and selection. To produce a new offspring, GA uses two main genetic operators, namely, crossover and mutation. The reproduction operator of the GA algorithm used in this paper can be described as follows [22]-[24]:

a. Select two candidates from the parent population (Parent1 and Parent2) at random; they must be mutually different from each other.
b. Apply a single-point crossover and mutation operator to yield an offspring population according to the following:

$$\begin{aligned} &\text{Offspring}_1=\alpha\cdot\text{Parent}_1+(1-\alpha)\cdot\text{Parent}_2 \\ &\text{Offspring}_2=(1-\alpha)\cdot\text{Parent}_1+\alpha\cdot\text{Parent}_2 \\ &\text{Offspring}_{3\text{-}5}=\pm\beta+\text{Parent}_{3\text{-}5} \end{aligned} \tag{3}$$

where α is the crossover rate and β is the mutation rate. The values of the GA reproduction parameters used in this study are tabulated in Table 1.

Table 1. GA parameters

Parameters	Values
Population size, NP	5
Crossover rate, α	∈[± 0.8]
Mutation rate, β	∈[± 0.05]
Maximum generations, Gmax	25

3.2. Particle swarm optimization (PSO)

PSO attempts to mimic the social behavior of flocking birds when searching for food. In PSO, each individual of the potential solution, called a particle, flies around in a multidimensional search space, looking for the optimal solution based on its own and its neighbors' experiences. The reproduction operator of the PSO algorithm used in this paper can be described as follows [25]-[29]:

a. Determine the particle's best known position, P_{best}, and the population's best known position, G_{best}.
b. Calculate the parent velocity to yield an offspring population according to the following:

$$\begin{aligned} &\text{Vel}_{i+1}=K\cdot\left[\text{Vel}_i+C_1\cdot\text{rand}\left(P_{besti}-\text{Parent}_i\right)+C_2\cdot\text{rand}\left(G_{besti}-\text{Parent}_i\right)\right] \\ &\text{Offspring}_{i+1}=\text{Parent}_i+\text{Vel}_{i+1} \end{aligned} \tag{4}$$

where K is the inertia weight and C_1 and C_2 is the acceleration constant. The PSO parameters used in this study are tabulated in Table 2.

Table 2. PSO parameters

Parameters	Values
Population Size, NP	5
Acceleration Constants, C_1=C_2	1.5
Inertia weight, W	0.5
Maximum Generations, G_{max}	25

3.3. Differential evolution (DE)

DE is a simple evolutionary algorithm using similar operator-like GAs such as crossover and mutation. The main difference is that GA relies primarily on crossover, while DE relies on mutation operation. DE creates an offspring by combining the parent individual and several other individuals of the same population. In this paper, the "DE/rand-to-best/1/bin" scheme has been selected because of its good performance for the case under study. DE's reproduction operator can be described as follows 10], [30], [31]:

a. Select the population's best known individual, G_{best}.
b. Select two candidates from the parent population (Parent1 and Parent2) at random; they must be mutually different from each other.
c. Apply mutation and crossover operators to produce the trial individual and offspring according to the following:

$$\begin{aligned} &\text{Trial}_i=\text{Parent}_i + F\cdot\left(G_{best}\text{-Parent}_i\right) + F\cdot\left(\text{Parent}_1\text{-Parent}_2\right) \\ &\text{Offspring}_i=\begin{cases}\text{Trial}_i, \text{ if rand} < CR \\ \text{Parent}_i, \text{ otherwise}\end{cases} \end{aligned} \tag{5}$$

where F is the mutation rate and CR is the mutation rate. The DE parameters used in this study are tabulated in Table 3.

Table 3. DE parameters

Parameters	Values
Population Size, NP	5
Crossover Rate, CR	0.9
Mutation Rate, F	0.7
Maximum Generations, Gmax	25

3.4. Cuckoo search (CS)

CS is inspired by the obligate brood parasitism of some species of a bird family called cuckoo in combination with the Lévy flight behavior of some birds and fruit flies. The concept of CS is similar to PSO (using particles), but the step sizes in CS are characterized by the random walk based on Lévy flight. Mathematically, Lévy flight has movement lengths chosen from a probability distribution with a power-law tail, $Levy(\lambda) \sim x^{-\lambda}(1<\lambda<3)$, where x is the step length and λ is the variance. The reproduction operator of the CS algorithm can be described as follows [12], [32]-[34]:

a. Select the population's best known individual, G_{best}.
b. Apply a Lévy flights operator to yield an offspring population according to the following:

$$\begin{aligned} &\sigma_i= \beta\cdot\left(\alpha\cdot\text{randn/abs(randn)}\right)^{\frac{1}{1.5}}\cdot(\text{Parent}_i\text{-}G_{best}) \\ &\text{Offspring}_i=\text{Parent}_i+\sigma_i\cdot N_i(0,1) \end{aligned} \tag{6}$$

where α is the Lévy coefficient and β is the scaling factor. The CS parameters used in this study are tabulated in Table 4.

Table 4. CS parameters

Parameters	Values
Population Size, NP	5
Levy coefficient, α	0.7
Scaling factor, β	0.01
Maximum Generations, Gmax	25

3.5. Evolutionary programming (EP)

EP is a search algorithm designed to simulated evolution that iteratively generates increasingly appropriate solutions. It was first proposed as an alternative approach to classic artificial intelligence (AI) in computers. EP has the advantage of using a mutation-only reproduction operator and can easily be designed for adapting the parameters of the mutation operator during the reproduction process. In this paper, classical EP, which uses a Gaussian distribution function for updating the offspring, has been selected because of its ease of use and provides comparatively good results. The reproduction operator of the EP algorithm can be described as follows [35-39]:

Apply mutation operator to yield an offspring population according to the following:

$$\begin{aligned} &\sigma_i = \beta \cdot (P_{PV_i} / P_{PV_{max}}) \\ &Offspring_i = Parent_i + \sigma_i \cdot N_i(0,1) \end{aligned} \tag{7}$$

where β is the scaling factor. All EP parameters used in this study are tabulated in Table 5.

Table 5. EP parameters

Parameters	Values
Population Size, NP	5
Scaling factor, β	0.01
Mutation Technique	Gaussian∈[± σ]
Maximum Generations, Gmax	25

4. BENCHMARKING METHODOLOGY FOR SC-BASED MPPT

To evaluate the performances of different SC MPPT algorithms fairly, a standardized evaluation process is required. Unfortunately, this process is absent in previous literature; thus, the performances of the MPPT algorithms are not verified independently. Moreover, the module technology, power ratings, and environment conditions in which the experiments were set up are all different. In addition, the partial shading experiments carried out are never unique. This raises questions on the legitimacy of the claims as different shading patterns result in different MPPT efficiencies. With these concerns, this paper attempts to propose a methodology to benchmark the SC MPPT algorithms based on a simple flow diagram shown in Figure 2. Because of their recent popularity, five different algorithms are chosen, namely, differential evolution (DE), evolutionary programming (EP), cuckoo search (CS), particle swarm optimization (PSO), and genetic algorithm (GA).

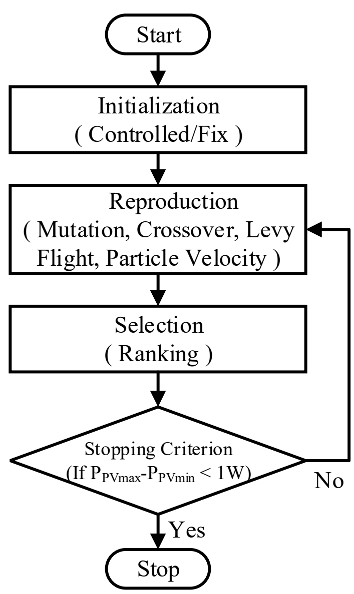

Figure 2. The benchmark methodology for population-based SC algorithms

4.1. Problem formulation

For consistency, the initialization, selection, and stopping criteria are fixed at the same conditions as discussed in the previous section. Thus, the performance of each algorithm is tested based on its own unique metaphor in the reproduction stage. Furthermore, benchmarking the performance of MPPT using the normal irradiance (uniform) condition is not adequate as the resulting P-V curve is single modal (i.e., with only one unique peak), and with such a simple condition, all MPPTs are able to converge to the peak very quickly. As a result, clearly differentiating the performances of the algorithm is difficult.

A more challenging situation is to subject the PV system to partial shading condition. The phenomena are due to the shadows from clouds, neighboring buildings, trees, chimneys, towers, etc., where certain parts of the PV array are shaded while others receive uniform irradiance. During partial shading, the shaded modules experience a large amount of rush currents, resulting in excessive heat (hot spot) that may cause permanent damage. To relieve the stress on the shaded modules, bypass diodes are fitted across them [15]. However, multiple peaks in the P-V curve are then created. Consequently, the problem is transformed from single modal to multimodal. This condition poses a serious challenge to any MPPT technique because of the difficulty to distinguish the global from the local peaks.

5. PV SYSTEM MODELING

The two-diode PV cell model [40]-[42], depicted in Figure 3, is utilized for simulation. It is chosen because of its superior accuracy, particularly at a low irradiance level. The output current of the cell is given by the following:

$$I = I_{PV} - I_{o1} \cdot \left[\exp\left(\frac{V+IR_s}{a_1 V_{T1}}\right) - 1\right] - I_{o2} \cdot \left[\exp\left(\frac{V+IR_s}{a_2 V_{T2}}\right) - 1\right] - \left(\frac{V+IR_s}{IR_p}\right) \quad (8)$$

Where I_{o1} and I_{o2} are the reverse saturation currents of diodes 1 (D_1) and 2 (D_2), respectively, V_{T1} and V_{T2} are the thermal voltages of the respective diodes, and a_1 and a_2 represent the diode ideality constants. The I_{o2} term in (9) compensates the recombination loss in the depletion region, as described in [41].

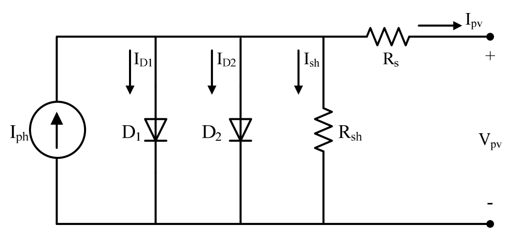

Figure 3. A Two-diode model of PV cell

For a string with *N* number of modules in series (N_{cell}), (9) can be extended to the following:

$$I = I_{PV} - I_{o1} \cdot \left[\exp\left(\frac{V+IR_s}{a_1 V_{T1}}\right) - 1\right] - I_{o2} \cdot \left[\exp\left(\frac{V+IR_s}{a_2 V_{T2}}\right) - 1\right] - \left(\frac{V+IR_s \times N_{cell}}{IR_p \times N_{cell}}\right) \quad (9)$$

The simulations model of the PV system was based on a MATLAB/Simulink simulator developed in [43]. The array is simulated using the BP MSX-60 module. Its specifications at the standard test conditions (STC) are shown in Table 6.

Table 6. Electrical parameters of MSX-60 module at STC

Parameters	Values	Parameters	Values
Maximum Power (P_{max})	60 W	Temperature coefficient of V_{oc}	-(80±10) mV/^{0}C
Voltage at Pmax (V_{mpp})	17.1 V	Temperature coefficient of I_{sc}	-(0.065±0.015) % / ^{0}C
Current at Pmax (I_{mpp})	3.5 A	Temperature coefficient of power	-(0.5±0.05) % / ^{0}C
Open circuit voltage (V_{oc})	21.1 V	NOCT	47±2 ^{0}C
Short circuit current (I_{sc})	3.8 A	Operating Temperature	25 ^{0}C

For simplicity, only a stand-alone system with a DC–DC boost converter load is considered. The circuit is shown in Figure 4. It consists of five modules in a series, connected to the converter with the MPPT controller. The system need not be extended to a grid-tied one because the objective is to evaluate the performance of the MPPT, which is on the DC side. The optimum value of the circuit components used in this study are discussed in detail in [44]. The converter-switching frequency (fs) is 20 kHz. Meanwhile, the inductor (L) is set to 1 mH, the filter capacitor (C_1 and C_2) value is 47 μF, and the load resistor (R) is 200 Ω. All the MPPT algorithms are coded using the M-file. The input variables are G and T.

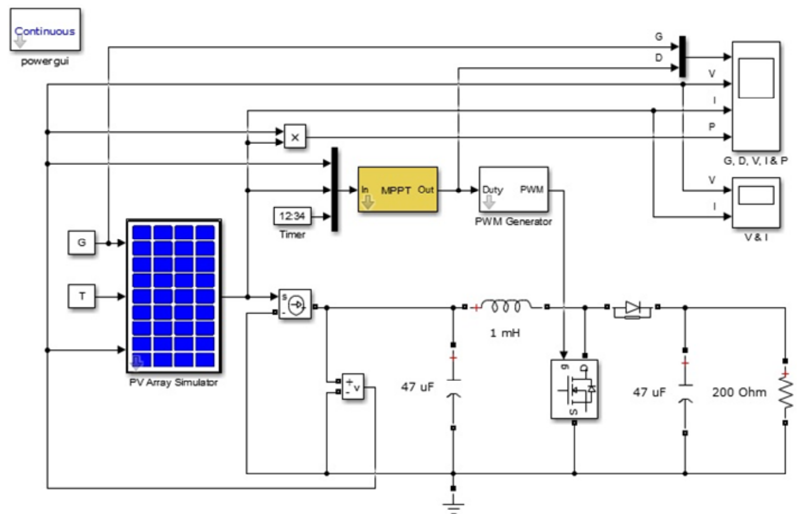

Figure 4. The simulation model of stand-alone PV system

The goal of the optimization is to track the MPP as fast as possible and with the highest consistency. In terms of objective function formulation, that goal can be described as the following:

$$f = \max\left\{\sum_{n=1}^{N} V_{PVn} \cdot I_{PVn}\right\} \tag{10}$$

The objective (fitness) function (f) is the output power of the PV, while N is the number of modules. Variables V_{PV} and I_{PV} are the PV array's output voltage and current, respectively. During initialization, five different values of duty cycle (D_1 to D_5) are generated with accordance to (2). Each of these duty cycles will be sent to the PWM block to generate a PWM switching waveform to the MOSFET at a sampling rate of 0.1 s [44]. Then the MPPT block will calculate the PV power based on the sensing PV array voltage (V_{PV}) and current (I_{PV}). The same process will be repeated for each iteration.

6. RESULTS AND ANALYSIS

In this study, the array in Figure 5 is partially shaded with five different values of irradiance patterns, as described in Table 7. Because of the operation of the bypass diode, the step waveform I-V curve shown in Figure 6 is created. Figure 7 shows the resulting P-V curve. Besides the global peak (MPP), the curve exhibits four other local peaks. The MPP voltage and current are located at 51.479 V and 2.181 A, respectively, while the maximum power (i.e., the final fitness value) to be achieved is 112.278 W.

The performance of each SC-based MPPT algorithm is evaluated based on several criteria, namely, speed, accuracy, complexity, and success rate of convergence. The overall results of each performance criterion are tabulated in Table 8. Also, the ranking of each criteria is indicated by subscript (in bracket) number in Table 8.

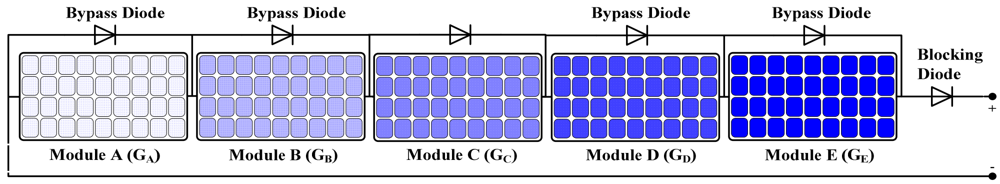

Figure 5. Five PV modules connected in series under partial shaded condition given in Table 7

Table 7. Irradiance for shading patern

Module	A (G_A)	B (G_B)	C (G_C)	D (G_D)	E (G_E)
Irradiance (G = 1.0 = 1000 W/m^2)	0.2	0.4	0.6	0.8	1.0

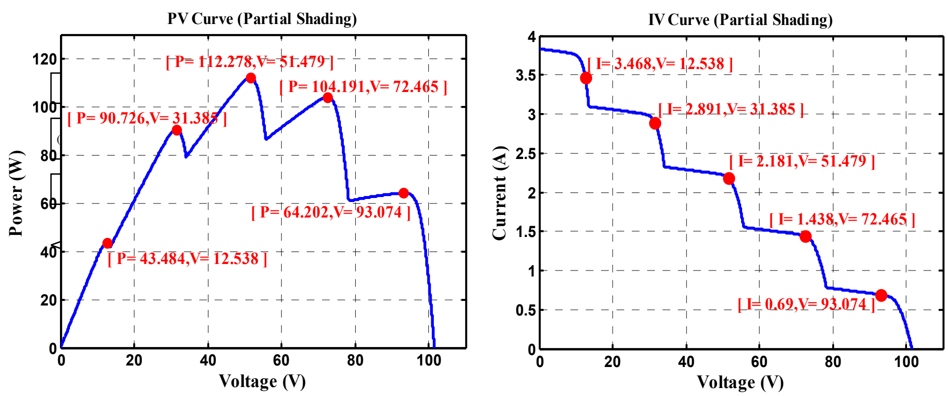

Figure 6. P-V & I-V characteristics under partial shaded condition given in Table 7

6.1. Speed of convergence

The speed of convergence is the number of iterations required by the algorithm to reach the final fitness value. Because of their stochastic nature, the algorithms produce different results at each run. This can be observed by the variation in the trajectories produced by every run, as shown in Figures 7–11. Therefore, taking a conclusion from a single run would not be a fair representation of the algorithm's performance. To overcome this ambiguity, each MPPT method is executed for 100 runs, and the results are averaged as in Figure 12. The iteration limit at each run is set at 15 since most of the methods converge to the solution in less than this prescribed value.

As can be seen, EP is the fastest algorithm to reach MPP (P_{PVmax}) convergence. In average, it requires 6 iterations. Moreover, its convergence trajectories are less scattered as compared to others; GA requires 8, while CS, PSO, and DE converge to MPP after 10 iterations. The rapid convergence of EP is due to the simple Gaussian distributed random numbers in generating an offspring, as described by (8). The random numbers are generated by the controlled scaling factor (σ_i). As the iteration increases and the tracked power approaches MPP, σ_i will decrease as shown in Figure 13. As a result, the step size will decrease and thus prevent unnecessary searching within the area where the global MPP does not exist.

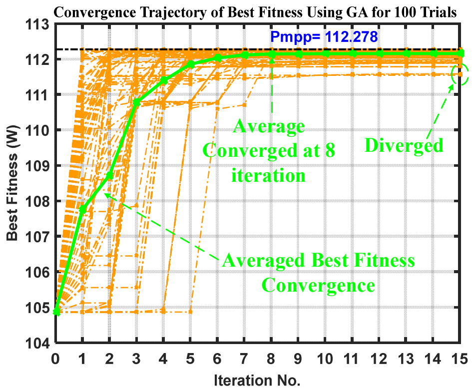

Figure 7. Convergence trajectory of tracked best fitness (MPP) for GA

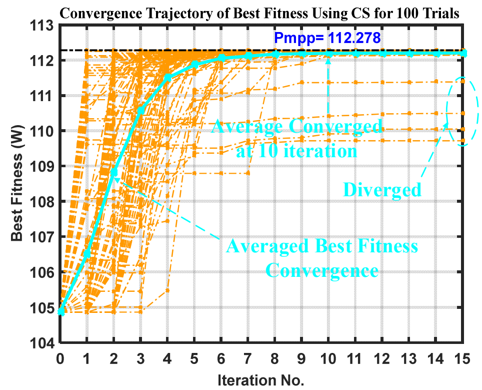

Figure 8. Convergence trajectory of tracked best fitness (MPP) for CS

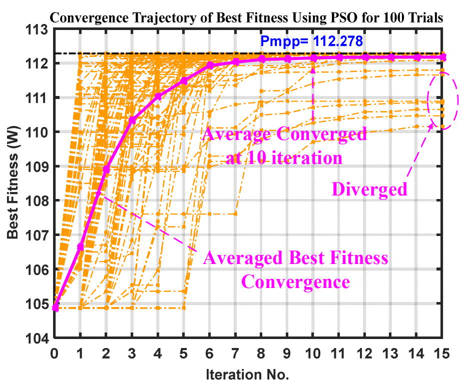

Figure 9. Convergence trajectory of tracked best fitness (MPP) for PSO

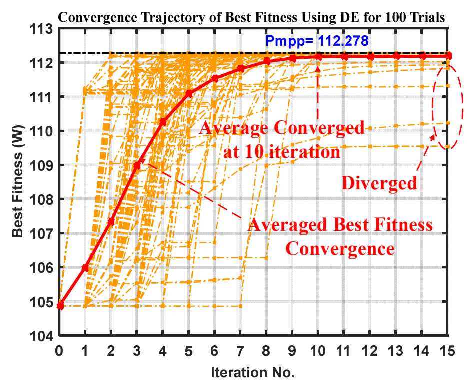

Figure 10. Convergence trajectory of tracked best fitness (MPP) for DE

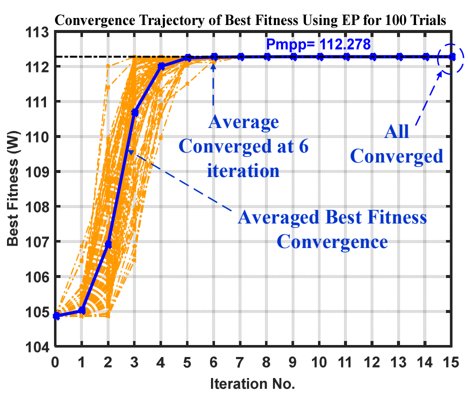

Figure 11. Convergence trajectory of tracked best fitness (MPP) for EP

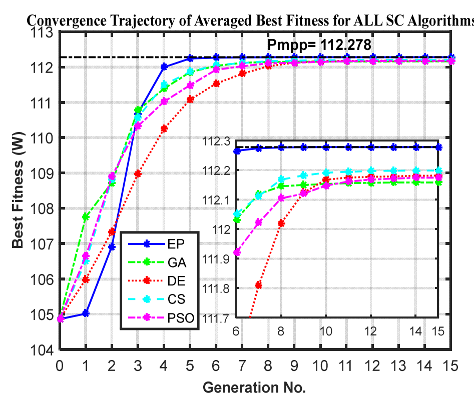

Figure 12. Averaged convergence trajectory of tracked best fitness (MPP) for ALL algorithms

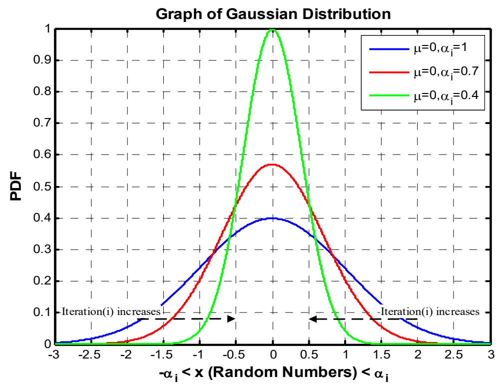

Figure 13. Gaussian distribution (σi decreases as the number of iterations increases)

6.2. Accuracy

Figure 14 shows the distribution of tracked fitness (MPP) for all algorithms for 100 runs. The accuracy is defined as the closeness of the tracked value to the maximum fitness value (i.e., the global MPP). To evaluate the accuracy, two statistical analyses—namely, mean absolute error (MAE) and standard deviation (STD)—were performed. MAE is an average value of the absolute error used to measure the closeness of the predictions to the expected outcomes. It is computed using the following:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\mathrm{err}_i| = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| \tag{11}$$

where f_i is the prediction value and yi the expected value. Meanwhile, STD is a measure of variability or diversity; it shows how much variation or dispersion exists from the average (mean or expected value). The standard deviation is given by the formula:

$$\mathrm{STD} = \sqrt{\sum_{i=1}^{n}\frac{(f_i-\mu)^2}{n}} \tag{12}$$

where f_i represents each value in the population, μ is the mean value of the population, and n is the number of values in the population. The results are tabulated in Table 8. The most accurate algorithm is EP; it exhibits the lowest MAE (0.052) and STD (0.111). The tracked MPPs are very close to the global MPP with the minimum value of a tracked MPP of 111.707 W. The least accurate algorithm is GA, which has the highest MAE (1.302) and STD (1.910); the tracked MPPs are scattered, away from the global MPP.

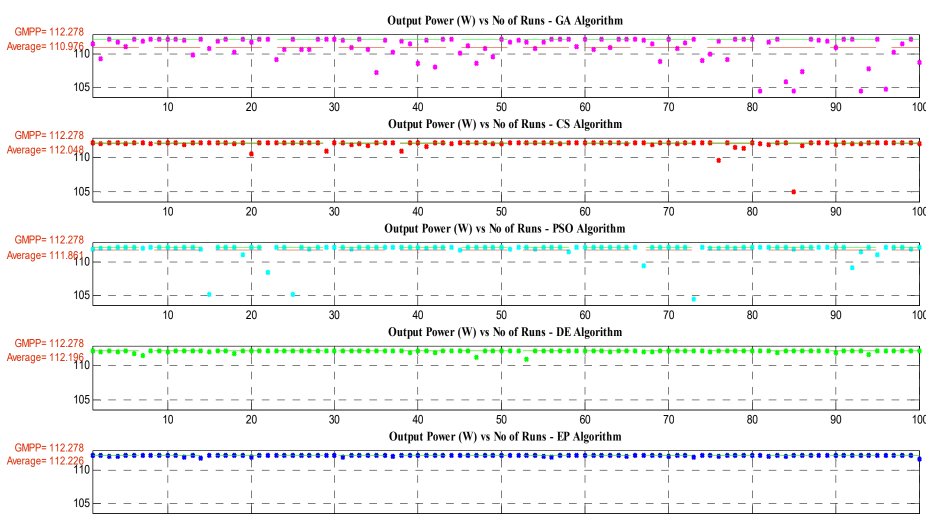

Figure 14. Distribution of tracked fitness under 100 runs for, GA, CS, PSO, DE, and EP

Table 8. Summary of analysis results

SC	Fitness (Output Power, W) Min	Max	Meanu	Accuracy: Mean Absolute Error, MAE	Accuracy: Standard Deviation, STD	Success Rate (%)	Convergence Speed (Average of Iterations No.)	Complexity (Average of CPU Time (ms))
GA	104.345	112.278	110.976	1.302 (5)	1.910 (5)	97 (2)	8 (2)	0.252 (3)
CS	104.935	112.278	112.048	0.230 (3)	0.803 (3)	96 (3)	10 (3)	0.297 (5)
PSO	104.345	112.278	111.861	0.417 (4)	1.358 (4)	94 (4)	10 (3)	0.210 (2)
DE	111.053	112.278	112.196	0.082 (2)	0.189 (2)	96 (3)	10 (3)	0.276 (4)
EP	111.707	112.278	112.226	0.052 (1)	0.111 (1)	100 (1)	6 (1)	0.182 (1)

* The performance ranking is indicated by superscript numbers

6.3. Complexity

The complexity of each algorithm is determined by measuring the average CPU time required to complete each iteration. To do so, the tic/toc function provided in MATLAB was used. A high average CPU time implies that the formulation of the algorithm is long and complicated. An algorithm with low CPU time will allow a fast and low-cost hardware implementation because the MPPT sampling time can be lowered. The average CPU time taken to process each algorithm is plotted in Figure 15. As seen, the least time is EP; it consumes an average processing time of 0.182 ms. This proves that (8) is a simple and less complicated formulation in generating offspring. The highest is CS, which requires an average time of 0.297 ms.

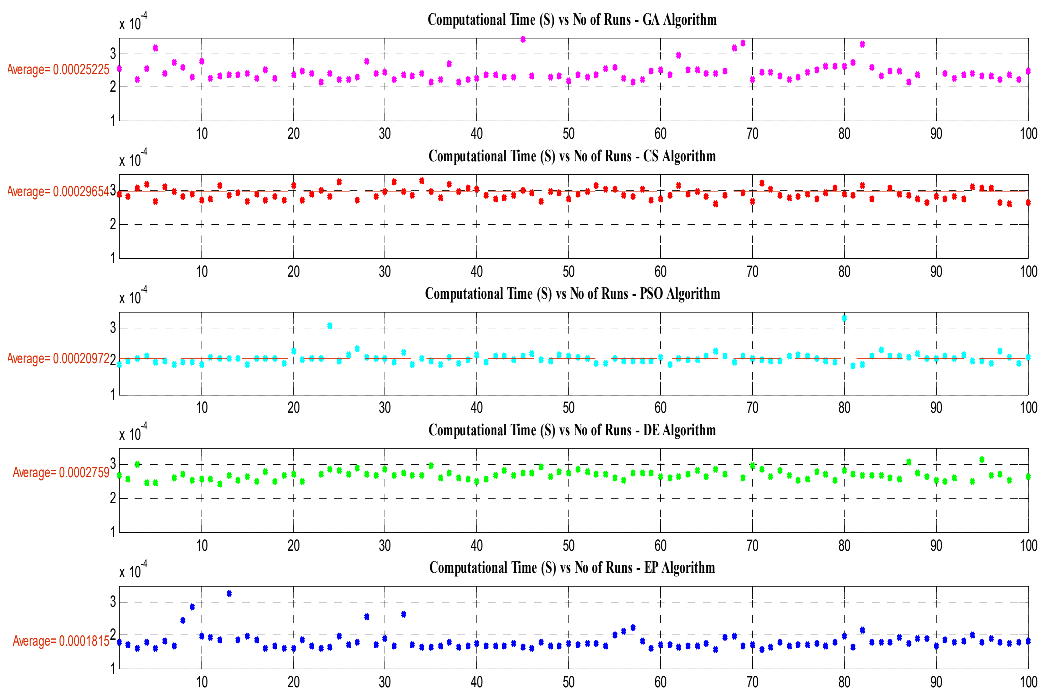

Figure 15. Distribution of CPU time under 100 runs for GA, CS, PSO, DE, and EP

6.4. Success Rate

The success rate is defined as the number of runs that successfully reach the MPP with 1 W tolerance, which represent less than a 1% ripple of the PV power. The corresponding best fitness trajectory for all algorithms over 100 runs is shown in Figure 13. The most successful algorithm to track MPPT is EP; it consistently obtained a success rate of 100%. This shows that the success rate is high when the search area is controlled to focus toward global MPP as in (8). The nearest rival, GA, exhibits 97% success, followed by CS with 96%, DE with also 96%, and PSO with 94%.

7. CONCLUSION

Critical evaluations of five different SC-based MPPT algorithms—namely, genetic algorithm (GA), cuckoo search (CS), particle swarm optimization (PSO), differential evolution (DE), and evolutionary programming (EP)—have been presented for global MPP tracking, which works in conjunction with a boost DC–DC converter. Furthermore, a standardized benchmarking procedure to critically evaluate the performance of various SC MPPT algorithms has been proposed. In general, EP appears to be the most promising and encouraging algorithm to be used in MPPT for a PV system under partial shading conditions. EP dominates and ranks first for all performance criteria study in this paper. However, the potential of other algorithms cannot be denied because their performances can be further enhanced through better parameters tuning. In addition, the reproduction step of each algorithm has a high potential for improvement using a hybrid algorithm. Also, different algorithms may work well on some optimization problems but may not work well on others. Therefore, this article will provide a set of guidelines or benchmarks on how to evaluate the performances of different algorithms fairly.

ACKNOWLEDGEMENTS

This research was supported and funded by the Institute of Research Management & Innovation (IRMI), Universiti Teknologi MARA, under the Lestari (01/03/2016) grant.

REFERENCES

[1] D. Hohm and M. E. Ropp, "Comparative study of maximum power point tracking algorithms," Progress *in photovoltaics: Research and Applications,* vol. 11, pp. 47-62, 2003.
[2] T. Esram and P. L. Chapman, "Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques," *Energy Conversion, IEEE Transactions on,* vol. 22, pp. 439-449, 2007.
[3] K. Ishaque and Z. Salam, "A review of maximum power point tracking techniques of PV system for uniform insolation and partial shading condition," *Renewable and Sustainable Energy Reviews,* vol. 19, pp. 475-488, 2013.
[4] M. Tajuddin, M. Arif, S. Ayob, and Z. Salam, "Perturbative methods for maximum power point tracking (MPPT) of photovoltaic (PV) systems: a review," *International Journal of Energy Research,* vol. 39, pp. 1153-1178, 2015.
[5] J. P. Ram, T. S. Babu, and N. Rajasekar, "A comprehensive review on solar PV maximum power point tracking techniques," *Renewable and Sustainable Energy Reviews,* vol. 67, pp. 826-847, 2017.
[6] H. Patel and V. Agarwal, "Maximum Power Point Tracking Scheme for PV Systems Operating Under Partially Shaded Conditions," *Industrial Electronics, IEEE Transactions on,* vol. 55, pp. 1689-1698, 2008.
[7] H. Taheri, Z. Salam, K. Ishaque, and Syafaruddin, "A novel Maximum Power Point tracking control of photovoltaic system under partial and rapidly fluctuating shadow conditions using Differential Evolution," in *Industrial Electronics & Applications (ISIEA), 2010 IEEE Symposium on*, 2010, pp. 82-87.
[8] K. Ishaque, Z. Salam, H. Taheri and A. Shamsudin, "Maximum Power Point Tracking for PV system under partial shading condition via particle swarm optimization," *2011 IEEE Applied Power Electronics Colloquium (IAPEC)*, Johor Bahru, 2011, pp. 5-9.
[9] Z. Salam, J. Ahmed, and B. S. Merugu, "The application of soft computing methods for MPPT of PV system: A technological and status review," *Applied Energy,* vol. 107, pp. 135-148, 2013.
[10] M. F. N. Tajuddin, S. M. Ayob, Z. Salam, and M. S. Saad, "Evolutionary based maximum power point tracking technique using differential evolution algorithm," *Energy and Buildings,* vol. 67, pp. 245-252, 2013.
[11] B. Subudhi and R. Pradhan, "A comparative study on maximum power point tracking techniques for photovoltaic power systems," *IEEE Transactions on sustainable energy,* vol. 4, pp. 89-98, 2013.
[12] J. Ahmed and Z. Salam, "A Maximum Power Point Tracking (MPPT) for PV system using Cuckoo Search with partial shading capability," *Applied Energy,* vol. 119, pp. 118-130, 2014.
[13] Z. Salam, Z. Ramli, J. Ahmed, and M. Amjad, "Partial Shading in Building Integrated PV System: Causes, Effects and Mitigating Techniques," *International Journal of Power Electronics and Drive Systems (IJPEDS),* vol. 6, pp. 712-722 2015.
[14] Z. Salam, M. Z. Ramli and J. Ahmed, "Comparative study of software and hardware solutions for partial shading in PV system," *2015 IEEE Conference on Energy Conversion (CENCON)*, Johor Bahru, 2015, pp. 480-485.
[15] Z. Salam and M. Z. Ramli, "Distributed maximum power point tracker for additional energy harvesting during partial shading of PV system," *2015 First Workshop on Smart Grid and Renewable Energy (SGRE)*, Doha, pp. 1-5, 2015.
[16] J. Ahmed and Z. Salam, "A critical evaluation on maximum power point tracking methods for partial shading in PV systems," *Renewable and Sustainable Energy Reviews,* vol. 47, pp. 933-953, 2015.
[17] R. Ali and M. S. Beg, *Applications of Soft Computing for the Web*. Springer, 2018.
[18] J. T. Alander, "On optimal population size of genetic algorithms," in *CompEuro '92 . 'Computer Systems and Software Engineering',Proceedings.*, 1992, pp. 65-70.

[19] R. L. Haupt, "Optimum population size and mutation rate for a simple real genetic algorithm that optimizes array factors," *IEEE Antennas and Propagation Society International Symposium. Transmitting Waves of Progress to the Next Millennium. 2000 Digest. Held in conjunction with: USNC/URSI National Radio Science Meeting (C*, Salt Lake City, UT, 2000, pp. 1034-1037 vol.2.
[20] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," *Urbana,* vol. 51, pp. 61801-2996, 1991.
[21] P. J. Hancock, "An empirical comparison of selection methods in evolutionary algorithms," in *Evolutionary Computing*: Springer, 1994, pp. 80-94.
[22] M. Z. Ali, N. H. Awad, P. N. Suganthan, A. M. Shatnawi, and R. G. Reynolds, "An improved class of real-coded Genetic Algorithms for numerical optimization," *Neurocomputing,* vol. 275, pp. 155-166, 2018.
[23] M. Sangdani, A. Tavakolpour-Saleh, and A. Lotfavar, "Genetic algorithm-based optimal computed torque control of a vision-based tracker robot: Simulation and experiment," *Engineering Applications of Artificial Intelligence,* vol. 67, pp. 24-38, 2018.
[24] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A new genetic algorithm for solving optimization problems," *Engineering Applications of Artificial Intelligence,* vol. 27, pp. 57-69, 2014.
[25] K. Ishaque and Z. Salam, "A deterministic particle swarm optimization maximum power point tracker for photovoltaic system under partial shading condition," *IEEE transactions on industrial electronics,* vol. 60, pp. 3195-3206, 2013.
[26] Y. Chen, L. Li, J. Xiao, Y. Yang, J. Liang, and T. Li, "Particle swarm optimizer with crossover operation," *Engineering Applications of Artificial Intelligence,* vol. 70, pp. 159-169, 2018.
[27] K. Ishaque, Z. Salam, M. Amjad, and S. Mekhilef, "An improved particle swarm optimization (PSO)–based MPPT for PV with reduced steady-state oscillation," *IEEE transactions on Power Electronics,* vol. 27, pp. 3627-3638, 2012.
[28] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*: Springer, 2011, pp. 760-766.
[29] M. Miyatake, F. Toriumi, T. Endo, and N. Fujii, "A Novel maximum power point tracker controlling several converters connected to photovoltaic arrays with particle swarm optimization technique," *in Power Electronics and Applications, 2007 European Conference on*, 2007, pp. 1-10.
[30] M. A. Ramli, K. Ishaque, F. Jawaid, Y. A. Al-Turki, and Z. Salam, "A modified differential evolution based maximum power point tracker for photovoltaic system under partial shading condition," *Energy and Buildings,* vol. 103, pp. 175-184, 2015.
[31] X. Yu, X. Yu, Y. Lu, G. G. Yen, and M. Cai, "Differential evolution mutation operators for constrained multi-objective optimization," *Applied Soft Computing,* 2018.
[32] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 2009, pp. 210-214.
[33] X. Li and M. Yin, "Modified cuckoo search algorithm with self adaptive parameter method," *Information Sciences,* vol. 298, pp. 80-97, 2015.
[34] J. Ahmed and Z. Salam, "A soft computing MPPT for PV system based on Cuckoo Search algorithm," in *Power Engineering, Energy and Electrical Drives (POWERENG), 2013 Fourth International Conference on*, 2013, pp. 558-562.
[35] D. B. Fogel, *System identification through simulated evolution: A machine learning approach to modeling*. Ginn Press, 1991.
[36] D. B. Fogel, L. J. Fogel, and J. W. Atmar, "Meta-evolutionary programming," in *Signals, Systems and Computers, 1991. 1991 Conference Record of the Twenty-Fifth Asilomar Conference on*, vol.1, pp. 540-545, 1991.
[37] Y. W. Yang, J. F. Xu, and C. K. Soh, "An evolutionary programming algorithm for continuous global optimization," *European journal of operational research,* vol. 168, pp. 354-369, 2006.
[38] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary computation,* vol. 3, pp. 82-102, 1999.
[39] N. Hashim, Z. Salam, and S. Ayob, "Maximum power point tracking for stand-alone photovoltaic system using evolutionary programming," in *Power Engineering and Optimization Conference (PEOCO), 2014 IEEE 8th International*, 2014, pp. 7-12.
[40] K. Ishaque, Z. Salam, H. Taheri, and Syafaruddin, "Modeling and simulation of photovoltaic (PV) system during partial shading based on a two-diode model," *Simulation Modelling Practice and Theory,* vol. 19, pp. 1613-1626, 2011.
[41] K. Ishaque, Z. Salam, and H. Taheri, "Simple, fast and accurate two-diode model for photovoltaic modules," *Solar Energy Materials and Solar Cells,* vol. 95, pp. 586-594, 2011.
[42] V. J. Chin, Z. Salam, and K. Ishaque, "An accurate modelling of the two-diode model of PV module using a hybrid solution based on differential evolution," *Energy Conversion and Management,* vol. 124, pp. 42-50, 2016.
[43] K. Ishaque, Z. Salam, and Syafaruddin, "A comprehensive MATLAB Simulink PV system simulator with partial shading capability based on two-diode model," *Solar Energy,* vol. 85, pp. 2217-2227, 2011.
[44] N. Hashim, Z. Salam, D. Johari, and N. F. M. N. Ismail, "DC-DC boost converter design for fast and accurate MPPT algorithms in stand-alone photovoltaic system," *International Journal of Power Electronics and Drive Systems (IJPEDS),* vol. 9, pp. 1038-1050, 2018.