Parameter extraction of solar photovoltaic modules using penalty-based differential evolution

Kashif Ishaque\textsuperscript{a,b}, Zainal Salam\textsuperscript{a,*}, Saad Mekhilef\textsuperscript{c}, Amir Shamsudin\textsuperscript{a}

\textsuperscript{a}Faculty of Electrical Engineering, Universiti Teknologi Malaysia, UTM 81310, Skudai, Johor Bahru, Malaysia
\textsuperscript{b}Karachi Institute of Economics and Technology, Karachi 75190, Pakistan
\textsuperscript{c}Department of Electrical Engineering, Faculty of Engineering, University Malaya, Kuala Lumpur, Malaysia


corresponding author. Tel.: +60 75536187; fax: +60 75566272.
E-mail addresses: kashif.ishaque@pafkiet.edu.pk (K. Ishaque), zainals@fke.utm.my (Z. Salam).

Abstract

This paper proposes a penalty based differential evolution (P-DE) for extracting the parameters of solar photovoltaic (PV) modules at different environmental conditions. The two diode model of a solar cell is used as the basis for the extraction problem. The analyses carried out using synthetic current–voltage (I–V) data set showed that the proposed P-DE outperforms other Evolutionary Algorithm methods, namely the simulated annealing (SA), genetic algorithm (GA), and particle swarm optimization (PSO). P-DE consistently converges to the global optimum values very rapidly. The performances are evaluated using the well known quality test and student T-tests. Furthermore, the P-DE extraction method is practically validated by six solar modules of different types (multi-crystalline, mono-crystalline, and thin-film). The results were found to be in close agreement with the experimental I–V data set, especially at very low irradiance values. The latter can be very useful in predicting the performance of the solar system under partial shading conditions. The main application of the proposed work is the possibility of developing a highly accurate simulator for solar PV system designer.

1. Introduction

Solar is envisaged to be an important source of energy in the future. In particular, the photovoltaic (PV) power system, which converts solar energy to electrical power, is becoming a popular renewable energy source due to its long term economic prospect and ease of maintenance. However, due to high initial cost of such a system, optimal capturing of the available solar energy has to be ensured. Enormous amount of work has been carried out to physically improve the performance of solar cells/modules [1–8]. However, it appears that a proper system design also plays a vital role in increasing the overall efficiency. One area that could complement...
this effort is the development of a reliable and efficient PV simulator which can be used to optimize the system design prior to installation [9–11].

The accuracy of commercially available software for PV module or system simulation mainly depends on the accuracy of the solar cell/module models and the extraction method being used to determine the model's parameters. The choice of a model that closely emulates the characteristics of solar modules is very crucial; a model is known to be accurate if it fits the measured I–V data at all operating conditions. Over the years, several models are introduced – among the more popular ones are the circuit-based single diode [12] and the two diode model [13]. The latter, despite it is more computationally intensive, is preferable because its I–V characteristics closely resemble the behavior of a physical module [13].

Generally, there are two possible approaches to extract the solar module parameters: (1) the analytical [14–17] and (2) numerical extraction techniques [18–23]. The former 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 (Isc), open-circuit voltage (Voc), and slopes of the I–V characteristic at the axis intersections. Accuracy-wise, the approach relies heavily on the correctness of the selected points on the I–V curve. It has to be noted that the I–V curve is highly non-linear and any wrongly selected points may result in significant errors in the computed parameters. Furthermore, a typical module datasheet provides only information at Standard Test Condition (STC). However, it is known that the parameters vary with environmental conditions such as irradiance and temperature.

On the other hand, the numerical extraction technique is based on certain mathematical algorithm to fit all the points on the I–V curve. More accurate results can be obtained because all the points on the I–V curve are utilized. Deviation of several data points may not severely affect the accuracy of the parameters as in the case of the analytical approach. However the curve-fitting algorithm requires extensive computation. Its accuracy depends on the type of fitting algorithm, the cost function and the initial values of the parameters to be extracted [18]. As the number of parameters in the model increases, the conventional extraction methods lose their ability to provide accurate values.

Evolutionary Algorithms (EAs) appear to be a natural choice to extract the module parameters at conditions other than STC. Due to the fact that the objective function in the extraction process tend to be multi-modal, EA methods can be very effective regardless of gradient and initial condition information [9,24]. To date, various EA techniques such as genetic algorithm (GA) [25–28], simulated annealing (SA) [29] and particle swarm optimization (PSO) [30–34] have been employed for parameter extraction of solar module. For GA, serious shortcomings, namely low speed and degradation for highly interactive fitness function are reported [35,36]. Several authors, for example [29] employed SA for the extraction process. However, the trade-off between the cooling schedule and initial temperature is the major issue that makes SA a less preferable choice. Although PSO is seen as a better alternative to GA and SA, several drawbacks have been identified, namely (1) it cannot guarantee the consistency of extracted parameters [30] and (2) it requires a large number of iterations (typically >10,000) for the solution to converge [30]. To improve the consistency of the solutions, authors in [34] have employed PSO in conjunction with cluster analysis. The work is carried out in two phases. First, it uses the PSO algorithm to obtain the module parameters (solutions); thereafter, the cluster analysis technique is used to “filter” the non-feasible solutions. Although, such procedure increases the reliability and consistency, it requires that all previously visited points to be stored. In a complex multi-modal problem of high dimension, the computation burden, particularly memory requirement is increased. Furthermore, considering the fact that extraction is the main component of a PV system simulator, the overall simulation speed would be greatly compromised.

One issue that is of major concern in all the above-mentioned methods is premature convergence [37,38] which relates to the loss of diversity in the population. The problem can be traced by understanding the nature of these algorithms: the presence of randomness in the selection process along with a fixed population size. For a problem with multiple solutions, the entire population will soon have to converge to only one of these solutions. The loss in diversity is undesirable, since the maintenance of all potential solutions is crucial to locate the global optimum. Generally, the problem is avoided by broadening the search space of parameters [39]. However, this could lead to slow convergence. Moreover, besides searching for a solution that is optimal, the extraction process also needs to satisfy several boundary constraints. Even if the extracted parameters yield a perfect match to the corresponding I–V characteristic curves, the values of the obtained parameters may not be a true reflection of a solar module from the physical point of view. It is therefore very necessary to correctly identify the boundaries; for example, the values of parameters can never be negative or zero. In certain cases, the numerical relationships between the parameters must be observed.

Recently, another type of EA method – the differential evolution (DE) was introduced [40]. DE is known to exhibit several advantages; it features rapid convergence, good accuracy and compared to other EA methods, it requires fewer control parameters. The success of DE can be attributed to its robust mutation scheme – a distribution of substantial number of difference vector with considerable length reduces the probability of being trapped in a local minimum [40]. However, due to the differential scheme, the resulting parameters are more likely to be negative or exceeding the boundary constraints. Consequently, these “unfeasible” parameters will not take part in the next evolution, possibly leading to a premature convergence.

This is where the inclusion of some kind of a penalty function for DE becomes very crucial. Unlike the conventional DE which is unconstrained in nature, the penalty based DE (P-DE) will always ensure the availability of the solutions in the feasible region. As a result, more solutions are created; they actively take part in the evolution process, thus improving the diversity, accuracy and consistency of the algorithm.

In this work, the performance of P-DE for solar module parameter extraction is investigated and compared to three other popular EA methods, namely the SA, GA and PSO. The final goal is to determine the suitability of these methods when used in a PV system simulator. To evaluate their effectiveness, statistical analysis, namely the Quality and Student’s T-test are employed. This is in contrast to [24] and [41], in which a non-statistical approach is used for evaluation. The I–V curves used for these analyses are generated using synthetic data. Furthermore, in order to ensure its practicality, the P-DE method is validated using six solar modules of different types (multi-crystalline, mono-crystalline, and thin-film) from various manufacturers.

The remainder of this paper is organized in the following way. Section 2 describes the formulation of parameter extraction of the two diode model. Section 3 describes the P-DE algorithm proposed in this paper. Section 4 evaluates the effectiveness of proposed method using synthetic I–V data set based on quality and student’s T-tests. Section 5 shows the validation of the proposed method to the experimental I–V data set. Section 6 presents the two possible strategies for solar PV simulation using extracted parameters. Finally, Section 7 draws the conclusions of this paper.
2. Formulation of parameter extraction problem

The main objective of solar module parameter extraction problem is to minimize the difference between synthetic or measured and simulated solar module current at various environmental conditions. In literature, two main circuit models are used to describe the behavior of the solar cell. First is the single diode model which is primarily expressed by the Shockley diode equation. This model is known to have satisfactory performance under normal operating conditions but often fails to represent the true behavior of solar cell at low irradiance [42]. The more accurate counterpart, commonly known as the two diode model, includes the effect of recombination losses in the space-charge by incorporating an additional Shockley diode relationship in the current equation. This model significantly improves the accuracy but at the expense of computational additional of parameters.

Fig. 1 shows the two diode model, which will become the basis for the parameter extraction in this work. The output current of the cell can be described as:

\[ I = I_{ph} - I_{01} \left[ \exp \left( \frac{V + IR_s}{a_1 V_m} \right) - 1 \right] - I_{02} \left[ \exp \left( \frac{V + IR_s}{a_2 V_m} \right) - 1 \right] \]

(1)

where subscript C represents a single cell, \( I_{ph} \) is the cell-generated photocurrent; \( I_{01} \) and \( I_{02} \) are the reverse saturation currents of diode 1 and diode 2, respectively. The \( I_{02} \) term is introduced to compensate the recombination loss in the depletion region as described in [13].

Other variables are defined as follows: \( V_m \) (kT/q) is the thermal voltage of the solar cell, 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 \( a_2 \) are 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 (compared to the single diode model), it requires the computation of seven parameters, namely \( I_{ph}, I_{01}, I_{02}, R_s, R_p, a_1, a_2 \).

It can be noticed that Fig. 1 characterizes a single solar cell. However, in most of the cases, the I–V data is only available for commercial solar modules (combination of several cells) [10,12]. Hence, Eq. (1) needs to be transformed into module structure which is effectively the interconnection of solar cells in series or/and parallel \( (N_s \times N_p) \) configuration. When the cells are arranged in series configuration, only the voltages \( (V_{in}, V_m) \) and cell voltage \( V \) are scaled to \( N_s \) times. The same way, the parallel arrangement only increases the corresponding current components in the module by \( N_p \) times.

Accordingly, the output characteristics of a module for \( N_s \times N_p \) identical solar cells can be represented as follow [43]:

\[
\begin{align*}
I &= N_s I_{ph}, I_{01} = N_s I_{01}, I_{02} = N_s I_{02} \\
V &= N_s V_m, V_m = N_s V_m \\
R_p &= \frac{N_s}{N_p} R_p, R_p &= \frac{N_p}{N_s} R_p
\end{align*}
\]

(2)

Thus, the output current equation of the PV module can be written as:

\[ I = I_{ph} - I_{01} \left[ \exp \left( \frac{V + IR_s}{a_1 V_m} \right) - 1 \right] - I_{02} \left[ \exp \left( \frac{V + IR_s}{a_2 V_m} \right) - 1 \right] - \frac{V + IR_s}{R_p} \]

(3)

2.1. Fitness function

The extraction performance is evaluated using an objective function, \( J \). In this paper, \( J \) is formulated as the root mean square of the difference between the measured and simulated module current data. For \( N \) numbers of I–V data set, it can be written as:

\[ J = \sqrt{\frac{1}{N} \sum_{n=1}^{N} f(I_{m}, V_{m}, \Phi)^2} \]

(4)

where

\[ f(I, V, \Phi) = I_{ph} - I_{01} \left[ \exp \left( \frac{V + IR_s}{a_1 V_m} \right) - 1 \right] - I_{02} \left[ \exp \left( \frac{V + IR_s}{a_2 V_m} \right) - 1 \right] - \frac{V + IR_s}{R_p} \]

(5)

and \( \Phi = \{ I_{ph}, I_{01}, I_{02}, R_s, R_p, a_1, a_2 \} \) are the seven parameters of the PV module to be extracted.

The aim of the extraction procedure is to minimize Eq. (4) with respect to \( \Phi \). A smaller value of \( J \) implies the deviation between the module current and the one computed by the extraction method is small. Ideally, a zero value for \( J \) is desired. It can be seen from Eq. (4) that \( J \) is a nonlinear function with no apparent quadratic function. 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 [30].

3. Differential evolution

Differential evolution (DE) is a search and optimization algorithm which was first introduced in [40]. It has three main advantages: (1) ability to locate the accurate global optimum regardless of the initial parameter values, (2) rapid convergence, and (3) utilizing few control parameters. The optimization procedure in DE is similar to GA. However, unlike GA, which relies on crossover, DE primarily utilizes mutation operation (i.e. difference vector) as a search and selection mechanism. Like other EA methods, DE begins with the initialization of a random population, which is then improved using mutation, crossover, and selection. The process is repeated through the generations until the stopping condition is reached – usually a satisfactory (good) fitness value or a predefined limit for the number of generations \( G_{\text{max}} \).

DE works on a population of candidate solutions, \( P^G \). These candidate solutions are known as the individuals of the population. In particular, DE creates a population \((NP)\) of \( D \)-dimensional real-valued parameter vectors \( X^i_G \) as follows:

\[ P^G = (X^i_G) \ i = 1, 2, \ldots, NP, \quad G = 1, 2, \ldots, G_{\text{max}} \]

(6)

\[ X^i_G = (X^i_G^j) \ j = 1, 2, \ldots, D \]

(7)

The index, \( G = 1, 2, \ldots, G_{\text{max}} \), indicates the generation to which a vector belongs. Additionally, each vector has a population index, \( i \), which ranges from 1 to \( NP \). Parameters within vectors are indexed with \( j \) from 1 to \( D \). Fig. 2 shows the various stages of DE process in its operational block diagram.
3.1. Initialization

In order to begin the optimization process, an initial population of \( NP \) \( D \)-dimensional real-valued parameter vectors \( X_i^0 = [X_i^0, X_i^0, \ldots, X_i^0] \) is created. Each vector forms a candidate solution to the multidimensional optimization problem. Initial parameter values are randomly selected within the interval \([X_L, X_H]\), where \( X_L = [X_{1L}, X_{2L}, \ldots, X_{DL}] \) and \( X_H = [X_{1H}, X_{2H}, \ldots, X_{DH}] \) are the lower and upper bound of the search space, respectively. Hence,

\[
X_{ij}^0 = X_i + \text{rand}(1)(X_H - X_L) \tag{8}
\]

Fig. 3a explains the initialization procedure of DE. For simplicity, a 2-dimensional parameter vectors is shown here.

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 vector and finally an offspring vector is obtained by recombining the donor with the target vector is called trial vector. For a given parameter vector \( X_i^0 \), three vectors \((X_i^0, X_{jG}, X_{kG})\) are randomly selected in the range \([1, NP]\), such that the indices \( i, r_1, r_2, r_3 \) are distinct. A donor vector \( V_{iG} \) is created by adding the weighted difference between the two vectors to the third (base) vector, i.e.

\[
V_{iG} = X_{i}^{0} + F(X_{jG} - X_{kG}) \tag{9}
\]

where \( F \) is a mutation scaling factor, which is typically chosen from the range \([0,1]\). Fig. 3b illustrates the mutation process on a 2-D parameter space (showing constant cost contours of an arbitrary fitness function).

The inherent differential mutation in DE exhibits three properties that are crucial for an efficient mutation scheme [41]. First, DE guarantees a distribution with a zero mean value by randomly sampling the difference vectors. This implies that \( x_{r2} - x_{r3} \) has the same probability of being selected as the opposite \( x_{r3} - x_{r2} \) has. Secondly, the distribution of vector differentials is automatically self-scaled; DE scales the mutation step sizes by scaling their relative magnitudes. Thirdly, DE is rotational invariant since the mutation distribution generated by difference vectors will always lie within the feasible boundaries.

3.3. Crossover

The donor vector \( V_{iG}^{c+1} \) and the target vector \( X_i^{0} \) are mixed to yield the trial vector

\[
U_j^0 = [U_{j1}^0, U_{j2}^0, \ldots, U_{jD}^0] \tag{10}
\]

In DE algorithm, two kinds of crossover methods are used i.e. exponential and binomial (or uniform) [40]. In this work, binomial crossover strategy is used which can be described as:

\[
U_{ij}^0 = \begin{cases} 
V_{ij}^0, & \text{if } \text{rand} \leq \text{CR} \text{ or } j = j_{\text{rand}} \\
X_{ij}^0, & \text{otherwise}
\end{cases} \tag{11}
\]

where \( \text{CR} \) is known as the crossover rate. It is another control parameter, just like \( F, j_{\text{rand}} \in [1, 2, \ldots, D] \) is a randomly chosen index, which ensures that \( U_{ij}^0 \) attains at least one element from \( V_{ij}^0 \). Fig. 4a plots the possible trial vectors that can result from uniformly crossing a mutant vector, \( V_{ij}^0 \), with the vector \( X_{ij}^0 \).

In the conventional DE, each population vector is crossed with a randomly generated mutant vector. In view of the fact that the current population of vectors already satisfied all the boundary constraints, only contributions from the mutant vectors have the potential to violate the parameter limits. Therefore, the checks on the bounds are required only when a mutant parameter is selected for the trial vector. Fig. 5 depicts the resulting trial vector obtained through conventional DE for 10 generations. For simplicity, only trial vectors for \( l_{11} \) and \( l_{22} \) are shown. It can be observed that many of the \( l_{11} \) and \( l_{22} \) vectors are located in the infeasible region – violating the specified boundaries set for them.

In an effort to bring these vectors back into the feasible range, a penalty function is proposed in this work. It ensures that the parameter values (trial vectors) lie within the allowable range after recombination. Any parameter that violates the specified limits is replaced with random values using:

\[
U_{ij}^{c+1} = \begin{cases} 
U_{ij}^{c+1} - \text{rand}(0, 1)(X_{ij} - X_{H}) , & \text{if } U_{ij}^{c+1} \geq X_{H} \\
U_{ij}^{c+1} + \text{rand}(0, 1)(X_{L} - X_{ij}) , & \text{if } U_{ij}^{c+1} \leq X_{L} \end{cases} \tag{12}
\]

The effect of proposed penalty function for the two-dimensional parameter space is shown in Fig. 4b. It can be observed that Eq. (12) takes the progress of solution toward the optimum into account by shifting a violated parameter towards the feasible region.

In addition, it should be noted that the selection of suitable search range also plays a crucial role. If a narrow search range is employed, the difference term, i.e. \((X_{H} - X_{L})\) in Eq. (12) could not be large. Consequently, the shifted parameter may still lie in the infeasible region. To avoid such situation, a broader search range should be chosen. Such selection offers two possible advantages.

3.4. Evaluation and Selection

The selection operation at \( G = G + 1 \) is described as

\[
X^{c+1} = \begin{cases} 
U_{ij}^{c} , & \text{if } J(U_{ij}^{c}) < J(X_{ij}^{c}) \\
X_{ij}^{c} , & \text{otherwise}
\end{cases} \tag{13}
\]

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 decline.

In summary, due to the introduction of the penalty function, the selection of new population in DE will always be directed towards a feasible region and hence a better solution will be obtained. Additionally, a large value of \( F \) will increase the difference term in Eq. (9) which results in the mutant vectors distribution to be multi-modal. Once the population settles around the global minima region, the difference vector distribution becomes uni-modal and
mutation steps exhibit both a scale and an orientation that are appropriate for a local search. Consequently, the algorithm will converge to global minima vary rapidly. The pseudo code of extraction process using proposed algorithm is described in Fig. 7.

4. Results with synthetic data

The solar module parameter extraction using the selected EA methods (GA, SA, PSO and P-DE) are evaluated against the synthetic data obtained from [14]. Such approach is also utilized by other researchers, for example [30]. The synthetic data were calculated using the two diode model with the following seven parameters at STC: $I_{ph} = 3.8$ A, $I_{o1} = 4.7 \times 10^{-10}$ A, $I_{o2} = 2.11 \times 10^{-10}$ A, $R_s = 0.32 \Omega$, $R_p = 200 \Omega$, $a_1 = 1$ and $a_2 = 2$. Using these values, the synthetic $I-V$ curve is generated. Then the $I-V$ curve fitting using the above-mentioned EA methods is performed.

To ensure a fair evaluation, similar simulation conditions, i.e. population size, maximum generation number and search ranges are maintained. Each EA method is executed for 30 runs. The population size ($NP$), is chosen to be 70. This is a reasonable choice; typical value of $NP$ ranges between $5 \times D$ to $10 \times D$. The maximum generation number ($G_{\text{max}}$) is set to 40,000. Even though, P-DE can converge with much lesser 1000 iterations, this value is selected to be consistent with [30]. The mutation factor ($F$) is set at 0.8. There is no strict rule on the selection of $F$ but in most cases, $F > 0.4$ [40]. The crossover rate ($CR$) is chosen to be 1. Large value of $CR$ intensifies the diversity of population, thus improving the convergence speed [40]. Moreover, a high value of $CR$ is desirable as the parameters in the model are highly correlated [24]. The search ranges were set as follows: $I_{ph} \in [0, 7.8], I_{o1,2} \in [1e^{-12}, 1e^{-5}], R_s \in [0, 1], R_p \in [50, 1000], a_{1,2} \in [0.5, 4]$. Note that the search ranges are deliberately designed to be very broad to ascertain which method converge to global minima vary rapidly. The pseudo code of extraction process using proposed algorithm is described in Fig. 7.
converges most rapidly with highest level of consistency. The computations are carried out using the standard desktop PC with 3 GHz duo core Intel processor, 4 GB RAM, under Windows XP operating system.

The DE/best/1/bin strategy is employed for the 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. The work described in [44] is used as the basis for the SA implementation with the following settings: the initial temperature $T_0$ was calculated based on [45]; iteration number $L = 10$ in every temperature value and cooling factor $\beta = 0.95$. For GA, a crossover rate, $P_c = 0.8$ and mutation rate, $P_m = 0.2$, is used. Furthermore, a GA function named as “gr” in MATLAB is utilized. Finally for the PSO, the following control variables are used: learning factors $c_1 = c_2 = 2$, inertia factors $w_{\text{max}} = 0.9$, $w_{\text{min}} = 0.4$ and velocity clamping factor $V_{\text{max}} = 0.5$ [30]. The PSO has been implemented according to the description given in [30].

Table 1 displays extracted parameters obtained by the four algorithms; they are selected from the best of 30 runs. It can be seen that P-DE is exceptionally accurate; the extracted parameters match almost exactly to the synthetic values [14]. Other methods yield mixed results; PSO appears to be very close, followed by SA and finally GA. It should be noted that these are the best values; therefore, the results should be taken cautiously. It shall be shown later that certain amounts of deviation are observed when the maximum, minimum and mean values are considered.

Additionally, computational speed is compared. It can be clearly observed that the P-DE method is the fastest; it takes 42 s to compute the parameters for 40,000 generations. The nearest is PSO, which requires approximately three times longer. The computation time for the GA and SA are much slower. However, as shall be noted in the following section, PSO will not converge to the desirable solution when a low number of generations are used. On the other hand, P-DE can consistently converge to the global optimum even for a low number of generations (e.g. 500). The superior computational speed of P-DE can be especially useful when it is used for PV simulator.

4.1. Statistical analysis

To evaluate the consistency of the EA methods to locate the true global optimum, a comprehensive statistical analysis using the quality test is carried out. An independent run (trial) can be classified as a “success” when the final fitness value ($J$) is below a predetermined limit known as the value-to-reach, or “VTR”. A trial that does not reach the VTR within a predetermined maximum number of evaluations, $G_{\text{max}}$, is treated as a “failure”. By employing VTR and $G_{\text{max}}$ as the criteria for success evaluation, one can estimate (1) the probability of an EA method to locate the basin of attraction to which the global optimum belongs and (2) the speed of convergence. In this case, the VTR and $G_{\text{max}}$ are set to $1/2^210^{-6}$ and 40,000, respectively.

The following numerical test for convergence has been adopted. In the $i$th run, if the VTR is not achieved at the end of $G_{\text{max}}$, this run is considered unsuccessful. Accordingly, the convergence measure ($CM$) that arrives at the optimal solution with a sufficient precision can be calculated as follows:

$$CM = \frac{\sum_{i=1}^{nT} J_{\text{final}}}{nS}$$

where $n_s$ and $n_T$ are the successful and total runs of an EA method, respectively. Using the convergence measure, the quality of the EA method can be computed as [46]:

$$Q = \frac{CM}{P_{\text{cm}}}$$

where $P_{\text{cm}}$ is the percentage of successful runs. Eq. (15) indicates that for a smaller value of $Q$, a better convergence performance is achieved. If $P_{\text{cm}} = 0$ then $Q = \infty$; this implies that the algorithm never reaches the VTR for the specified generation size. For each EA method, the $Q$ values are listed in Table 2. Additionally, their minimum ($J_{\text{min}}$), maximum ($J_{\text{max}}$), average ($J_{\text{mean}}$) and standard deviation ($\sigma$) values are also tabulated.

Fig. 8 shows the qualitative representation of the average evolution performance for all methods. For GA, after 200 iterations, the evolution process tends to be very slow, which implies that new Childs are not sufficiently created. As a result, GA prematurely

---

**Figure 7.** Pseudo-code for the proposed P-DE algorithm.
converges to a local minima at $J_{\text{min}} = 0.03$. This is also evident from Table 2; the minimum, maximum and mean values for $J$ are in the order of $10^{-2} – a value which is very far from VTR. The performance of SA and PSO is somewhat better. In the best cases, as illustrated in Table 2, $J_{\text{min}}$ for SA and PSO are relatively close to VTR. However, their $J_{\text{max}}$ values are still far from VTR, resulting in poor standard deviation. Furthermore, it can be seen in Fig. 8 that the exploration capability for both SA and PSO is not efficient. As a result, their $J$ appears to stagnate after 15,000 generations. In contrast, P-DE exhibits very low $J$, an indication of high accuracy. From the zoomed view of Fig. 8, it can be observed that up to about 500 iterations, its $J$ is constantly improved until it settles to a mean value of $2.997 \times 10^{-5}$. Furthermore, the values of $J_{\text{min}}, J_{\text{max}}$, and $J_{\text{mean}}$ are the same; its standard deviation is negligible, which implies that the solution obtained for every individual run is highly consistent. In addition, since $Q = J_{\text{min}}$, P-DE is guaranteed to locate to the global minimum for every run.

The excellent performance of P-DE can be attributed to several factors. First, the large values of $F$ and $CR$ enhance the diversity of parameter search space; consequently, each generation result in a new set of children. Additionally, an inherent refining capability due to penalty function guarantees the parameters to be always located in the feasible region. Finally, the P-DE’s one-to-one selection or “knock-out” competition ensures that any parameter that loses the single competition is eliminated; only winning parameters are allowed to take part in the next generation. The overall effect by these factors results in highly consistent and accurate solutions with rapid convergence.

The consistency and reliability of P-DE solutions avoid the requirement for a post-processing procedure to filter out the scattered parameter, such as the cluster analysis as reported in [34].

Another approach to demonstrate the statistical significance of the obtained results is to utilize the well known Student’s T-test [47]. A confidence level of 0.95 was used in the test. The final values of $J$ returned by the P-DE are compared to the final value obtained by other algorithms used as a benchmark. Table 3 shows the results of the test. Indicated with “+” shows that, the P-DE statistically outperforms the other EA methods mentioned in the column. The “-” indication is the case when the two EA methods have the same performance and indicated with “_” is the case when the other EA method outperforms the P-DE.

To investigate the effect of control parameters ($NP, CR, F$) on the performances of P-DE, two types of analyses are carried out. For consistency, the same value of generation size i.e. 40,000 is used. A control parameter can be classified as “acceptable” when $J < VTR$; otherwise it is treated as “unacceptable”. First, the effect of $NP$ is examined. For this purpose, $NP$ is varied while $CR$ and $F$ are fixed at 1 and 0.8, respectively. Fig. 9a illustrates values of $J_{\text{min}}$ as $NP$ is varied from 10 to 100. As shown in the figure, for $NP \geq 30$, $J$ converges to VTR. But for $NP < 30$, the values of $J$ could not converge, hence unacceptable. It has to be noted that the typical value of $NP$ suggested by many researchers is $8D–10D$ [30]. For the case under investigation, since $D = 7$, any value of $NP$ below 60 is considered to be very low. However, as can be observed in Fig. 9a, even with a low population size, P-DE appears to work very well. The advantage of using low value of $NP$ is the significant reduction in the computational speed.

Fig. 9b shows the result when $CR$ and $F$ are varied while $NP$ is held at 70. A total of 49 combinations ($7 \times 7$) for the values of $CR$ and $F$ (ranged from 0.4 to 1) are tested. It can be observed that P-DE performs very well for a wide range of $CR$ and $F$, except at the extremes.

---

**Table 1**

Extracted parameter using various methods (the best result for 30 runs).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Synthetic values</th>
<th>P-DE</th>
<th>SA</th>
<th>GA</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{\text{in}}$</td>
<td>3.800A</td>
<td>3.800A</td>
<td>3.799A</td>
<td>3.779A</td>
<td>3.800A</td>
</tr>
<tr>
<td>$I_{\text{c1}}$</td>
<td>$4.700 \times 10^{-10}$ A</td>
<td>$4.700 \times 10^{-10}$ A</td>
<td>$5.637 \times 10^{-10}$ A</td>
<td>$9.110 \times 10^{-9}$ A</td>
<td>$4.621 \times 10^{-10}$ A</td>
</tr>
<tr>
<td>$I_{\text{c2}}$</td>
<td>$2.110 \times 10^{-06}$ A</td>
<td>$2.110 \times 10^{-06}$ A</td>
<td>$4.251 \times 10^{-06}$ A</td>
<td>$2.687 \times 10^{-06}$ A</td>
<td>$1.977 \times 10^{-06}$ A</td>
</tr>
<tr>
<td>$R_s$</td>
<td>0.320 $\Omega$</td>
<td>0.320 $\Omega$</td>
<td>0.319 $\Omega$</td>
<td>0.173 $\Omega$</td>
<td>0.320 $\Omega$</td>
</tr>
<tr>
<td>$R_p$</td>
<td>200,000 $\Omega$</td>
<td>200,000 $\Omega$</td>
<td>201,120 $\Omega$</td>
<td>508,900 $\Omega$</td>
<td>199,890 $\Omega$</td>
</tr>
<tr>
<td>$a_1$</td>
<td>1</td>
<td>1.005</td>
<td>1.008</td>
<td>1.380</td>
<td>0.999</td>
</tr>
<tr>
<td>$a_2$</td>
<td>2</td>
<td>2.001</td>
<td>2.185</td>
<td>3.998</td>
<td>1.984</td>
</tr>
<tr>
<td>Time (s/run)</td>
<td>--</td>
<td>42</td>
<td>322</td>
<td>360</td>
<td>110</td>
</tr>
</tbody>
</table>

**Table 2**

Algorithmic performance values for P-DE, GA, SA, and PSO.

<table>
<thead>
<tr>
<th></th>
<th>$J_{\text{min}}$</th>
<th>$J_{\text{max}}$</th>
<th>$J_{\text{mean}}$</th>
<th>$\sigma$</th>
<th>$Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>$2.997 \times 10^{-9}$</td>
<td>$2.997 \times 10^{-9}$</td>
<td>$2.997 \times 10^{-9}$</td>
<td>$4.360 \times 10^{-20}$</td>
<td>$2.997 \times 10^{-9}$</td>
</tr>
<tr>
<td>SA</td>
<td>$5.990 \times 10^{-5}$</td>
<td>$2.590 \times 10^{-4}$</td>
<td>$1.292 \times 10^{-4}$</td>
<td>$6.830 \times 10^{-5}$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>GA</td>
<td>$2.070 \times 10^{-2}$</td>
<td>$3.500 \times 10^{-2}$</td>
<td>$3.100 \times 10^{-2}$</td>
<td>$4.080 \times 10^{-3}$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>PSO</td>
<td>$5.217 \times 10^{-6}$</td>
<td>$1.980 \times 10^{-2}$</td>
<td>$5.290 \times 10^{-3}$</td>
<td>$7.710 \times 10^{-3}$</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>
From this observation, both CR and F should be set in the range of 0.6–0.9 to ensure the rapid convergence to global optimum. However, CR = 1 can still be utilized provided that F is selected between 0.7–0.9. These observations are in close agreement to [40].

5. Results for P-DE with experimental data

The extraction method is validated by measured parameters using selected solar modules. The experimental data is extracted from the manufacturer’s datasheet. Six different modules of different brands/models are utilized; these include the multi-crystalline (S75 and S115), mono-crystalline (SM55 and SQ150PC) and thin-film (ST36 and ST40) types. The experimental data is collected at three different irradiance levels, namely 1000 W/m², 600 W/m², and 200 W/m². The variations in temperature are not considered for brevity.

It must also be noted that although the extracted parameters might fit well to the I–V characteristics curves, there are possibilities that computed values do not make any sense in the physical...
point of view. To ensure that the extracted parameters realistically reflect the characteristics of physical module, an additional constraint on the saturation current is imposed: $Io_s$ is generally set to be 2–7 orders of magnitude larger than $Io_p$ [48].

With regard to the search ranges, for the experimental case, they are selected based on the data sheet specifications of each module and the characteristics of physical module, an additional constraint on the saturation current is imposed:

$$Io_s = 2\text{–}7\text{ orders of magnitude larger than } Io_p$$

Using the following relationships:

$$Io_p = Io_s - \frac{(Io_c + K_c \Delta T) G}{G_{STC}}$$

$$Io_s = Io_p = \frac{(Io_c + K_c \Delta T) V_{oc}}{G - G_{STC}}$$

$$Io_s = 2\text{–}7\text{ orders of magnitude larger than } Io_p$$

where $G$ and $G_{STC}$ are the actual and STC solar irradiance.

For a typical solar module, the series resistance $R_s$ is very low – in most of the cases, less than 2 $\Omega$ [49]. However, parallel resistance $R_p$ value is usually very high. The values of ideality factor $a_1$ and $a_2$ are typically between $0.5 \leq a \leq 4$ [13]. Considering these well-established information, the search ranges for experimental data are set as follows:

$$Io_p \in [0, 2Io_{ph}], Io_{s1,2} \in [0, Io_p \times 10^{5}], R_s \in [R_{so}, 2], R_p \in [R_{po}, 5000], a_{1,2} \in [0.5, 4]$$. 

Table 4

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Multi-crystalline</th>
<th>Mono-crystalline</th>
<th>Thin film</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G = 1000\text{ W/m}^2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Io_s$ (A)</td>
<td>4.700</td>
<td>4.710</td>
<td>3.450</td>
</tr>
<tr>
<td>$Io_p$ (A)</td>
<td>$9.860 \times 10^{-9}$</td>
<td>$7.030 \times 10^{-9}$</td>
<td>$2.340 \times 10^{-10}$</td>
</tr>
<tr>
<td>$Rs$ (k$\Omega$)</td>
<td>0.290</td>
<td>0.450</td>
<td>0.523</td>
</tr>
<tr>
<td>$R_s$ (k$\Omega$)</td>
<td>366.740</td>
<td>550.680</td>
<td>414.00</td>
</tr>
<tr>
<td>$a_1$</td>
<td>1.170</td>
<td>1.160</td>
<td>1.000</td>
</tr>
<tr>
<td>$a_2$</td>
<td>2.810</td>
<td>2.760</td>
<td>2.790</td>
</tr>
<tr>
<td>Time (s)</td>
<td>0.4</td>
<td>0.36</td>
<td>0.37</td>
</tr>
</tbody>
</table>

$G = 600\text{ W/m}^2$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Multi-crystalline</th>
<th>Mono-crystalline</th>
<th>Thin film</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Io_s$ (A)</td>
<td>2.810</td>
<td>2.840</td>
<td>2.080</td>
</tr>
<tr>
<td>$Io_p$ (A)</td>
<td>$9.890 \times 10^{-10}$</td>
<td>$7.970 \times 10^{-9}$</td>
<td>$5.040 \times 10^{-10}$</td>
</tr>
<tr>
<td>$Rs$ (k$\Omega$)</td>
<td>0.387</td>
<td>0.030</td>
<td>0.614</td>
</tr>
<tr>
<td>$R_s$ (k$\Omega$)</td>
<td>411.600</td>
<td>812.00</td>
<td>408.010</td>
</tr>
<tr>
<td>$a_1$</td>
<td>1.050</td>
<td>1.330</td>
<td>1.000</td>
</tr>
<tr>
<td>$a_2$</td>
<td>1.250</td>
<td>2.010</td>
<td>2.910</td>
</tr>
<tr>
<td>Time (s)</td>
<td>0.3</td>
<td>0.29</td>
<td>0.29</td>
</tr>
</tbody>
</table>

$G = 200\text{ W/m}^2$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Multi-crystalline</th>
<th>Mono-crystalline</th>
<th>Thin film</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Io_s$ (A)</td>
<td>0.941</td>
<td>0.975</td>
<td>0.699</td>
</tr>
<tr>
<td>$Io_p$ (A)</td>
<td>$2.700 \times 10^{-8}$</td>
<td>$6.420 \times 10^{-10}$</td>
<td>$4.540 \times 10^{-11}$</td>
</tr>
<tr>
<td>$Rs$ (k$\Omega$)</td>
<td>0.112</td>
<td>0.040</td>
<td>0.320</td>
</tr>
<tr>
<td>$R_s$ (k$\Omega$)</td>
<td>312.3</td>
<td>813.39</td>
<td>354.3</td>
</tr>
<tr>
<td>$a_1$</td>
<td>1.240</td>
<td>1.030</td>
<td>1.050</td>
</tr>
<tr>
<td>$a_2$</td>
<td>2.540</td>
<td>2.800</td>
<td>1.220</td>
</tr>
<tr>
<td>Time (s)</td>
<td>0.25</td>
<td>0.26</td>
<td>0.25</td>
</tr>
</tbody>
</table>

$$Io_1 = Io_2 = Io_0 = \frac{(Io_c + K_c \Delta T) V_{oc}}{G - G_{STC}} - 1$$

$$R_{so} = 0; R_{po} = \frac{V_{MP}}{Io_c - Io_0} = \frac{V_{oc} - V_{MP}}{Io_0}$$

Fig. 12. IV characteristics obtained from PDE for thin-film solar modules.
It should be noted that the resulting search ranges obtained by Eq. (16)–(18) will always ensure the availability of module parameters within the specified range. Moreover, since all the search ranges are computed using the datasheet parameters and some well-known information, this method can be applied to any type of solar cell/module/array.

As P-DE can converge very rapidly to the global minimum, \( G_{\text{max}} \) is set to a very low value, i.e., 500. This number is much lower than the one used for the synthetic data (40,000) and therefore the computational time is expected to be much faster. The remaining parameters for the P-DE implementation were chosen to be consistent with the synthetic case. Furthermore, as P-DE always converge to the global minimum, it is sufficient for the experiments to be based on a single independent run.

The solid line in Fig. 10–12 show the I–V curves generated using the parameters obtained by P-DE. It can be seen that the extraction results in I–V curves that accurately fit the whole range of experimental dataset. In particular, the accuracies at low irradiances match almost perfectly. Accurate extraction of solar module parameters at low irradiance is very crucial when the module is subjected to certain mismatch conditions, for example partial shading. For completeness, Table 4 shows the numerical values of the extracted parameters using P-DE for all modules under investigation.

To further validate the accuracy of P-DE, the extracted parameters are compared to the ones obtained using GA. This is achieved by comparing the absolute error, which is defined as the absolute difference between the experimental and computed solar module current values of the I–V curves for a given voltage point. Fig. 13 shows the results of the comparison for all the six modules of interest. In general, P-DE gives the better performance than GA for all cases. In particular, it should be noted that the error occurs around the MPP region is higher, but for P-DE the maximum error is kept below 0.02. On the other hand, the maximum error for GA exceeds beyond 0.1, which is a magnitude higher than the former.

6. Application of the extracted data for solar array simulation

As mentioned in the Introduction, the main application of the parameter extraction using P-DE is to design an accurate PV system simulator. Two types of circuit-based simulation schemes can be realized. Fig. 14 shows a circuit model having a current...
controlled source ($I_0$) and two resistors ($R_s$ and $R_p$). The parameter extraction block computes the seven model parameters based on the $I-V$ data set. Then, the computational block computes the current source signal ($I_s$) using the extracted parameters. This computational block can be carried out using any circuit simulator that is equipped with math functions facilities such as MATLAB, PSpice, PSim and PSCAD.

Another approach to circuit modeling is depicted in Fig. 15. The value of the current is obtained through numerical solution of the $I-V$ equation. The solution of Eq. (3) can be implemented using a numerical method (e.g., Newton Raphson) in any circuit simulator that has the programming capability.

7. Conclusion

In this paper, the seven parameters of a solar PV module, namely the photo current ($I_{ph}$), diode saturation currents ($I_{01}$, $I_{02}$), diode ideality factors ($a_1$, $a_2$), series resistance ($R_s$) and shunt resistance ($R_p$) are extracted using a penalty based differential evolution (P-DE). The proposed method offers several advantages such as: (1) accuracy of solution (2) consistency of solution, (3) speed of convergence and (4) less number of control parameters. The feasibility of the P-DE method has been validated by synthetic and experimental $I-V$ data set of six solar modules of different types (multi-crystalline, mono-crystalline, and thin-film) from various manufacturers. It was found that proposed method is very accurate and converges to the solution very rapidly with high consistency.

Acknowledgments

The authors would like to thank Universiti Teknologi Malaysia and the Ministry of Higher Education, Malaysia for providing the facilities and research grant (Vot No. 68704) to conduct this research.

References


