CHAPTER-4

EXTRACTION OF PV CELL PARAMETERS USING EVOLUTIONARY APPROACHES
4. Extraction of PV Cell parameters using Evolutionary Approaches

4.1 Introduction

The behavior of the photovoltaic cell / module is generally described by its current-voltage characteristic, in which their shapes are greatly dependent on the values of the five parameters $I_{ph}$, $I_0$, $A$, $R_s$, $R_{sh}$, besides their dependence on solar irradiance and module temperature. The Determination of these parameters at any solar irradiance and module temperature conditions makes easy to apply the model for predicting the energy production of photovoltaic module.

The enormous amount of work has been carried out to physically improve the physical performance of solar cells/modules [31-38]. 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 [39-41]. 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 [42] and the two diode model [43]. The latter, despite it is more computationally extensive, is preferable because its I–V characteristics closely resemble the behavior of a physical model [43].

Generally, there are two possible approaches to extract the solar module parameters: (1) the analytical [44–47] and (2) the numerical extraction techniques [48–53]. 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 ($I_{sc}$), open-circuit voltage ($V_{oc}$), 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 on 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 a 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 [48]. 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 tends to be multi-modal, EA methods can be very effective regardless of gradient and initial condition information [39, 54]. To date, various EA techniques such as genetic algorithm (GA) [55–58], Simulated Annealing (SA) [59] and Particle Swarm Optimization (PSO) [30–34] have been employed for parameter extraction of solar module.

In a complex multi-modal problem of high dimension, the computation burden, particularly memory requirement is increased. Furthermore, considering the fact that the 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 [67, 68] 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 with 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 [69].
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.

4.2 Fitness Function / Objective Function Formulation for Parameter Extraction

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 the 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 [72].

The more accurate model, which is 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 of additional parameters. The Figure- 4.1 shows the two diode model, which shall become the basis for the parameter extraction in this work.

![Two Diode Model of a PV cell](image)

Figure- 4.1: Two Diode Model of a PV cell

The output current of the cell can be described as:
IC = IPHC - I01C \left[ \exp \left( \frac{Vc + IRp}{a1VthC} \right) - 1 \right] - I02C \left[ \exp \left( \frac{Vc + IRp}{a2VthC} \right) - 1 \right] - \frac{Vc + IRp}{RpC} \quad \ldots \quad (4.1)

Where, C represents a single cell. Iph is the cell-generated photocurrent; I01 and I02 are the reverse saturation currents of diode 1 and diode 2, respectively. The I02 term is introduced to compensate the recombination loss in the depletion region as described in [43]. Other variables are defined as follows: Vth (kT/q) is the thermal voltage of the solar cell, q is the electron charge (1.60217646 \times 10^{-16} \text{ C}), k is the Boltzmann constant (1.3806503 \times 10^{-23} \text{ J/K}) and T is the temperature of the p–n junction in Kelvin.

Variables a1 and a2 are the diode ideality constants; a1 and a2 represent the diffusion and recombination current component, respectively. Although a greater accuracy can be achieved using this model (compared to the single diode model), it requires the computation of seven parameters, namely Iph, I01, I02, and Rp, Rs, a1 and a2. It is noticed that Figure-4.1 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) [40, 42]. Hence, Equation-4.1, needs to be transformed into a module structure which is effectively the interconnection of solar cells in series or/and parallel (Ns - Np) configuration.

When the cells are arranged in series configuration, only the voltages (Vth, Voc and cell voltage V) are scaled to Ns times. In the same way, the parallel arrangement only increases the corresponding current components in the module by Np times. Accordingly, the output characteristics of a module for Ns _ Np identical solar cells can be represented as follows

\[
\begin{align*}
I &= Np \cdot I_C, \\
I_{ph} &= Np \cdot I_{phC}, \\
I_{01} &= Np \cdot I_{01C}, \\
I_{02} &= Np \cdot I_{02C}, \\
V_{oc} &= Ns \cdot V_{ocC}, \\
V_{th} &= Ns \cdot V_{thC}, \\
R_s &= (Ns/Np) \cdot R_{scC},
\end{align*}
\] .. (4.2)

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

\[
I = I_{ph} \cdot I_{01} \left[ \exp \left( \frac{V + IRp}{a1V_{thC}} \right) - 1 \right] - I_{02} \left[ \exp \left( \frac{V + IRp}{a2V_{thC}} \right) - 1 \right] - \frac{V + IRp}{RpC} \quad \ldots \quad (4.3)
\]
4.3 Objective Function

The extraction performance is evaluated using an objective function, $J$. In this proposed work, $J$ 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:

$$J = \sqrt{\frac{1}{N} \sum_{m=1}^{N} f(I_m, V_m, \phi)^2} \quad \cdots \cdots \cdots \quad (4.4)$$

Where

$$f(I,V,\phi) = I_{ph} - I_{o1} \left[ \exp \left( \frac{V + IR_s}{a_1V_{th}} \right) - 1 \right] - I_{o2} \left[ \exp \left( \frac{V + IR_s}{a_2V_{th}} \right) - 1 \right] - \left( \frac{V + IR_s}{R_p} \right) - 1 \quad \cdots \cdots \cdots \quad (4.5)$$

Where $\phi = [I_{ph}, I_{o1}, I_{o2}, R_s, R_p, a_1, a_2]$ are the seven parameters of the PV module to be extracted. The aim of this extraction procedure is to minimize Equation (4.4) 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 $J$ is desired. It can be seen from Equation (4.4) 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. Two Evolutionary Approaches are evaluated in this thesis one is Genetic Algorithm and the other is Differential Evolution.

4.4 Genetic Algorithm

Genetic Algorithm is essentially inspired by Darwin’s theory of Natural selection and Natural evolution. In GA, the fitness of a population is improved by the processes of selection, crossover and mutation. Genetic algorithms (GA) are a subclass of what is known as evolutionary algorithms [43]. These are computational models that mimic natural evolution in their design and implementation; i.e. they are based on survival of the fittest.

GA is different from conventional search techniques in that they operate on a coded parameter set of the solution, are global in their search; make use of a cost
function that does not involve derivatives and finally employ pseudo-probabilistic rules and not deterministic ones. Genetic algorithms have been used in recent years in solving optimization problems in science and engineering applications \[44, 45\]. Implementation of Gas involves making the following preliminary decisions.

(1) **Solution Encoding:** This involves coding a possible solution (individual) as a string of variables using some alphabet, e.g. binary \{0, 1\}. Individuals are likened to chromosomes and variables to genes. A chromosome (solution) is composed of several genes (variables).

(2) **Evaluation Function:** It determines the fitness score attached to each chromosome (solution). The higher this score, the greater is the chance of an individual (solution) has been selected for reproduction.

(3) **Initial Population Generation:** Generation of the initial population (set of possible solutions) can be random or from known approximate solution(s).

(4) **Selection Criterion:** Methods of selecting individuals for reproduction are numerous and include roulette wheel sampling, stochastic universal sampling, tournament selection, elitism, sigma scaling, rank selection etc.

(5) **Recombination Reproduction:** This is achieved through two genetic operators, namely crossover and mutation. A number of variations of crossover are in use such as single-point, multi-point or uniform crossover. Single-point crossover where binary encoding is used, a locus (bit location) is randomly chosen. Bits after that locus are exchanged between two chromosomes to create two offspring (new solutions). A mutation on the other hand involves randomly flipping some of the bits in a string (chromosome). A very small probability is usually attached to the occurrence of mutation at each bit location (e.g. 0.001). This operation is performed to ensure that new areas of the solution are explored. The Figure-4.2 illustrates a typical GA flow chart.
(6) **Termination criteria:** The algorithm can be terminated if the maximum number of generations (iterations) is achieved, or convergence of the solution is attained (i.e. All solutions yield the same fitness value or differ by less than a specified tolerance). Based on the above decisions, the search algorithm can be invoked.

![Figure- 4.2: Typical GA Flow Chart](image)

**4.4.1 GA implementation for solar cell parameter extraction**

The GA optimization in the proposed work is done using the GA function available in the Matlab. The optimization function is to be minimized for a specified objective value. Initially the starting values; the starting value for the parameters are estimated. An initial generation of 20 chromosomes is randomly generated in the form of vectors with values in a range ± 20% of the starting values. Then, the initial generation is declared as the current generation. Then the characteristics of each individual of the current generation are calculated.

Fitness values of all individuals of the current generation are calculated using .The individuals of the current generation are ordered after the fitness values in descending order. The number of iterations does starts a new generation by maintaining the most efficient 10 individuals of the current generation. The growing
of the new generation is continued using the crossover operation. For crossover two parents, PA and PB are selected using a roulette-type selection, with weights derived from the ordering followed by an intermediary crossover.

After applying crossover, the new generation is completed using the mutation operation. Mutation is applied to a randomly chosen individual, changing one or more parameters. For each modified parameter the value of the descendent maintained in range of ± 20% of the choice parameter. Due to the random character of the initial generation, the final value of fitness and the number of generations required to stabilize fitness have some variation.

The optimization function is saved as a separate entity and called by the inbuilt GA algorithm of Matlab. The initial values, the range in terms of upper bound and lower bound are specified so that the algorithm searches and an effective zone that can result in faster convergence. The algorithm has options to display the result in terms of various parameters like, Genome, plot fitness function value, population distribution, etc; these give an insight into the function of the proposed algorithm shown in Figure4.3.
A Matlab implementation of a GA [46] is used to extract the parameters of a solar cell under illumination. Implementation of the GA for solar cell parameter extraction is evaluated using an objective function, $J$. In this proposed work, $J$ 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:

$$J = \sqrt{\frac{1}{N} \sum_{m=1}^{N} f(I_m, V_m, \phi)^2} \quad \text{…………… (4.6)}$$

Where,

$$f(I,V,\phi) = I_{P_h} - I_{o1} \left[ \exp \left( \frac{V + IR_s}{a_1 V_{th}} \right) - 1 \right] - I_{o2} \left[ \exp \left( \frac{V + IR_s}{a_2 V_{th}} \right) - 1 \right] - \frac{(V + IR_s)}{R_p} - 1 \quad \text{---(4.7)}$$

Where, $\phi = [I_{ph}, I_{o1}, I_{o2}, R_s, R_p, a_1, a_2]$ are the seven parameters of the PV module to be extracted. The aim of the extraction procedure is to minimize equation-(4.6) 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. The synthetic data is 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, $I_{o2} = 2.11 \times 10^{-6}$ A, $R_s = 0.32$ X, $R_p = 200$ X, $a_1 = 1$ and $a_2 = 2$. Using these values, the synthetic I–V curve is generated. The GA program used [46] is developed for maximization of a multivariable function. Thus, $-f_2 (\ldots)$ is used as the cost function and the optimal solution is attained when $f_2 (\ldots) = 0$. The GA parameters are as mentioned below:

- Population type: Double Vector with Populations size = 20
- Fitness scaling: Rank
- Selection function: Stochastic uniform Reproduction
- Crossover fraction: 0.8
- Mutation function: Adaptive
- Generations: 100
4.4.2 Genetic Algorithm execution plots

![Graphs showing Genetic Algorithm execution plots](image)

Figure- 4.3: Execution plots of GA

4.5 Differential Evolution

Differential evolution (DE) is a search and optimization algorithm which was first introduced. It has mainly three 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
the GA. However, unlike GA, which relies on crossover, DE primarily utilizes mutation operation (i.e. Differentiate 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 Gmax. DE works with a population of candidate solutions, PG. 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 XGi as follows:

\[
P_X^G = \begin{array}{c}
(X_i^G)_{i=1,2,\ldots,NP}, \\
G=1,2,\ldots,G_{\text{max}}
\end{array} \quad \ldots \ldots \ldots (4.8)
\]

\[
X_i^G = (X_{j,i}^G)_{j=1,2,\ldots,D} \quad \ldots \ldots \ldots (4.9)
\]

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 by \( j, \) from 1 to D. Figure-4.4 shows the various stages of the DE process in its operational block diagram.

4.5.1 Initialization

In order to begin the optimization process, an initial population of NP D-dimensional real-valued parameter is created. Each vector forms a candidate solution to the multidimensional optimization problem. Initial parameter values are randomly selected within the interval \([XL, XH]\), where \( XL = [X_1, L, X_2, L \ldots XD, L] \) and \( XH \)
= [X1, H, X2, H, . . ., XD, H] are the lower and upper bound of the search space, respectively.

![Diagram showing a vector population generated such that the allowed parameter region is entirely covered.](image)

**Figure-4.5: Initialization of DE**

### 4.5.2 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 are randomly selected in the range [1, NP], such that the indices r1, r2 and r3 are distinct.

The inherent differential mutation in DE exhibits three properties that are crucial for an efficient mutation scheme [71]. First, DE guarantees a distribution with a zero mean value by randomly sampling the difference vectors. 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 have the same orientation as the level lines of the objective function.
4.5.3 Crossover

The donor vector and the target vector are mixed to yield the trial vector in the DE algorithm, two kinds of crossover methods are used i.e. Exponential and binomial (or uniform) [70]. In this work, binomial crossover strategy is used which can be described as:

$$U_{j,i}^G = \begin{cases} V_{j,i}^G, & \text{if } (\text{rand} \leq \text{CR} \text{ or } j = j_{\text{rand}}) \\ X_{j,i}^G, & \text{otherwise} \end{cases} \quad \ldots \ldots (4.10)$$

Where, CR is the Cross over rate.

In 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.

4.5.4 Evaluation and Selection

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

$$X_{i}^{G+1} = \begin{cases} U_{i}^G, & \text{if } J(U_{i}^G) < J(X_{i}^G) \\ X_{i}^G, & \text{otherwise} \end{cases} \quad \ldots \ldots (4.11)$$
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.

**4.6 DE Based Parameter Extraction**

The solar module parameter extraction using the proposed method is evaluated against the synthetic data obtained. Such approach is also utilized by other researchers, for example the synthetic data is calculated using the two diode model with the following seven parameters at STC: \( I_{ph} = 3.8 \, \text{A} \), \( I_{o1} = 4.7 \times 10^{-10} \, \text{A} \), \( I_{o2} = 2.11 \times 10^{-6} \, \text{A} \), \( R_s = 0.32 \, \text{X} \), \( R_p = 200 \, \text{X} \), \( a_1 = 1 \) and \( a_2 = 2 \). Using these values, the synthetic I–V curve is generated. The population size (NP), is chosen to be 70. A typical value of NP ranges between 5D to 10D. The maximum generation number (Gmax) is set to 40,000. Even though, P-DE can converge with much lesser 1000 iterations, this value is selected to be consistent. [60].

The mutation factor (F) is set at 0.8. There is no strict rule on the selection of F but in most of the cases, \( F > 0.4 \) [70]. The crossover rate (CR) is chosen to be 1. A large value of CR intensifies the diversity of the population, thus improving the convergence speed [70]. Moreover, a high value of CR is desirable as the parameters in the model are highly correlated [54]. 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 the number of difference vector and “bin” describes the binomial crossover technique.

**4.7 Implementation of Differential Evolution**

Unlike Genetic Algorithm (GA) where an inbuilt function in the Matlab can be used, Differential Evolution necessitates the implementation of custom built solution for executing optimization using Differential Evolution. The differential evolution algorithm is implemented for the proposed work is starts optimization to minimize the cost returned by the objective function shown in Figure-4.7. The program, which used in the proposed work, uses the following input and output arguments.
4.7.1 Output Arguments

The function bestmem selects a Best population member, the function
best value - Lowest evaluated cost
best Fct Parameters - Structure like input objFctParams containing the best
parameter set.
No of Iterations - Number of iterations done

4.7.2 Input Arguments

DEParams - Structure with parameters for DE.
**ParamDefCell**: Cell specifying the parameters to optimize.

**ObjFctHandle**: Handle to the objective function.

**ObjFctSettings**: Additional settings to be passed

**ObjFctParams**: Struct with initial parameters.

**OptimInfo**: Info about current optimization task. Fields 'title' and 'subtitle' are displayed and included in saved files if existing. No influence on optimization.

The structure **DEParams** contains the following fields.

- **VTR**: "Value to Reach" (set to empty matrix for no VTR).
- **NP**: Number of population members
- **F**: DE-step size F from interval [0, 2].
- **CR**: Crossover probability constant from interval [0, 1]
- **Strategy**: 1 --> DE/best/1/exp (def.) 6 --> DE/best/1/bin
  2 --> DE/rand/1/exp 7 --> DE/rand/1/bin
  3 --> DE/rand-to-best/1/exp 8 --> DE/rand-to-best/1/bin
  4 --> DE/best/2/exp 9 --> DE/best/2/bin
  5 --> DE/rand/2/exp 10 --> else DE/rand/2/bin
- **Maxiter**: Maximum number of iterations.
- **Maxtime**: Maximum time (in seconds) before finishing optimization.
  Set to empty or INF for no time limit
- **Max clock**: Time (as returned by function clock.m) when to finish optimization. Set to empty for no end time.
- **Minvalstddev**: Population is reinitialized if the standard deviation of the cost values in Population is lower than minvalstddev
- **save History**: Save intermediate results.
- **display Results**: Draw graphs for visualization of the optimization result.

The parameters are defined using **ParamDefCell** function which has the following format like Parameter Name; Parameter Ranges which is specify the initial value and the final value, Parameter Quantization.
4.8 Differential Algorithm execution plots

Figure-4.8: Execution plots of DE
Table-4.1: The initial search ranges for the parameters to be extracted given as upper bounds and lower bounds for the algorithms

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Parameter</th>
<th>Minimum Value</th>
<th>Maximum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I&lt;sub&gt;ph&lt;/sub&gt;</td>
<td>5.0</td>
<td>5.8</td>
</tr>
<tr>
<td>2</td>
<td>I&lt;sub&gt;o1&lt;/sub&gt;</td>
<td>1e-6</td>
<td>5e-6</td>
</tr>
<tr>
<td>3</td>
<td>I&lt;sub&gt;o2&lt;/sub&gt;</td>
<td>0</td>
<td>1e-6</td>
</tr>
<tr>
<td>4</td>
<td>a&lt;sub&gt;1&lt;/sub&gt;</td>
<td>1.2</td>
<td>2.0</td>
</tr>
<tr>
<td>5</td>
<td>a&lt;sub&gt;2&lt;/sub&gt;</td>
<td>1.6</td>
<td>2.5</td>
</tr>
<tr>
<td>6</td>
<td>R&lt;sub&gt;s&lt;/sub&gt;</td>
<td>4.8e-3</td>
<td>5.5e-3</td>
</tr>
<tr>
<td>7</td>
<td>R&lt;sub&gt;p&lt;/sub&gt;</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

Table-4.2: Tabulation of Extracted Values using Genetic Algorithm and differential evolution

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Parameter</th>
<th>Ideal Value</th>
<th>Value Extracted using GA</th>
<th>Value Extracted using DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I&lt;sub&gt;ph&lt;/sub&gt;</td>
<td>5.34</td>
<td>5.349</td>
<td>5.35</td>
</tr>
<tr>
<td>2</td>
<td>I&lt;sub&gt;o1&lt;/sub&gt;</td>
<td>1e-6</td>
<td>0.003</td>
<td>0.0025</td>
</tr>
<tr>
<td>3</td>
<td>I&lt;sub&gt;o2&lt;/sub&gt;</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>a&lt;sub&gt;1&lt;/sub&gt;</td>
<td>1.6</td>
<td>1.57</td>
<td>1.7</td>
</tr>
<tr>
<td>5</td>
<td>a&lt;sub&gt;2&lt;/sub&gt;</td>
<td>2</td>
<td>1.92</td>
<td>1.9</td>
</tr>
<tr>
<td>6</td>
<td>R&lt;sub&gt;s&lt;/sub&gt;</td>
<td>5.1e-3</td>
<td>0.254</td>
<td>0.129</td>
</tr>
<tr>
<td>7</td>
<td>R&lt;sub&gt;p&lt;/sub&gt;</td>
<td>160</td>
<td>162</td>
<td>160</td>
</tr>
</tbody>
</table>

4.9 Results and Discussions

In Figure-4.8 shows the P-V Output characteristic curves between GA and DE Methods of extraction for Shell SQ 175 Module. The proposed model takes sunlight irradiance and cell temperature as input parameters and outputs the P-V characteristics under various conditions. A technique for improving the accuracy of the extracted values of solar cell parameters using Genetic algorithm and Differential evolution has-been implemented. It is based on formulating the parameter extraction problem as a search and optimization one. Since determination of the search range is
of importance in applying these techniques, one of the known extraction methods is to be used to determine approximate values for the solar cell parameters.

![Graph](image)

**Figure-4.9:** P-V Output characteristic curves Between GA and DE Methods

**Table-4.3:** Results Comparison of Maximum Power by Genetic Algorithm (GA) & Differential Evaluation (DE)

<table>
<thead>
<tr>
<th>Output Parameters of PV Cells</th>
<th>GA</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extracted Peak Power (Watts)</td>
<td>174</td>
<td>174.8</td>
</tr>
<tr>
<td>Average Number Of Iterations for 25 runs</td>
<td>40</td>
<td>25</td>
</tr>
<tr>
<td>Average Convergence Time for 25 runs</td>
<td>20 seconds</td>
<td>20 seconds</td>
</tr>
</tbody>
</table>
4.10 Summary

This chapter mainly deals with the functions which are necessary to extract the parameters of PV Cells. The objective function is optimized by minimization of the functional fitness value by using GA and DE approaches. The impact of non uniform insolation over PV panels is also clearly considered in the extraction of the parameters. It is concluded that both DE and GA methods converge well with the data sheet values. The extracted peak power of GA and DE are given in Table 4.3. From the results, DE method scores over the GA method in terms of faster response and slower convergence time.