Available online at www.sciencedirect.com
ScienceDirect
Solar Energy 85 (2011) 1768–1779
SOLAR ENERGY
www.elsevier.com/locate/solener

A critical evaluation of EA computational methods for Photovoltaic cell parameter extraction based on two diode model

Kashif Ishaque, Zainal Salam *, Hamed Taheri, Amir Shamsudin

Faculty of Electrical Engineering, Universiti Teknologi Malaysia, UTM 81310, Skudai, Johor Bahru, Malaysia

Received 18 January 2011; received in revised form 10 March 2011; accepted 16 April 2011
Available online 20 May 2011

Communicated by: Associate Editor Takhir Razykov

Abstract

Due to its ability to handle nonlinear functions regardless of the derivatives information, evolutionary algorithms (EA) are envisaged to be very effective for extracting parameter of photovoltaic (PV) cell. This paper presents critical evaluation of the parameters extraction of two diode PV model using three EA methods, namely Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Differential Evolution (DE). For DE, two variations are proposed: (1) boundary based differential evolution (B-DE) and (2) penalty based differential evolution (P-DE). The performance of each method is evaluated based on several factors: accuracy and consistency of solution; speed of convergence; computational efficiency and the required number of control parameters. Comparisons are carried out using synthetic data and are validated by six PV modules of different types (multi-crystalline, mono-crystalline, and thin-film) from various manufacturers. Information derived from these critical evaluations can be useful to determine the best computational method to build an efficient and accurate PV system simulator.

Keywords: Differential evolution; Genetic algorithm; Particle swarm optimization; Parameter extraction; Penalty function; Photovoltaic energy

1. Introduction

Due to the high initial cost of a photovoltaic (PV) power system, optimal capturing of the available solar energy has to be ensured. The energy yield is typically predicted using simulation – carried out during the design stage. For a PV system simulator, the choice of a model that closely emulates the characteristics of PV cells is very crucial. Over the years, several models are introduced. Among these, the more popular ones are the circuit-based single diode model with R_s (Walker, 2001) and R_p (Villalva et al., 2009) and the two diode model (Chih-Tang et al., 1957). The latter, despite requiring more computation efforts, are preferable as its current–voltage (I–V) characteristics resemblance closer to the output of a physical cell (Chih-Tang et al., 1957). To achieve accurate simulation, the values of the circuit parameters that model the cell, namely the PV current (I_{PV}), diode saturation currents (I_{o1}, I_{o2}), diode ideality factors (a_1, a_2), series resistance (R_s) and shunt resistance (R_p) have to be correctly determined.

Generally, two types of approach are used to obtain the above-mentioned parameters: (1) the analytical (Charles et al., 1985; Chan and Phang, 1987; De Soto et al., 2006; Ortiz-Conde et al., 2006) and (2) numerical extraction (Gottschalg et al., 1999; Nakanishi et al., 2000; Jervase et al., 2001; Chegaar et al., 2004; Haouari-Merbah et al., 2005; Chegaar et al., 2006; Duffie and Beckman 2006; Liu et al., 2008) techniques. The analytical technique requires information on several key points of the I–V characteristic curve, i.e. the current and voltage at the maximum power point (MPP), short-circuit current (I_{sc}), open-circuit voltage (V_{oc}), and slopes of the current–voltage

* Corresponding author. Tel.: +60 75536187; fax: +60 75566272.
E-mail addresses: kashif@fkegraduate.utm.my (K. Ishaque), zainals@fke.utm.my (Z. Salam).

0038-092X/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.solener.2011.04.015

characteristic at the axis intersections. Although simple, its accuracy is strongly dependent on the selected points on the I–V curve. If one or more of these values are incorrectly specified, the errors for the computed parameters can be very significant. Alternatively, the numerical extraction technique is based on certain mathematical algorithm to fit all the points on the I–V curve. The advantage of this approach is clear; more accurate model parameters can be obtained because all points on the I–V curve are utilized. The shortcoming, however, is that the fitting requires extensive computation. Furthermore the accuracy depends on the type of fitting algorithm, the cost function and the initial values of the parameters to be extracted. It was reported that inappropriate selection of initial values may lead non-convergence of the algorithm (Gottschalg et al., 1999; Jervase et al., 2001).

Evolutionary algorithms (EA) techniques have gained much attention due to its ability to handle nonlinear functions without requiring derivatives information. EA is a stochastic optimization method that appears to be very efficient in optimizing real-valued multi-modal objective functions. Recently, various EA algorithms such as genetic algorithm (GA) (Jervase et al., 2001; Moldovan et al., 2009), particle swarm optimization (PSO) (Ye et al., 2009) and differential evolution (DE) (da Costa et al., 2010) are employed for parameter extraction of the various PV cell models. Although, generally, EA is considered to be more promising than other computational methods, there appears to a gap in the literature for an in-depth to study on the comparative performance of different types of EA algorithms when applied to PV cell model extraction. In our view, such evaluation is a fundamental requirement to build an efficient and accurate PV simulator. Hence this work is carried out.

To objectively evaluate the performance of EA methods, several criteria are introduced: (1) accuracy of solution, (2) consistency of solution, (3) speed of convergence, (4) computational efficiency, and (5) the required number of control parameters. The preferred EA method will be judged on its ability to optimize on these issues.

This paper evaluates three most popular EA methods, namely GA, PSO and DE. The first two methods, i.e. GA and PSO are mainly derived from the work published in Jervase et al. (2001) and Ye et al. (2009), respectively. For DE, only one paper can be found in literature (da Costa et al., 2010). Even then, it was unclear whether the DE algorithm proposed in (da Costa et al., 2010) converges to the global minima. Furthermore, the work does not involve any penalizing approach to ensure the solution lies in feasible region, prompting questions on the validity of the published results. Because of these shortcomings, two modified version of DE is introduced in this work. First is the boundary based DE (B-DE), which is accomplished by utilizing the boundary conditions in the conventional DE method (Storn and Price, 1997). The boundary conditions are applied to force the PV parameters to be in a certain feasible region. The second is called the penalty based DE (P-DE). This method is developed by employing a penalty function in the conventional DE (Storn and Price, 1997). The introduction of penalty function ensures the availability of the model parameters in the feasible region. All the four methods are evaluated using a synthetic I–V data (Ye et al., 2009). Furthermore, the best EA method is validated with six PV modules of different types (multi-crystalline, mono-crystalline, and thin-film) from various manufacturers. Finally a table is proposed to compare the performance of each method; although the table may not be appropriate for benchmarking, it can be used as a guideline to indicate the best EA method to extract the parameters of a two diode PV cell model.

2. Two diode model of PV Cell

The two diode model that becomes the basis of evaluation is depicted in Fig. 1. The output current of the module is well known and can be described as:

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

where I_{PV} is the current generated by the incidence of light; I_{o1} and I_{o2} are the reverse saturation currents of diode 1 and diode 2, respectively. The I_{o2} term is introduced to compensate for the recombination loss in the depletion region as described in Chih-Tang et al. (1957). Other variables are defined as follows: V_{T1} and V_{T2} (both equal to $N_s kT/q$) are the thermal voltages of the PV module having N_s cells connected in series, q is the electron charge ($1.60217646 \times 10^{-19}$ C), k is the Boltzmann constant ($1.3806503 \times 10^{-23}$ J/K) and T is the temperature of the p–n junction in Kelvin. Variables a_1 and a_2 represent the diode ideality constants; a_1 and a_2 represent the diffusion and recombination current component, respectively. Although greater accuracy can be achieved using this model than the single diode model (Chih-Tang et al., 1957), it requires the computation of seven parameters, namely I_{PV}, I_{o1}, I_{o2}, R_p, R_s, a_1 and a_2. Note that Eq. (1) is a transcendental equation which cannot be solved by a direct analysis.

3. Parameter extraction of PV Cell

3.1. Methodology

The cell parameters are extracted as follows. Given a set of I–V data of a particular cell, the curve fitting process

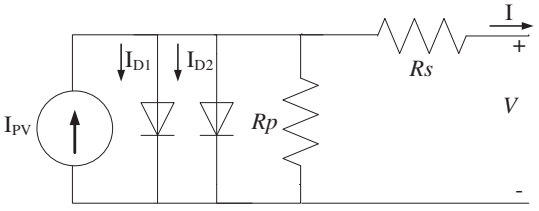

Fig. 1. Two diode model of PV cell.

using GA, PSO, B-DE and P-DE is applied to update the values of model parameters in Eq. (1). To apply the EA methods, Eq. (1) has to be re-written as:

$$y(I,V,\Phi) = I_{PV} - I_{01}\left[\exp\left(\frac{V+IR_S}{a_1V_{T1}}\right)-1\right] - I_{02}\left[\exp\left(\frac{V+IR_s}{a_2V_{T2}}\right)-1\right] - \left(\frac{V+IR_s}{R_p}\right) - I \quad (2)$$

where $\Phi = [I_{PV}, I_{o1}, I_{o2}, R_s, R_p, a_1, a_2]$ are the parameters set to be extracted. After each iteration, the values in Φ is updated and then checked if it has converge to the correct value. The procedure ends when intended maximum number of iteration has been reached.

3.2. Objective function

To objectively evaluate the performance of each EA method, an objective function J is introduced. In this work, the objective function can be given by:

$$J = \sqrt{\frac{1}{N}\sum_{m=1}^{N} y(I_m, V_m, \Phi)^2} \quad (3)$$

where I_m and V_m are the data pair of I–V characteristics curve, respectively; N is the number of data. Hence, the extraction procedure is based on the minimization of J in conjunction to Φ. A smaller value of J implies the least deviation between the I and V data and the one computed by the EA method. Ideally, a zero value J is desired. It can be seen from Eq. (3) that J is a nonlinear function with no apparent quadratic function. This leads to several local minima with one global minimum. Conventional iterative methods that attempt to solve this problem require the gradient information. However, it is not easy to estimate the gradient when there are unobservable states with discontinuous types of nonlinearity exist in the model (Ye et al., 2009). Note that throughout this paper, we shall use the terms "objective function value" and "fitness value" interchangeably. But, due to minimization problem, a lower objective function value will correspond to higher fitness.

4. Brief review of EA techniques

4.1. Genetic algorithm

The genetic algorithm (GA) is based on the theory of biological evolution (Holland, 1975). Fig. 2 shows the flow chart of GA. The common operators used in GAs are described as follows:

4.1.1. Selection

This procedure selects the chromosomes that contribute in the reproduction process to give birth to the next generation. Only the best chromosomes are considered for the next generation. The selection process can be realized by various techniques, including the elitist model, the ranking model, the roulette wheel procedure (Haupt and Haupt, 2004), etc.

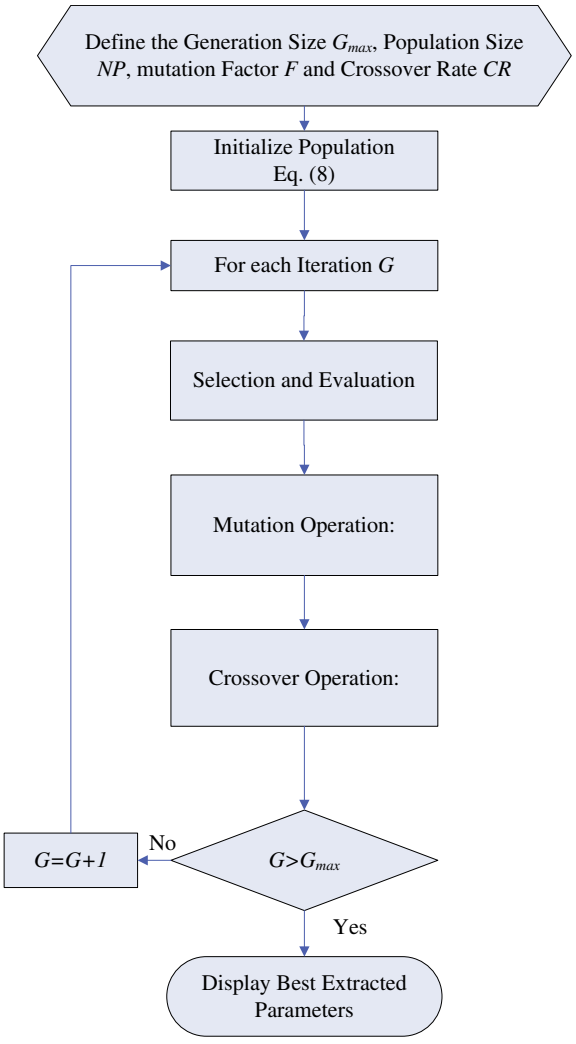

Fig. 2. Flow chart of GA.

4.1.2. Mutation

It introduces changes in some genes (parameters) of a chromosome in a population. This procedure is performed by GAs to explore new solutions. Random mutations alter a small percentage of the population (mutation rate) except for the best chromosomes. A mutation rate between 1% and 20% often works well. If the mutation rate is above 20%, too many good parameters can be mutated, causing the algorithm to stall. Note that the new value of each parameter should be in the $[X_{iL}, X_{iH}]$ corresponding interval. Consequently, after paring, mutated parameters are engaged to ensure that the parameters space is explored in new regions.

4.1.3. Crossover

This process uses two selected chromosomes from a current generation (parents) and crosses them with some probability (the "crossover probability" or "crossover rate") to

obtain two individuals for the new generation (offspring's). There are several types of crossover, but the simplest method is to arbitrarily choose one or more points (parameters) in the chromosome of each parent to mark as crossover points. Then the parameters between these points are merely swapped between the two parents.

Although GA has been used extensively in many applications, numerous computational difficulties, namely premature convergence, low speed, and degradation for highly interactive fitness function are reported (Zwe-Lee, 2004; Ji et al., 2006).

4.2. Particle swarm optimization (PSO)

Particle swarm optimization (PSO) is a stochastic, population-based search method, modeled after the behavior of bird flocks (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995). A PSO algorithm maintains a swarm of individuals (called particles), where each particle represents a candidate solution. Particles follow a simple behavior: emulate the success of neighboring particles, and achieve their own successes. The position of a particle is therefore influenced by the best particle in a neighborhood, as well as the best solution found by the particle. Particle position, x_i, are adjusted using:

$$x_i(j+1) = x_i(j) + v_i(j+1) \tag{4}$$

where the velocity component, v_i, represents the step size. The velocity is calculated by:

$$v_i(j+1) = wv_i(j) + c_1 r_1\{y_i(t) - x_i(t)\} + c_2 r_2\{\hat{y}_i(t) - x_i(t)\} \tag{5}$$

where w is the inertia weight, c_1 and c_2 are the acceleration coefficients, r_1, $r_2 \in U(0, 1)$, y_i is the personal best position of particle i, and $\hat{y}_i$ is the neighborhood best position of particle i. The inertia weight w plays an important role in balancing the global search and local search. A large w facilitates a global search while a small inertia weight facilitates a local search. It can be a positive constant or a positive decreasing linear function of iteration index j. In (Ye et al., 2009), the inertia weight was used from the following decreasing linear function:

$$w(j) = w_{\max} - (w_{\max} - w_{\min})\frac{j}{G_{\max}} \tag{6}$$

where $w_{\max}$ and $w_{\min}$ are the final weight and the initial weight, respectively, and $G_{\max}$ is the maximum iteration number. The velocity Eq. (2) is further updated by following law:

$$v(j+1) = \begin{cases} V_{\max} & \text{if } v(j+1) > V_{\max} \\ V_{\min} & \text{if } v(j+1) < -V_{\min} \\ v(j+1) & \text{otherwise} \end{cases} \tag{7}$$

where $V_{\max}$ is a constant that is set to clamp the unnecessary roaming of particles. Usually, the choice of $V_{\max}$ equals to the maximum allowable departure of any particle in that dimension (Shi and Eberhart, 1998). Fig. 3 depicts the flow chart of the PSO.

Due to excessive roaming of particles, long convergence time and large number of iterations are experienced (Ye et al., 2009). Furthermore, penalizing unnecessary movement of particles can affect the convergence performance. For instance, in a PSO based PV cell parameters extraction (Ye et al., 2009), the authors have used a common approach to penalize the velocity of the particles with a factor V_{max}. The global exploration ability of a PSO strongly depends on this factor. If $V_{\max}$ is too large, particles may fly past good solutions. Alternatively, a small value of $V_{\max}$ will strict the particles to fly beyond locally good solutions. Hence, the choice of $V_{\max}$ may be not be consistent for different types of PV modules.

4.3. Boundary based differential evolution (B-DE)

Differential evolution (DE) was first introduced in (Storn and Price, 1997). It has three main advantages: (1)

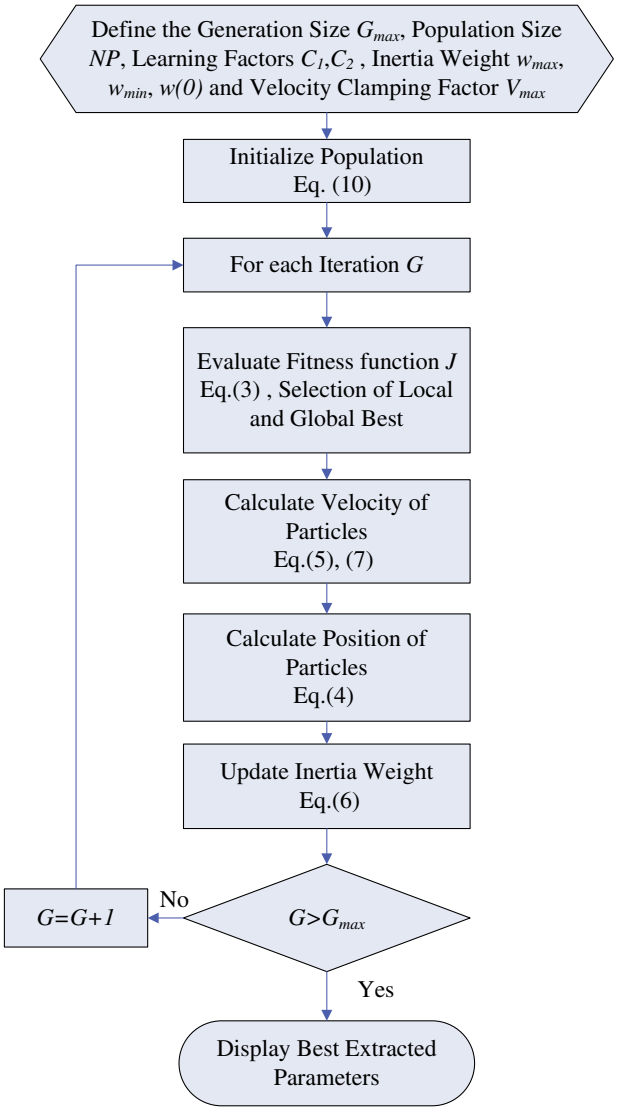

Fig. 3. Flow chart of PSO.

able to locate the accurate global optimum regardless of the initial parameter values (2) rapid convergence (3) utilizing few control parameters. The optimization procedure is similar to GA, but unlike GA, which relies on crossover, DE primarily utilizes mutation operation (i.e. difference vector) as a search mechanism and selection operation to direct the search toward the prospective regions in the search space. Like other EA family, DE relies on initial random population generation, which is then improved using selection, mutation, and crossover repeated through generations until the stopping condition is reached – usually a satisfactory good fitness value or a predefined maximum number of generations $G_{\max}$.

Like the other evolutionary algorithms, DE also works on a population, P_G, of candidate solutions. These candidate solutions are known as the individuals of the population. In particular, DE creates a population of NP D dimensional real-valued parameter vectors $X_{i,G}$ as:

$$P_{X,G} = (X_{i,G}) \quad i = 1, 2, \ldots, NP, \quad G = 0, 1, \ldots, G_{\max} \tag{8}$$

$$X_{i,G} = (X_{j,i,G}) \quad j = 1, 2, \ldots, D \tag{9}$$

The index, $G = 0, 1, \ldots, G_{\max}$, indicates the generation to which a vector belongs. Additionally, each vector has a population index, i, from 1 to Np. Parameters within vectors are indexed with j, from 1 to D. Fig. 4 shows the flow chart of DE process. It involves following stages:

4.3.1. Initialization

In order to begin the optimization process, an initial population of NP D dimensional real-valued parameter vectors $X_{i,G} = [X_{1,i,G}, X_{2,i,G}, \ldots, X_{j,i,G}, \ldots, X_{D,i,G}]$ is created. Each vector forms a candidate solution to the multidimensional optimization problem. Initial parameter values are usually randomly selected uniformly in the interval $[X_L, X_H]$,where $X_L = [X_{1,L}, X_{2,L}, \ldots, X_{D,L}]$ and $X_H = [X_{1,H}, X_{2,H}, \ldots, X_{D,H}]$ are the lower and upper bound of the search space, respectively.

$$X_{j,i,0} = X_L + rand[0, 1](X_H - X_L) \tag{10}$$

Fig. 5a explains the initialization procedure of DE. For simplicity, a 2-dimensional parameter space is shown here.

4.3.2. Mutation

Mutation is a perturbation or change with a random element. In DE-literature, a parent vector from the current generation is known as *target* vector, a mutant vector achieved through the differential mutation operation is called a *donor* vector and finally an offspring formed by recombining the donor with the target vector is called *trial* vector. For a given parameter vector $X_{i,G}$, three vectors $(X_{r1,G}, X_{r2,G}, X_{r3,G})$ are randomly selected in the range [1, NP], such that the indices i, r_1, r_2 and r_3 are distinct. A donor vector $V_{i,G}$ is created by adding the weighted difference between the two vectors to the third (base) vector as:

$$V_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) \tag{11}$$

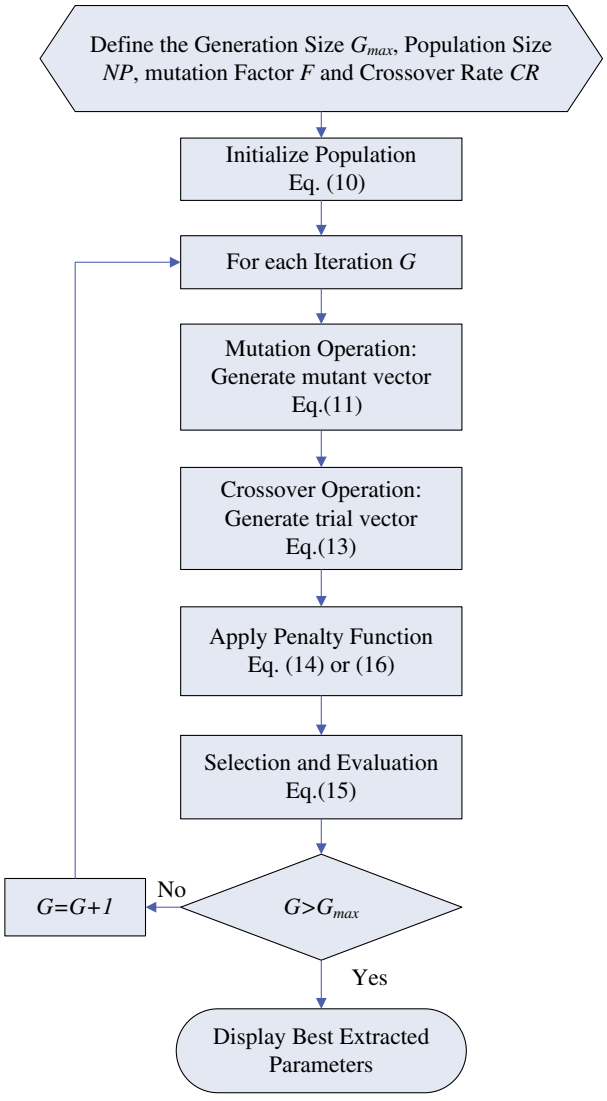

Fig. 4. Flowchart of DE.

where F is a mutation scaling factor, which is typically chosen from the range [0, 1]. Fig. 5b illustrates the process on a 2-D parameter space (showing constant cost contours of an arbitrary objective function).

4.3.3. Crossover

The donor vector $V_{i,G+1}$ and the target vector $X_{i,G}$ are mixed to yield the trial vector:

$$U_{i,G} = [U_{1i,G}, U_{2i,G}, \ldots.., U_{ji,G}, \ldots.., U_{Di,G}] \tag{12}$$

In DE algorithm, two kinds of crossover methods are used i.e. exponential and binomial (or uniform) (Price et al., 2005). In this work, binomial crossover strategy is used which can be described as:

$$U_{j,i,G} = \begin{cases} V_{j,i,G}, & \text{if } (rand \leqslant CR \text{ or } j = j_{rand}) \\ X_{j,i,G}, & \text{otherwise} \end{cases} \tag{13}$$

where CR is known as the crossover rate and appears as another control parameter of DE just like F. $j_{rand} \in [1, 2, \ldots, D]$

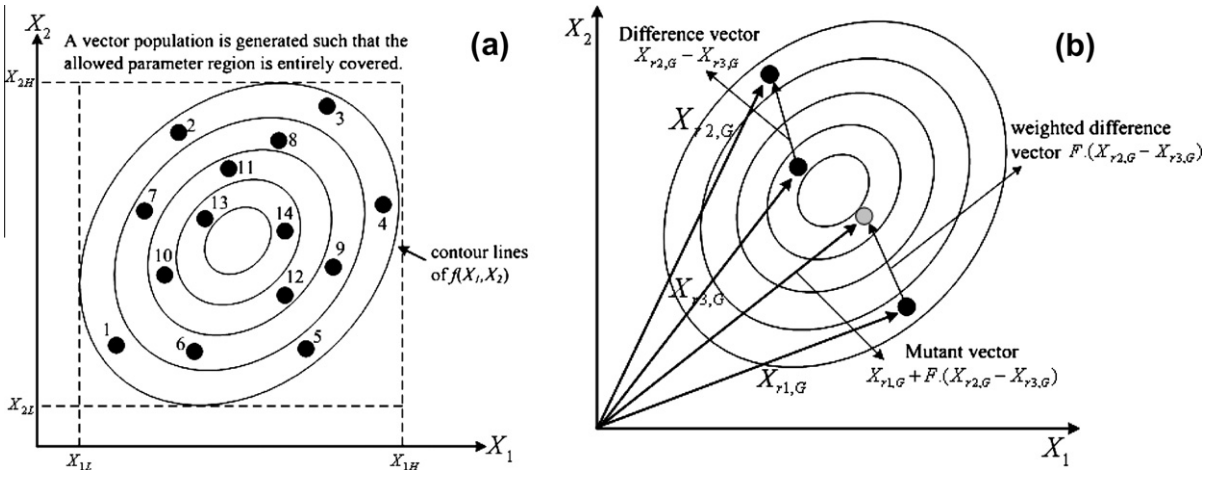

Fig. 5. (a) Initialization of DE (b) mutation process of DE.

is a randomly chosen index, which ensures that $U_{i,G}$ attains at least one element from $V_{i,G}$. Here, D is the number of parameters to be optimized.

A major problem in the conventional DE (Storn and Price, 1997) is the nonphysical values of the extracted parameters (Michalewicz and Schoenauer, 1996). To avoid such condition, the boundary differential evolution (B-DE) is proposed in this work. The boundary conditions will force the particles to remain in the search ranges as specified by the user. Fig. 6a describes the boundary conditions of the B-DE method. It can be seen that a number of the parameters (solutions) are lying on the boundary limits of the feasible region. Here, a common problem exists for selecting the suitable search ranges. If the solution is outside the search ranges, accurate results cannot be ensure. Therefore, a broader search range can easily solve this problem. Considering this fact, a very large search range (i.e. ±100 of synthetic values) is used in this work.

The boundary conditions are as follows:

$$U_{i,G}=\begin{cases} X_{iH} & \text{if } U_{i,G}>X_{iH} \\ X_{iL} & \text{if } U_{i,G}<X_{iL} \end{cases} \tag{14}$$

4.3.4. Evaluation and selection

The selection operation at $G=G+1$ is described:

$$X_{i,G+1}=\begin{cases} U_{i,G} & \text{if } J(U_{i,G})<J(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \tag{15}$$

where $J(X)$ is the objective function to be minimized. Thus, if the new trial vector acquires a lower value of the objective function, it swaps the corresponding target vector in the next generation; otherwise the target is preserved in the population. Hence, the population either gets better or remains the same in fitness status, but never declines.

4.4. Penalty based differential evolution (P-DE)

Due to the basic structure of DE (which combines three randomly selected individuals), the DE process is likely to produce solutions outside their allowed range. In the preceding section, the boundary conditions were employed to obtain the solution in the feasible regions. However, setting the parameter violations to the boundary conditions leads to a degradation of the solution's diversity. To avoid

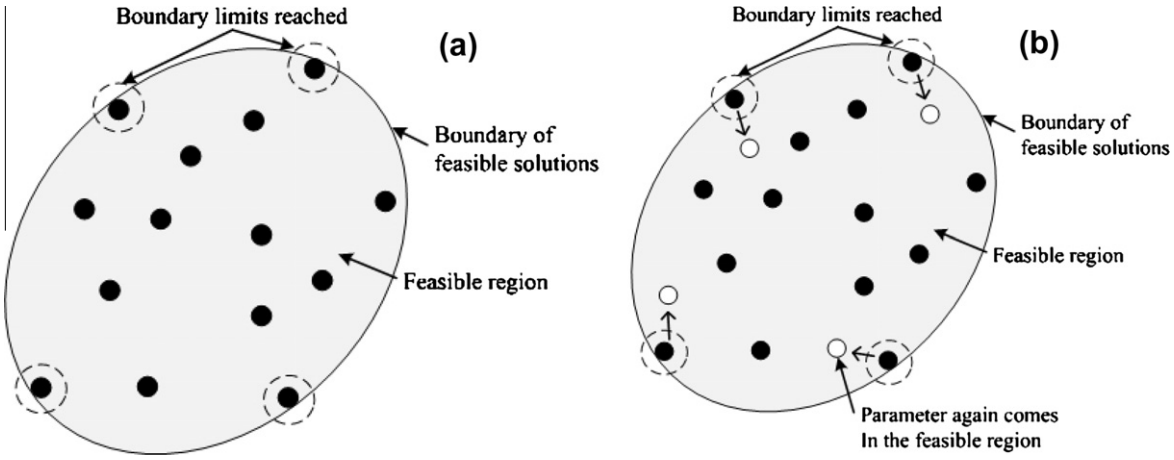

Fig. 6. (a) Effect of boundary conditions and (b) effect of penalty function.

the solution from being stuck at the boundary, a penalty function is proposed. It ensures the parameter values lie within the allowable range after recombination. Any parameter that violates the limits is replaced with random values using:

$$U_{i,G+1} = \begin{cases} U_{i,G+1} - rand[0,1](X_{iH} - X_{iL}) & \text{if } U_{i,G+1} > X_{iH} \\ U_{i,G+1} + rand[0,1](X_{iH} - X_{iL}) & \text{if } U_{i,G+1} < X_{iL} \end{cases} \quad (16)$$

Fig. 6b shows the effect of penalty function. In the B-DE, parameters (solution) remain at the boundaries of the feasible solution region. While in P-DE, due to the penalizing phenomena, parameters are again shifted to the feasible region.

Fig. 4 shows the flow chart of the proposed P-DE. This flowchart also can be used for the B-DE implementation. The only difference between P-DE and B-DE is the use of penalty function.

The pseudo-code for the proposed DE method is shown in Fig. 7.

5. Results with synthetic data

The evaluation of all four EA methods for parameter extraction is carried out using synthetic data obtained from (Ishaque et al., 2011). The synthetic data comparison approach is also utilized by other researchers, for example (Ye et al., 2009). In this work, the synthetic data were calculated using the two diode model with the following seven parameters: $I_{PV} = 8.21$ A, $I_{o1} = I_{o2} = 4.218 \times 10^{-10}$, $R_s = 0.32\ \Omega$, $R_p = 160.5\ \Omega$, $a_1 = 1$ and $a_2 = 1.2$. The temperature and irradiance are at the Standard Test Conditions (STC) i.e. 1000 W/m^2 and 298 K, respectively. Using these values, the synthetic $I-V$ curve is plotted. Then the $I-V$ curve fitting using GA, PSO, B-DE and P-DE are performed. The extracted parameters obtained for these two methods are evaluated against the synthetic data.

To ensure a fair evaluation for the four methods, similar simulation conditions, i.e. population size, maximum iteration number and search ranges are maintained. The population size (NP), is chosen to be 70. This is a reasonable choice; typical values of NP ranged between $5D$ and $10D$. The maximum generation number ($G_{\max}$) is set to 40,000. Even though, both B-DE and P-DE can converge within 10,000 iterations, this value is selected to be consistent with (Ye et al., 2009). The mutation scaling factor (F) is set at 0.8. There is no strict rule on the selection of F but in most cases, $F > 0.4$ (Gämperle et al., 2002). The crossover rate (CR) is chosen to be 1. Large value of CR intensifies the diversity of population, thus improving the convergence speed (Gämperle et al., 2002). Moreover, a high value of CR is desirable as the parameters in the model are highly correlated (Taheri et al., 2010). The search ranges are set to ±100% of the PV parameter values specified above.

The DE/best/1/bin strategy is used for the B−DE and P−DE. In this nomenclature, the word "best" defines the best vector from the current population, "1" specifies number of difference vector and "bin" describes the binomial crossover technique. For the GA implementation, a crossover rate, $P_c = 0.8$ and mutation rate, $P_m = 0.2$, is used. Furthermore, A GA function named as "*ga*" in MATLAB is utilized. In the PSO implementation, following controlling variables are used: learning factors $c_1 = c_2 = 2$, inertia factors $w_{\max} = 0.9$, $w_{\min} = 0.4$, and velocity clamping factor $V_{\max} = 0.5$ (Shi and Eberhart 1998). Each EA method is executed for 100 times and performance of each method is compared using following criteria:

Step 1
Setting values of the control parameters of DE:
population size NP, crossover rate CR, mutation factor F and the I–V data of PV module.
Step 2
Set the generation number $G = 0$
and randomly initialize a population of NP individuals with $X_{i,G} = [X_{1,i,G}, X_{2,i,G}, X_{3,i,G}, ..., X_{D,i,G}]$ and each individual uniformly distributed in the range $[X_L, X_H]$ as:
$X_{j,I,0} = X_L + rand\ [0,1](X_H - X_L)$ where
$X_L = [X_{1,L}, X_{2,L}, ..., X_{D,L}]$ and
$X_H = [X_{1,H}, X_{2,H}, ..., X_{D,H}]$ with $i = [1, 2, ...,NP]$.
Step 3
WHILE the stopping criterion is not satisfied
DO
FOR i = 1 to NP
Step 3.1 Mutation Step
Generate a donor vector $V_{i,G} = [V_{1,i,G}, V_{2,i,G}, ..., V_{D,i,G}]$ corresponding to the i_{th} target vector $X_{i,G}$ using the differential mutation scheme of DE as:
$V_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})$
Step 3.2 Crossover Step
Generate a trial vector $U_{i,G} = [U_{1,i,G}, U_{1,i,G}, ..., U_{D,i,G}]$ for the i_{th} target vector $X_{i,G}$ through binomial crossover in the following way:
$U_{j,i,G} = V_{j,i,G}$, if $(rand_{i,j}[0, 1] \le CR$ or $j = j_{rand})$
$X_{j,i,G}$, otherwise,
Step 3.2.1 Penalty Function
IF $U_{j,i,G} < X_L$, THEN $U_{j,i,G} = X_L - rand\ [0,1](X_H - X_L)$
ELSEIF $U_{j,i,G} > X_H$, THEN $U_{j,i,G} = X_H - rand\ [0,1](X_H - X_L)$
END IF
Step 3.3 Selection and Evaluation Step
Evaluate the trial vector $U_{i,G}$
IF $J(U_{i,G}) < J(X_{i,G})$, THEN $X_{i,G+1} = U_{i,G}$
ELSE $X_{i,G+1} = X_{i,G}$.
END IF
END FOR
Step 3.4 Increase the Generation Count
$G = G + 1$
END WHILE

Fig. 7. Pseudo-code for the Proposed DE Algorithm.

5.1. Accuracy of solution

Fig. 8a illustrates the convergence performance of all four methods. Note that, this is the best performance results of all four methods for 100 simulation runs. As expected, B-DE and P-DE outperform the other two methods. They exhibit very high fitness value J, hence offers very high accuracy. The accuracy of PSO is acceptable; fitness value J is about three orders of magnitude lower than DE. The worst is GA. It fails to reach to synthetic global minima.

To further investigate the accuracy of the extracted parameters, two types of error graphs are shown: Fig. 9a is the error between the best values and synthetic values (Ishaque, Salam et al.); Fig. 9b is the error between the mean value and the synthetic values. Clearly, P-DE has the smallest errors for both the best and mean values. The best values for P-DE and B-DE are same; which implies that B-DE is able to locate the global minima (synthetic values) correctly. However, by observing Fig. 9b, the accuracy of B-DE cannot be guaranteed, as considerably larger errors show the mean value is far from the synthetic value. For PSO, mean values are too far from the synthetic values, which reveals that PSO is a less preferable choice. This might be due to excessive roaming of particles in PSO. Table 1 summarizes the best extracted parameters obtained by all EA methods in these extensive simulations.

5.2. Consistency of solution

Consistency means how certain we can be of achieving a high quality solution. Due to the random nature, EA method is not guaranteed to yield the same solution at each run. This poses a great problem in determining whether the solution obtained through EA method is "truly" of good quality or whether there is a chance that a better quality

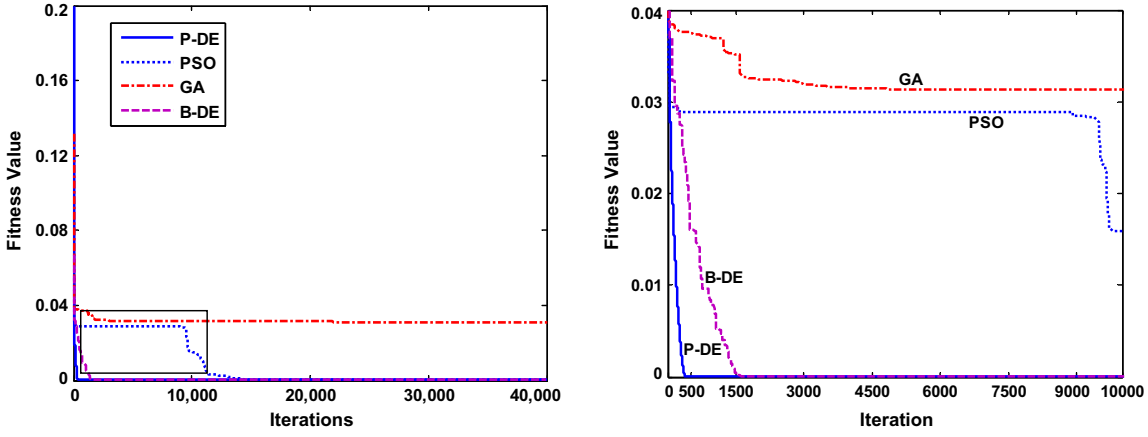

Fig. 8. Convergence performance of the fitness function with DE and GA.

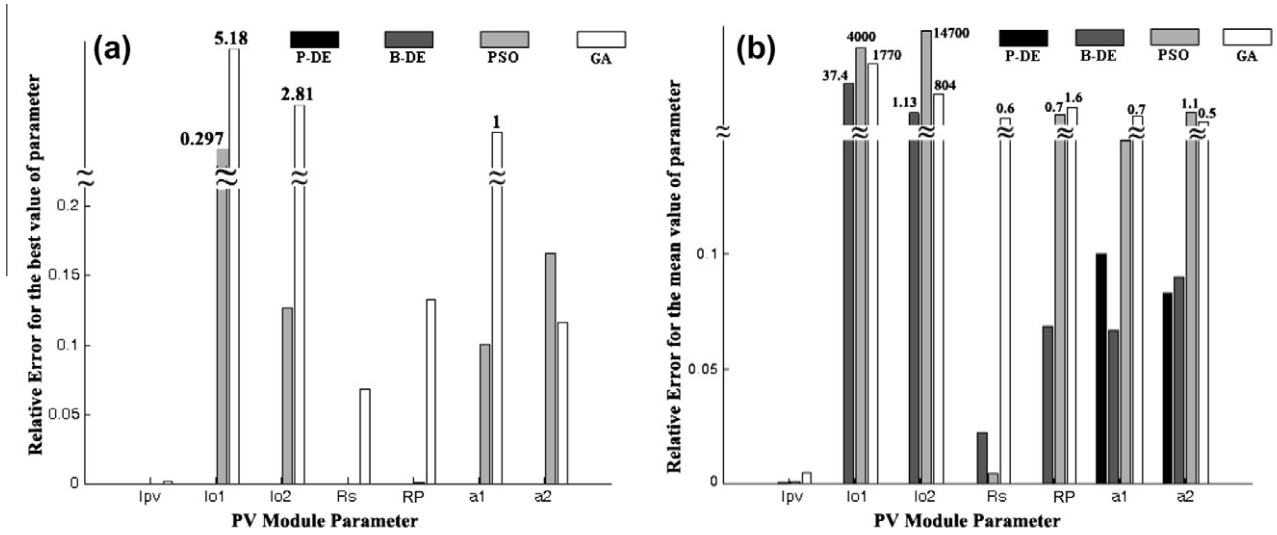

Fig. 9. Parameter comparison of all four methods, (a) relative error for the best value of extracted parameters and (b) relative error for the mean value of extracted parameters.

Table 1
Comparison of extracted parameter using various methods (best result for 100 runs).

Parameters	Synthetic values [27]	P-DE	B-DE	PSO	GA
I_{PV}	3.8 A	3.8 A	3.8 A	3.8 A	3.795 A
I_{o1}	4.704×10^{-10} A	4.7041×10^{-10} A	4.7041×10^{-10} A	3.305×10^{-10} A	2.91×10^{-9} A
I_{o2}	4.704×10^{-10} A	4.7048×10^{-10} A	4.7048×10^{-10} A	4.107×10^{-10} A	1.79×10^{-9} A
R_s	0.35 Ω	0.35 Ω	0.35 Ω	0.35 Ω	0.33 Ω
R_p	176.4 Ω	176.4 Ω	176.4 Ω	176.39 Ω	199.83 Ω
a_1	1	1	1	1.11	2
a_2	1.2	1.2	1.2	1	1.06
Fitness value J	–	1.704×10^{-9}	1.704×10^{-9}	1.33×10^{-6}	4.5×10^{-3}

Table 2
Comparison of STD values of extracted parameter using various methods.

Parameters	P-DE	B-DE	PSO	GA
I_{PV}	1.02×10^{-12}	0.005	0.01	0.005
I_{o1}	1.02×10^{-11}	1.47×10^{-7}	1.22×10^{-5}	8.13×10^{-6}
I_{o2}	3.34×10^{-14}	5.61×10^{-9}	4.05×10^{-5}	6.64×10^{-7}
R_s	3.34×10^{-14}	0.034	0.03	0.05
R_p	3.94×10^{-9}	54.92	738.6	79.4
a_1	0.1	0.13	5.63	0.25
a_2	0.1	0.2	12.92	0.26

solution might be obtained (Kumar et al., 1996). To investigate the consistency of the obtained solution, statistical analyses are carried out. For each parameter, standard deviation (STD) is calculated. A low value of standard deviation implies high consistency. Table 2 shows the result of standard deviation (STD) obtained by all EA methods. It can be seen that P-DE is the most consistent among all the four methods. A larger STD values for parallel resistance R_p in the remaining methods reveal that these cannot yield the synthetic values for each run.

Fig. 10b–h show the results of extracted parameters obtained by P-DE for 100 runs. The fitness value J results are also shown in Fig. 10a. It can be observed that for each run the fitness value is almost same. This implies that P-DE always extracts the synthetic values. It can be noted that for the ideality factors and saturation currents, a certain pattern exists. This is due to the equalization of two currents in (Ishaque, Salam et al.) i.e. either of ideality factors can be 1 or 1.2.

Table 3
Comparison of computation time using various methods (result for one run).

Parameters	P-DE	B-DE	PSO	GA
Time (s)	119	119	148	601

5.3. Speed of convergence

A zoomed view of convergence plot of Fig. 8a is shown in Fig. 8b. As can be seen, P-DE exhibits the best performance; it converges approximately three times faster than B-DE. Although the GA converges more rapidly than PSO but convergence to the global minima (synthetic values) is still difficult.

5.4. Computational efficiency

Table 3 outlines the computational efficiency of all EA methods. The P-DE and B-DE utilize slightly lesser time than PSO. Due to generation of too many eccentric and homogeneous individuals, GA requires much higher computation time.

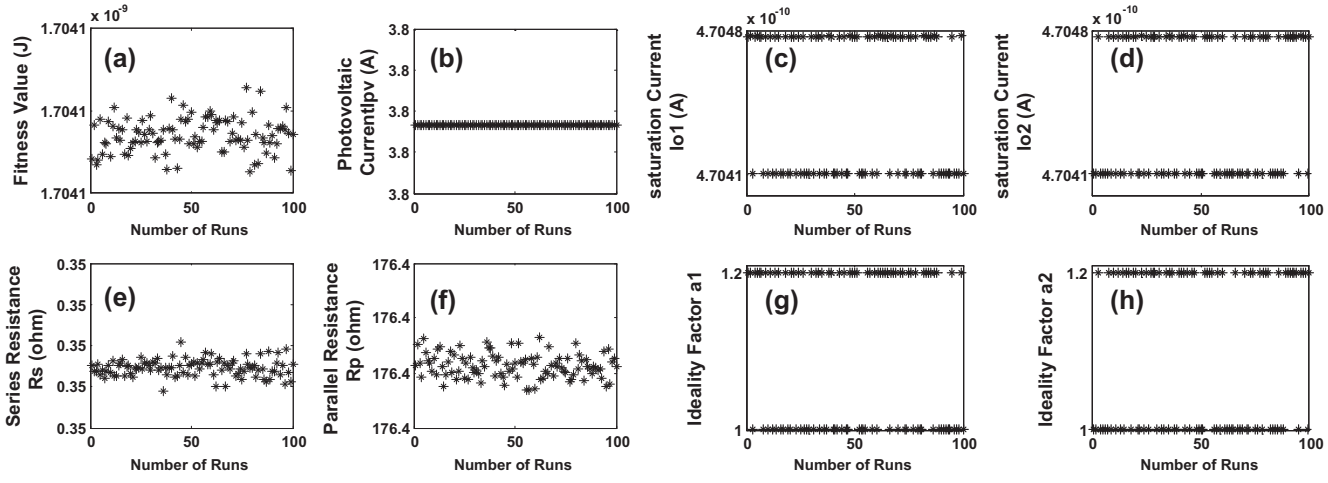

Fig. 10. (a) Fitness value for 100 runs (b–h) PV cell parameters for 100 runs.

Table 4
Summary of performance for various EA methods for parameter extraction of PV cell.

Evaluation criteria	GA	PSO	B-DE	P-DE
Accuracy of solution	Low	Medium	High	Very high
Consistency of solution	Medium	Low	Medium	Very high
Speed of convergence	Very low	Low	High	Very high
Computational efficiency	Very low	High	Very high	Very high
Number of control parameters	4	4	2	2

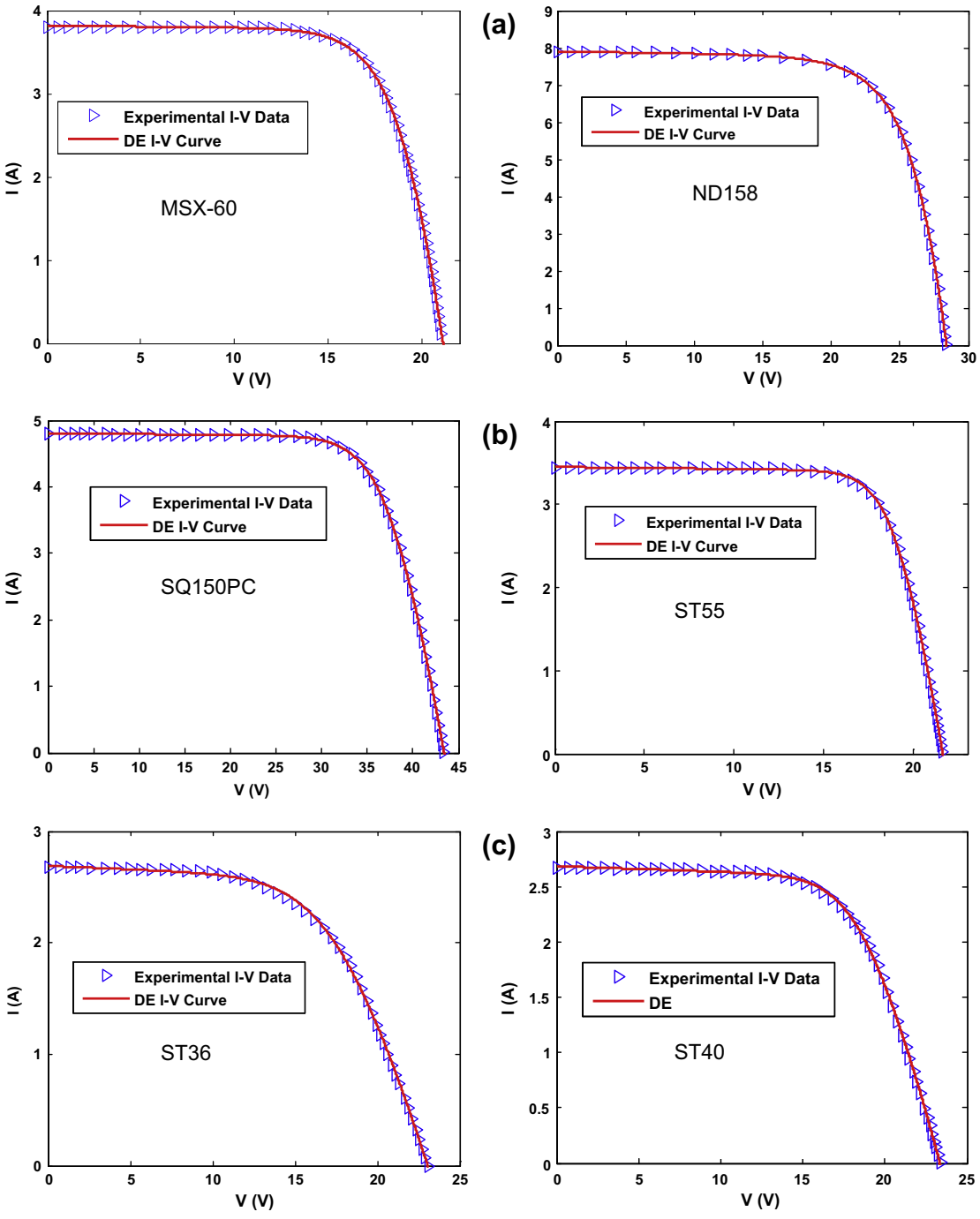

Fig. 11. *I–V* characteristics obtained from P-DE (a) multi-crystalline (b) multi-crystalline (c) thin-film PV modules.

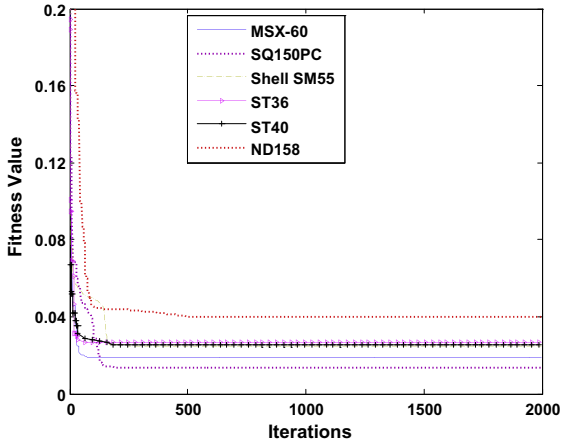

Fig. 12. Convergence performance of the fitness function using P-DE for various types of modules.

5.5. Number of control parameters

For any EA method, fewer control parameters ease the tuning effort in an optimization problem. The P-DE and B-DE exhibit only two parameters (F and CR) to that of PSO parameters (w, c_1, c_2 and V_{max}). Due to various control parameters (i.e. crossover rate, mutation factor, number of Childs in elite strategy and migration factor) involve in GA, it appears to be least preferable choice in parameter extraction.

Table 4 summarizes the performance of the four methods based on the five evaluation criteria described above. Clearly, P-DE method appears to be the best in all aspects of the evaluation. The proposed penalty function offers several advantages such as: (1) provides rapid convergence, (2) precise extraction parameters, (3) less number of iterations are required, (4) shows great consistency, and (5) few control parameters.

6. Results for DE with experimental data

The extraction method is validated by measured parameters using selected PV modules. From the comprehensive analyses carried out, it was clear that the P-DE is the most efficient method among the four; hence, only this method is considered here for physical validation. The experimental data is extracted from the manufacturer's datasheet. Six different modules of different brands/models are utilized; these include the multi-crystalline (MSX60 and ND158), mono-crystalline (SQ150PC and ST55) and thin-film (ST36 and ST40) types. Note that for experimental validation, the data is significantly fewer compared to the synthetic. However, for consistency, the same simulation parameters for P-DE are used. Fig. 11 show the curve fitting of P-DE to the experimental dataset. It can be seen that fitting were almost perfect, hence confirming the accurateness of the parameter extraction.

Fig. 12 depicts the convergence analysis for P-DE for all six PV modules. Clearly, the P-DE method has a very low fitness value and fast convergence. Table 5 shows the extracted parameters computed from P-DE in relation to the experimental $I-V$ data. The results matched very well. However, it can be further improved by increasing the number of data.

7. Conclusion

In this paper, the seven parameters of a PV cell, namely the PV current (I_{PV}), diode saturation currents (I_{o1}, I_{o2}), diode ideality factors (a_1, a_2), series resistance (R_s) and shunt resistance (R_p) are extracted using three evolutionary algorithms (EA) methods. Out of these, two methods, namely genetic algorithm (GA) and particle swarm optimization (PSO) are developed from the previous work in the literature. Furthermore, two variations of differential evolution (DE), namely boundary based differential evolution (B-DE) and penalty based differential evolution (P-DE) are proposed. All the methods are evaluated based on the following criteria. (1) Accuracy of solution (2) consistency of solution, (3) speed of convergence, (4) computational efficiency, and (5) the required number of control parameters. Results show that GA method in its conventional form is not appropriate to be used in parameter extraction. Slightly better results can be obtained using PSO at the expense of large number of iterations. In general, DE appears to be the most promising. However, among the two variation of DE, the P-DE method is superior in all aspects of the evaluation. The feasibility of the proposed P-DE method has been validated by synthetic and experimental $I-V$ data set. The latter includes six PV modules of different types

Table 5
Results of parameter extraction by P-DE for various types of modules.

Parameters/module	MSX60	ND158	SQ150PC	SM55	ST36	ST40
I_{PV}	8.18 A	7.91 A	4.8 A	3.45 A	2.71 A	2.68 A
I_{o1}	1.72×10^{9} A	8.94×10^{-6} A	1.55×10^{-8} A	3.03×10^{-4} A	1.02×10^{-5} A	9.7×10^{-5} A
I_{o2}	5.14×10^{10} A	2.9×10^{-4} A	1.03×10^{-8} A	4.9×10^{-12} A	1.16×10^{-6} A	1×10^{-10} A
R_s	0.24 Ω	0.07 Ω	0.8 Ω	0.56 Ω	1.46 Ω	1.42 Ω
R_p	147.2 Ω	178.4 Ω	1064.6 Ω	4.9 KΩ	182.6 Ω	440.6 Ω
a_1	1.08	1.7	1.23	3.55	2	3
a_2	1.15	2.9	1.23	0.86	2	1.06
Time (s)	3.37	2.44	3.04	2.97	2.89	3.0
Fitness value J	6.2×10^{-2}	3.8×10^{-2}	1.4×10^{-2}	2.4×10^{-2}	2.6×10^{-2}	2.5×10^{-2}

(multi-crystalline, mono-crystalline, and thin-film) from various manufacturers.

Acknowledgment

The authors would like to thank Universiti Teknologi Malaysia for providing the facilities and research grant to conduct this research.

References

Chan, D.S.H., Phang, J.C.H., 1987. Analytical methods for the extraction of solar-cell single- and double-diode model parameters from I–V characteristics. Electron Devices IEEE Transactions on 34 (2), 286–293.

Charles, J.-P. et al., 1985. Consistency of the double exponential model with physical mechanisms of conduction for a solar cell under illumination. Journal of Physics D: Applied Physics 18 (11), 2261.

Chegaar, M., Azzouzi, G., et al., 2006. Simple parameter extraction method for illuminated solar cells. Solid-State Electronics 50 (7–8), 1234–1237.

Chegaar, M., Ouennoughi, Z., et al., 2004. Extracting dc parameters of solar cells under illumination. Vacuum 75 (4), 367–372.

Chih-Tang, S., Noyce, R.N., et al., 1957. Carrier generation and recombination in P–N junctions and P–N junction characteristics. Proceedings of the IRE 45 (9), 1228–1243.

da Costa, W. T., Fardin, J.F., et al. (2010). Identification of photovoltaic model parameters by differential evolution. Industrial Technology (ICIT) IEEE International Conference on, 2010.

De Soto, W., Klein, S.A., et al., 2006. Improvement and validation of a model for photovoltaic array performance. Solar Energy 80 (1), 78–88.

Duffie, J.A., Beckman, W.A., 2006. Solar Engineering of Thermal Processes. Wiley.

Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. Micro Machine and Human Science, MHS '95. Proceedings of the Sixth International Symposium on..

Gämperle, R., Müller, S.D., et al., 2002. A parameter study for differential evolution. WSEAS Int. Conf. on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, pp. 293–298.

Gottschalg, R. et al., 1999. The influence of the measurement environment on the accuracy of the extraction of the physical parameters of solar cells. Measurement Science and Technology 10 (9), 796.

Haouari-Merbah, M., Belhamel, M., et al., 2005. Extraction and analysis of solar cell parameters from the illuminated current–voltage curve. Solar Energy Materials and Solar Cells 87 (1–4), 225–233.

Haupt, R.L., Haupt, S.E., 2004. Practical Genetic Algorithms. Wiley.

Holland, J., 1975. Adaptation in Natural and Artificial Systems. MIT press.

Ishaque, K., Salam, Z., et al., 2011. Simple, fast and accurate two-diode model for photovoltaic modules. Solar Energy Materials and Solar Cells 95 (2), 586–594.

Jervase, J.A. et al., 2001. Solar cell parameter extraction using genetic algorithms. Measurement Science and Technology 12 (11), 1922.

Ji, M., Jin, Z., et al., 2006. An improved simulated annealing for solving the linear constrained optimization problems. Applied Mathematics and Computation 183 (1), 251–259.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. Neural Networks, Proceedings of IEEE International Conference on.

Kumar, A., Srivastava, A., et al., 1996. Robust and distributed genetic algorithm for ordering problems. High Performance Distributed Computing, Proceedings of 5th IEEE International Symposium on.

Liu, C.-C., Chen, C.-Y., et al., 2008. Physical parameters extraction from current-voltage characteristic for diodes using multiple nonlinear regression analysis. Solid-State Electronics 52 (6), 839–843.

Michalewicz, Z., Schoenauer, M., 1996. Evolutionary algorithms for constrained parameter optimization problems. Evolutionary Computation 4 (1), 1–32.

Moldovan, N., Picos, R., et al., 2009. Parameter extraction of a solar cell compact model using genetic algorithms, CDE 2009. Electron Devices, Spanish Conference on.

Nakanishi, F., Ikegami, T., et al., 2000. Modeling and operation of a 10 kW photovoltaic power generator using equivalent electric circuit method. Photovoltaic Specialists Conference, Conference Record of the Twenty-Eighth IEEE.

Ortiz-Conde, A., García Sánchez, F.J., et al., 2006. New method to extract the model parameters of solar cells from the explicit analytic solutions of their illuminated I–V characteristics. Solar Energy Materials and Solar Cells 90 (3), 352–361.

Price, K.V, Storn, R.M., et al., 2005. Differential Evolution A Practical Approach to Global Optimization. Springer.

Shi, Y., Eberhart, R.C., 1998. Parameter Selection in Particle Swarm Optimization. In: Proceedings of the 7th International Conference on Evolutionary Programming VII. Springer, pp. 591–600.

Storn, R., Price, K., 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11 (4), 341–359.

Taheri, H., Salam, Z., et al., 2010. A novel maximum power point tracking control of photovoltaic system under partial and rapidly fluctuating shadow conditions using differential evolution. In: IEEE Symposium on Industrial Electronics and Applications Malaysia, pp. 82–87.

Villalva, M.G., Gazoli, J.R., et al., 2009. Comprehensive approach to modeling and simulation of photovoltaic arrays. Power Electronics, IEEE Transactions on 24 (5), 1198–1208.

Walker, G., 2001. Evaluating MPPT converter topologies using a MATLAB PV model. Journal of Electrical and Electronics Engineering, Australia 21 (1), 8.

Ye, M., Wang, X., et al., 2009. Parameter extraction of solar cells using particle swarm optimization. Journal of Applied Physics 105 (9), 094502–094508.

Zwe-Lee, G., 2004. A particle swarm optimization approach for optimum design of PID controller in AVR system. Energy Conversion, IEEE Transactions on 19 (2), 384–391.