applied sciences

MDPI

Article

Parameter Extraction of Photovoltaic Cells and Panels Using a PID-Based Metaheuristic Algorithm

Aseel Bennagi *, Obaida AlHousrya, Daniel T. Cotfas and Petru A. Cotfas

Department of Electronics and Computers, Faculty of Electrical Engineering and Computer Science, Transilvania University of Brasov, 500036 Brasov, Romania; obaida.alhousrya@unitbv.ro (O.A.); dtcotfas@unitbv.ro (D.T.C.); pcotfas@unitbv.ro (P.A.C.)

* Correspondence: aseel.ben-nagi@unitbv.ro

Abstract

In the world of solar technology, precisely extracting photovoltaic cell and panel parameters is key to efficient energy production. This paper presents a new metaheuristic algorithm for extracting parameters from photovoltaic cells using the functionality of the PID-based search algorithm (PSA). The research includes single-diode (SDM) and double-diode (DDM) models applied to RTC France, amorphous silicon (aSi), monocrystalline silicon (mSi), PVM 752 GaAs, and STM6-40 panels. Datasets from multijunction solar cells at three temperatures (41.5 °C, 51.3 °C, and 61.6 °C) were used. PSA performance was assessed using root mean square error (RMSE), mean bias error (MBE), and absolute error (AE). A strategy was introduced by refining PID parameters and relocating error calculations outside the main loop to enhance exploration and exploitation. A Lévy flight-based zero-output mechanism was integrated, enabling shorter extraction times and requiring a smaller population, while enhancing search diversity and mitigating local optima entrapment. PSA was compared against 26 top-performing algorithms. RTC France showed RMSE improvements of 0.67–2.10% in 3.35 s, while for the mSi model, PSA achieved up to 40.9% improvement in 5.57 s and 22.18% for PVM 752 in 8.52 s. PSA's accuracy and efficiency make it a valuable tool for advancing renewable energy technologies.

Keywords: photovoltaic cells; parameter extraction; PID-based search algorithm; optimization algorithm

Academic Editor: Fabrice Goubard

Received: 25 May 2025
Revised: 28 June 2025
Accepted: 30 June 2025
Published: 1 July 2025

Citation: Bennagi, A.; AlHousrya, O.; Cotfas, D.T.; Cotfas, P.A. Parameter Extraction of Photovoltaic Cells and Panels Using a PID-Based Metaheuristic Algorithm. *Appl. Sci.* **2025**, *15*, 7403. https://doi.org/10.3390/app15137403

1. Introduction

In response to combating climate change and satisfying the requirements of sustainable development, countries are rapidly moving to renewable energy sources like solar power, which is converted using photovoltaic (PV) panels into electric energy [1]. The development of PV systems is crucial for nations to take advantage of the fact that solar energy is accessible in many regions of the world [2]. The European Commission has established goals for renewable energy, which will likely result in improvements to existing technology and the introduction of whole new approaches. The target is for 32% of total energy consumption to come from renewable sources by the end of the decade and a remarkable 100% by 2050 [3]. The move to clean energy sources is becoming critical due to rising environmental concerns, and solar technology has achieved significant advancements and has made continuous attempts to improve photovoltaic technology. In order to maximize the effectiveness of solar power generation systems in response to changing environmental conditions, accurate parameter extraction from PV models is required.

"Parameter extraction" refers to identifying the key electrical parameters such as series resistance, shunt resistance, diode ideality factor, photocurrent, and saturation current in the single- and double-diode photovoltaic models, based on current–voltage (I–V) characteristics. Both analytical methods using complicated equations and metaheuristic algorithms have been used in this field. Both are especially useful when dealing with difficult issues like parameter extraction [4,5].

Analytical methods provide a basic strategy, but they are extremely assumption-dependent. However, metaheuristic algorithms, such as genetic algorithms, perform very well when faced with complicated problems and do not require derivatives [6]. Improvements in clean energy technologies are made possible by combining these analytical methods and metaheuristic algorithms to release photovoltaic systems' full potential [7].

To illustrate the broader effectiveness of metaheuristic techniques in energy optimization problems, the War Strategy Optimization (WSO) algorithm has been successfully applied to optimal power flow in wind-integrated systems, achieving significant reductions in electricity generation costs and voltage fluctuations [8].

Although metaheuristic algorithms have many potential benefits, they can also have limitations, including a slower convergence rate and a higher calculation time [9]. The pros and cons of the various approaches to parameter extraction are addressed in a review by Cotfas et al. [10]. Further information on these powerful optimization methods is provided by later review publications that categorize metaheuristic algorithms and their alternatives [11,12].

While this study centers on photovoltaic systems, the importance of accurate modeling and parameter estimation extends across energy technologies. For instance, Zhao et al. [13] emphasized how structural understanding and redox behavior critically affect performance in sodium-ion batteries—highlighting the universal relevance of precise parameter extraction in optimizing next-generation energy systems.

Despite extensive research in the domain of parameter optimization for photovoltaic systems, key challenges persist that limit the real-world applicability of existing algorithms. One notable issue is the computational intensity required for many of these algorithms to function optimally; they often necessitate large populations, a high number of iterations, and extended computational time [14–18]. This is particularly problematic as photovoltaic systems continue to grow in complexity, raising questions about the scalability and efficiency of current optimization techniques.

This paper introduces a PID-based search algorithm (PSA) that directly addresses these limitations [19]. Inspired by PID control systems, PSA is engineered for efficiency, balancing both population size and the number of iterations to significantly reduce computational time. In tests involving both single- and double-diode models across six types of photovoltaic cells and panels, PSA has demonstrated not only effective scalability but also improved efficiency, making it well-suited for complex, real-world applications. Importantly, PSA requires fewer parameters to be configured, making it well-suited for real-world applications.

PID controllers are widely used in control systems due to their simplicity, robustness, and effectiveness in correcting dynamic system deviations in real time. These properties make them well-suited as a foundation for optimization strategies. In the proposed PSA, the proportional, integral, and derivative components are reinterpreted as metaheuristic tuning mechanisms that adaptively adjust the exploration–exploitation balance. This control-inspired structure helps guide the search process more efficiently toward optimal solutions. By incorporating feedback-driven corrections instead of relying purely on stochastic behavior, PSA offers a more structured and dynamically responsive alternative to traditional metaheuristics, improving both convergence speed and robustness against stagnation.

Another prevalent challenge in the existing literature is the susceptibility of optimization algorithms to local minima [20–22]. Many algorithms, while designed to seek optimal solutions, often converge to suboptimal points due to this limitation. PSA counters this challenge through its inherent resistance to disturbances and fluctuations, which allows it to avoid being stuck in local minima. The PID-inspired search mechanism in PSA ensures a more robust convergence to the global optimum, thereby elevating its efficacy over algorithms that are prone to local minima. Additionally, the PSA features a zero-output mechanism that adapts the search to prevent becoming stuck in local optima by using Lévy flight in a probabilistic manner. In order to improve convergence towards the global optimum in later steps, the Lévy flight enables the PSA to progressively shift focus towards the phase of exploitation despite larger diversity in the search space during earlier steps.

The PSA's performance is evaluated through comparisons with twenty-six leading parameter extraction algorithms, each known for achieving optimal root mean square error (RMSE) in PV applications. However, computational efficiency and execution time scalability are the main weaknesses of the majority of these algorithms. To address these issues, this study significantly reduces execution time and iterations by using smaller population sizes under optimal RMSE. This will be achieved by determining preliminary error values outside the main iterative loop, hence providing a benchmark for further calculations in the incremental PSA. The PSA also applies a mechanism of zero-output with Lévy flight that self-tunes probabilistically to avoid entrapment in local optima, which helps to enhance the efficiency of search and solution quality.

Among the algorithms considered for comparison, the improved version of the snake optimization algorithm (ISOA) achieves a perfect balance between exploring and exploiting, proving possibilities with fewer variables and fewer iterations [23]. Reduced complexity makes it simpler to put into practice. While ISOA has optimized exploration–exploitation balance and accuracy, ISOA requires a high execution time to find the optimal RMSE, which took around 20.6 to 26.9 s per simulation to obtain the optimal RMSE value in the case of SDMs of the RTC France. For the RTC France DDM, it took around 26.87 s to reach the optimal RMSE.

The improved simultaneous heat transfer search algorithm (ISHTS) demonstrates a high level of accuracy in PV parameter estimation because the balance between exploration and exploitation has been enhanced using simultaneous heat transfer along with elitist perturbation strategies, resulting in an optimal RMSE [24]. The algorithm operates efficiently at 50,000 iterations. However, while ISHTS is highly accurate and optimizes the exploration phase to achieve the optimal RMSE value, it requires substantial processing time, with an average execution time of about 23 s per simulation for SDMs and DDMs.

The black widow optimization algorithm (BWOM), which takes its inspiration from the aggressive behaviors of black widow spiders, has certain distinct advantages over other optimization algorithms by using the cannibalism strategy and mating and reproduction mechanism. However, BWOM requires a higher number of search agents in order to obtain the optimal solution [25].

The enhanced generalized normal distribution optimization (NSGNDO) algorithm performs a balance between exploration and exploitation by means of neighborhood search strategies, which realize high accuracy in the PV parameter estimation [26]. In SDMs, NSGNDO achieves an optimal RMSE in 10,000 iterations and completes simulations in about 22 s. In DDMs, it is where NSGNDO yielded an optimal RMSE at about 38 s per run. However, in cases that are very large or highly nonlinear search spaces, the NSGNDO may face a long execution time. That is, the dual neighborhood search strategies make each iteration more complicated, possibly reflecting longer optimization times for high-dimensional problems.

The gaining–sharing knowledge-based algorithm (GSK) proves to be advantageous due to its high accuracy and efficiency when it comes to extracting parameters for solar PV systems [27]. To achieve a good balance between exploration and exploitation, the GSK algorithm employs a pair of crucial stages known as the "junior" and "senior" phases. However, the iteration in the case of SDMs is 30,000, and the iteration in the case of DDMs is 50,000, both of which resulted in a significant amount of computing time.

Improved shuffled complex evolution algorithm (ISCE) has major advantages include a basic yet effective global search, reliable performance, and less unpredictability [28]. This was performed by using the competitive complex evolution and reflection and contraction steps. However, when dealing with complicated situations that have multiple local low points, ISCE takes longer to make progress. ISCE achieved the optimal RMSE in the case of SDMs with 5000 iterations completing in approximately 19 s, while in the case of DDMs, it achieved an optimal RMSE with 10,000 iterations, averaging 24 s per simulation.

The hybrid successive discretization algorithm (HSDA) offers a number of advantages, including increased accuracy and a strong capability for global searching [29]. This is achieved by using vicinity construction around seed solutions, successive discretization techniques, and adaptive adjustment of search radius. However, these benefits require a lot of processing power to fully realize them.

The DIWJAYA algorithm relies on an improved JAYA with an individual weighting scheme, yielding high precision in PV parameter estimation, balancing between exploration and exploitation due to its adaptive weighting mechanism, population mean inclusions, and a Gaussian mutation strategy [30]. For the SDM, DIWJAYA reaches an optimal RMSE in 50,000 iterations, completing simulations in approximately 27 s. For the DDM, it achieves an optimal RMSE with reduced convergence times, averaging 30 s per run. While DIWJAYA's advanced features improve convergence speed and solution quality, its higher iteration demands limit scalability in PV parameter extraction, as the algorithm requires 50,000 iterations to reach the optimal RMSE.

The atomic orbital search algorithm (AOS) is inspired by the quantum atomic model, whose concept is based on the movement of electrons around a nucleus's orbitals. It employs hypothetical orbital layers and photon absorption and emission to find the best possible solution [31]. It is a population-based algorithm, and its performance comes from the fact that it can find the best global solutions quickly and with fewer iterations.

A memory-based improved gorilla troops optimizer (MIGT), demonstrating outperformance of other metaheuristic algorithms by combining the gorilla memory-saving technique and the explorative gorilla with an adaptive mutation mechanism [32]. However, the optimization process requires a high number of iterations in order to obtain the optimal solution, including increased computational load due to the memory-saving technique and potential performance fluctuations in high-dimensional PV models.

While parameter extraction in PV systems has been widely studied using metaheuristic algorithms, the current literature still struggles with balancing accuracy, convergence speed, and scalability. The novelty and contributions of the proposed work for extracting parameters of photovoltaic cells and panels are the following:

- The work is the first that uses the PID-based Search Algorithm (PSA) to extract parameters for both single- and double-diode models for six types of photovoltaic cells and panels (RTC France, amorphous silicon, and monocrystalline silicon photovoltaic cells, PVM 752 GaAs, and STM6-40 photovoltaic panels). Additionally, multijunction solar cells at three temperatures (41.5 °C, 51.3 °C, and 61.6 °C) were utilized.
- The PSA was compared with twenty-six top-performing algorithms and consistently achieved RMSE and MBE levels that are equivalent to or better than the best-performing algorithms. It also proved computationally efficient, requiring a smaller

population and faster extraction times, validating its effectiveness for both PV cells and panels.
- The PSA's performance is considerably improved by adjusting the proportion, integral, differential, and beta (β) parameters, aiding the algorithm in both exploration and exploitation phases.
- Improving the effectiveness of the PSA by calculating current and historical errors of the deviation signal outside the main iterative loop provides crucial beginning values for the incremental PSA and a solid foundation for further computations.
- Implementing a Lévy flight-based zero-output mechanism that allows the PSA to probabilistically adjust to avoid local optima entrapment. This mechanism enables broader search diversity early in the process, whereas λ is decreased over time in order to gradually shift the algorithm's focus on exploitation in later stages.

The following is the plan for this paper: photovoltaic cell and panel models, statistical tests, problem complexity, and the proposed algorithm are described in Section 2. Results and discussions are further discussed in Section 3. Finally, the conclusions and suggestions for further study are presented in Section 4.

2. Materials and Methods

The aim of this paper is to extract PV parameters with maximum accuracy and minimal computation time. The first step involves gathering PV datasets. All datasets used in this study were obtained from previously published experimental studies. This includes data for monocrystalline silicon (mSi) and amorphous silicon (aSi) solar cells [29], multijunction solar cells at three temperatures (41.5 °C, 51.3 °C, and 61.6 °C) [33], as well as data for the RTC solar cell [34], PVM 752 GaAs [29], and STM6-40 panels [35]. Next, the selected models, SDM and DDM, were analyzed. An improved PID-based incremental search algorithm (PSA) was then implemented to improve the PV parameter extraction performance. To validate the results, some of the top algorithms in recent research were selected for comparison using three statistical tests.

2.1. PV Cell Models

PV systems are characterized by their ability to convert sunlight into electrical energy. As a result, knowing the behavior of the photovoltaic cell is critical for assessing its performance. Several models have been utilized to acquire the extracted parameters. The maximum power, short-circuit current, and open-circuit voltage can be obtained using their current–voltage characteristics. The single-diode model (SDM) and double-diode model (DDM) are presented since they are the most well-known models utilized in the challenge of extracting parameters, with the SDM being simpler and requiring fewer parameters to be extracted than the DDM [36,37]. The PSA is developed for optimizing the SDM and DDM parameters for three photovoltaic cells and three photovoltaic panels.

2.1.1. Single-Diode Model (SDM)

The SDM is widely utilized in the academic and research literature to extract the parameters of PV cells and panels. The SDM's equivalent circuit is seen in Figure 1, as the diagram depicts a circuit with photocurrent, a diode, a shunt, and series resistance. The diode is connected in parallel with the shunt resistance, where the shunt resistance is utilized to consider the leakage current, while the series resistance accounts for ohmic losses. Equation (1) describes mathematically the equivalent circuit. It is obtained using Kirchhoff's laws and used to calculate the output current [36].

$$I = I_{ph} - I_{sd}\left[exp\left(\frac{V + IR_s}{nV_t}\right) - 1\right] - \frac{V + IR_s}{R_{sh}} \quad (1)$$

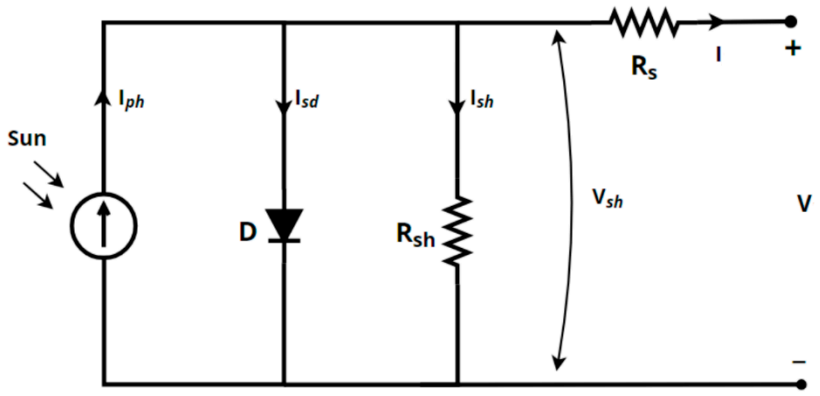

Figure 1. Equivalent circuit of single-diode model.

The photogenerated current is denoted by I_{ph}, whereas the reverse saturation current is denoted by I_{sd}; n represents the ideality factor of the diode, while the shunt resistance and series resistance are represented by R_{sh}, and R_s, respectively. The thermal voltage V_t is calculated as $V_t = k \cdot T/q$, where q is the elementary electric charge with a value of $1.60217646 \times 10^{-19}$ C, and the Boltzmann constant is represented as k with a value of $1.3806503 \times 10^{-23}$ J/K. Lastly, T denotes the temperature of the photovoltaic cell. In order to ensure consistency in the computations for the extraction of parameters for both PV cells and panels, it is advised to use the same values for q and k.

In the case of the SDM, there are five parameters that need to be extracted from the single-diode cell, which are I_{ph}, I_{sd}, n, R_{sh}, and R_s.

2.1.2. Double-Diode Model (DDM)

This model takes into account the effects of recombination current loss in the depletion region by incorporating a second diode in the PV cell. Although this model is more accurate than the SDM, it is also difficult to put into practice. In contrast to the SDM, the DDM contains two parallel-connected diodes in its equivalent circuit, as seen in Figure 2. Equation (2) describes the mathematical formulas required to calculate the output current [37].

$$I = I_{ph} - I_{sd1}\left[exp\left(\frac{V + IR_s}{n_1 V_t}\right) - 1\right] - I_{sd_2}\left[exp\left(\frac{V + IR_s}{n_2 V_t}\right) - 1\right] - \frac{V + IR_s}{R_{sh}}, \quad (2)$$

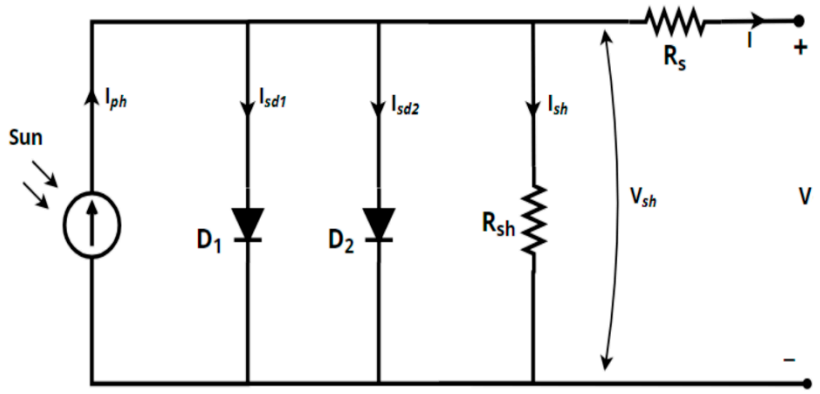

Figure 2. Equivalent circuit of double-diode model.

In this model, the PV cell is characterized by the seven parameters: I_{ph}, I_{sd1}, I_{sd_2}, n_1, n_2, R_{sh}, and R_s. The reverse saturation current and ideality factor of the diffusion mechanism are indicated by I_{sd1} and n_1, whereas the parameters that represent the generation–recombination process are denoted by I_{sd_2} and n_2.

2.1.3. PV Module

In the case of the PV module, the photovoltaic cells are connected in series or in parallel. Equation (3) shows the calculation of the output current when considering a PV module [38].

$$I = N_p\ I_{ph} - N_p\ \ I_{sd1}\left[exp\left(\frac{N_p\ \ V + N_s\ IR_s}{n_1 N_p\ N_s\ \ V_t}\right) - 1\right] - N_p\ \ I_{sd_2}\left[exp\left(\frac{N_p\ \ V + N_s\ IR_s}{n_2 N_p\ N_s\ \ V_t}\right) - 1\right] - \frac{N_p\ V + N_s IR_s}{N_s\ R_{sh}} \quad (3)$$

where N_s is the number of series connections between PV cells, while N_p is the number of parallel PV cell connections. For this study, N_p is set to be equal to 1 because, in general, for industrial modules, the photovoltaic cells are connected in series. The parameters for the PV module will be used to extract both models, which in the SDM requires five parameters to be extracted, and the second bracket from Equation (3) can be considered as 0. On the other hand, in the case of the DDM, to improve the functionality of the module, seven parameters must be extracted.

2.1.4. Statistical Tests

To evaluate the performance of the PSA, an objective function that minimizes the differences between the measured current collected from the dataset and the estimated current obtained from the algorithm must be provided. This study uses the root mean square error, known as RMSE, see Equation (4), as the primary objective function due to its ability to penalize larger errors more significantly than other metrics, making it particularly effective for precise parameter estimation. Moreover, RMSE provides a comprehensive measure of accuracy and reliability, as it considers both the magnitude and distribution of errors.

Additionally, this study uses the mean bias error, known as MBE, shown in Equation (5), and absolute error, known as AE, shown in Equation (6), as another measure of performance to further assess the algorithm's accuracy. These metrics are used to compare the PSA's performance to those of other algorithms in the literature [37].

$$\begin{gathered} RMSE(X) = \sqrt{\frac{\sum_{i=1}^{n}\ (I_{ic} - I_{im}\ )^2}{n}} \\ X(SDM) = I_{ph}\ , I_{sd1},\ R_s,\ R_{sh}, and\ n_1 \\ X(DDM) = I_{ph}\ , I_{sd1},\ I_{sd2}, R_s,\ R_{sh},\ n_1 and\ n_2, \end{gathered} \quad (4)$$

$$MBE = \frac{\sum_{i=1}^{n}\ (I_{ic} - I_{im}\ )}{n} \quad (5)$$

$$AE = \sum_{i=1}^{n} |I_{ic} - I_{im}| \quad (6)$$

where X represents the five parameters from the SDM and the seven parameters from the DDM. n is the total number of measured points, I_{ic} is the algorithm's calculated current, and I_{im} is the actual measured current.

2.2. Problem Complexity

While the models of PV used in the paper, in particular the SDM and DDM, are relatively simple in structure, the multi-objective nature of the parameter optimization problem makes it really challenging, in fact [39]. Among the important factors that the PSA needs to consider in a balanced manner are the accuracy of the parameter extraction, computational efficiency, and also robustness for different population sizes and iteration counts.

Each of the parameters in SDMs and DDMs affects the accuracy of the model, and its calibration should be performed as precisely as possible to reach minimum RMSE. However, the complexity of this task is compounded by the interdependencies between these parameters:

adjusting one parameter can impact the behavior of others, potentially leading to a less accurate solution or longer computational time. Moreover, to achieve optimum performance, different types of PV cells and panels add extra complexity. Each cell type has different characteristics that can differently affect the efficacy of the algorithm. For this, a solution is required that will be able to adapt dynamically to changing conditions and avoid well-known challenges such as local minima and computationally expensive execution times [40].

In regard to this, PSA dynamically adjusts PID factors and a Lévy flight mechanism in terms of population sizes and iteration counts, balancing needs for exploration with the goal of efficient convergence. The following section presents the PSA with design features to address these challenges in PV parameter extraction.

2.3. The Proposed Algorithm

The chosen search algorithm for attaining the goal of the research is the PID-based search algorithm known as PSA [19], which aims to achieve accurate extraction of photovoltaic cell characteristics while balancing the number of iterations and population size to reduce the algorithm's execution time.

Proportional–integral–derivative control, or incremental PID control, illustrates the change from the prior moment's value of the control quantity. Using a recursive process, this difference becomes the next governing parameter. The incremental PID controller generates an output value for the actuator, which in turn controls the controlled object depending on the output value, all in response to the user-established target value. After each adjustment, the actual value of the item being controlled is gathered by a sensor and sent back to the incremental PID controller.

This research makes use of a discretized incremental PID search to complement the metaheuristic method's search mechanism. PID control is designed to provide system stability by efficiently compensating for any disturbances in the object being managed. In this study, it regards each individual as if they were the goal value and the previously best individual in the community as the abstracted target value, which models the PID regulation process to correct the deviance of each individual relative to the optimal individual. The goal of this optimization technique is to improve the efficiency of the population as a whole by using a PID control mechanism, system deviation correction mechanism, adaptive tuning via zero output, and Lévy flight mechanism.

2.3.1. Initialization

The optimization process begins with the initialization of certain variables obtained from the input parameters. These variables include the parameters to be optimized using lower and upper limits, population size, maximum number of iterations, and PID coefficients. This properly constructed setup is critical to producing a population of viable solutions and evaluating their fitness levels.

$$\boldsymbol{x}_{ij} = \left(\boldsymbol{u}_j - \boldsymbol{l}_j\right) \cdot r_1 + \boldsymbol{l}_j \tag{7}$$

The initial solutions (population) are generated randomly within the provided lower bound $\boldsymbol{l}_j$ and upper bound $\boldsymbol{u}_j$. r_1 is a random number between 0 and 1. The fitness of each solution, calculated using the objective function to extract the parameters of the PV panel, is evaluated and stored. The fitness function used is the RMSE, see Equation (4). The algorithm identifies and stores the best solution and its fitness value.

2.3.2. Calculating System Deviations

The algorithm performs the current and previous errors of the deviation signal only within the main iterative loop. To improve performance, an initial calculation of the

deviation signal is also conducted outside the main loop, providing starting values for the incremental PSA and setting the baseline for further calculations. The errors show how the current matrix of solutions from the previous iteration differs from the optimal solution. The primary iterative process begins, which runs for a maximum number of generations. In each generation, the algorithm calculates the fitness function of each solution. If a solution performs better than the previously identified best solution, it is replaced as the best new solution. Then, the algorithm updates the calculation of current and previous errors in each generation, following the PSA. The difference between the best and current solutions forms the basis for updating the errors. The current and previous errors of iteration t are defined as follows:

$$\boldsymbol{e}_k(t) = \boldsymbol{x}^*(t-1) - \boldsymbol{X}(t-1) \tag{8}$$

$$\boldsymbol{e}_{k-1}(t) = \boldsymbol{e}_k(t-1) + \boldsymbol{x}^*(t) - \boldsymbol{x}^*(t-1) \tag{9}$$

where $\boldsymbol{x}^*(t)$ is the best solution vector at iteration t, and $\boldsymbol{X}(t)$ is the matrix of all solutions at iteration t. The deviation signal $\boldsymbol{e}_k(t)$ is calculated as the difference between the historical best individual $\boldsymbol{x}^*(t-1)$ and the current individual $\boldsymbol{x}(t-1)$ at each iteration as shown in Equation (7). This approach eliminates the need for external detecting devices or a specialized controlling system to obtain the deviation signal, as it is derived inherently within the algorithm by tracking the progress of each individual in relation to the best-known solution in the search space. This internal feedback loop allows the PSA to autonomously adjust the control output based on the system's deviation without additional hardware requirements.

Output values, zero output values, and a variable η are calculated. The output values utilize the PSA, with the PID coefficients applied to a weighted sum of the current error, the difference between current and previous errors, and the difference between the current and twice the previous errors plus the two-step errors, as in the following:

$$\Delta\boldsymbol{u}(t) = K_p * \boldsymbol{r}_2 * [\boldsymbol{e}_k(t) - \boldsymbol{e}_{k-1}(t)] + K_i * \boldsymbol{r}_3 * \boldsymbol{e}_k(t) + K_d * \boldsymbol{r}_4 * [\boldsymbol{e}_k(t) - 2\boldsymbol{e}_{k-1}(t) + \boldsymbol{e}_{k-2}(t)] \tag{10}$$

where the PID coefficients K_p, K_i, and K_d are the proportion, integral, and differential adjustment coefficients, respectively; and $\boldsymbol{r}_2$, $\boldsymbol{r}_3$, and $\boldsymbol{r}_4$ are vector matrices containing random numbers between 0 and 1.

The zero output values utilize the Lévy flight function $\boldsymbol{L}$, providing an exploration mechanism to avoid local minima. The variable η, used to balance the influence of the output and zero output values, decreases with progressing generations. The zero output values are defined as follows:

$$\boldsymbol{o}(t) = [cos(1 - t/T) + \lambda \boldsymbol{r}_5 * \boldsymbol{L}] * \ \boldsymbol{e}_k(t) \tag{11}$$

By replacing:

$$\lambda = \left[\frac{ln(T - t + 2)}{ln(T)}\right]^2 \tag{12}$$

$$\boldsymbol{L} = \frac{\boldsymbol{u}\sigma}{|\boldsymbol{v}|^{1/\beta}}; \sigma = \left[\frac{\Gamma(1+\beta) \times sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{(1+\beta)}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}}\right]^{\frac{1}{\beta}} \tag{13}$$

With $\boldsymbol{u}$ and $\boldsymbol{v}$ being matrices of randomly generated numbers coming from a standard normal distribution, β a factor set to 2, and $\boldsymbol{r}_5$ a vector of random numbers from 0 to 1.

With this, the solutions are updated using a combination of output and zero output values, moderated by η:

$$\boldsymbol{x}(t+1) = \boldsymbol{x}(t) + \eta\Delta\boldsymbol{u}(t) + (1 - \eta)\boldsymbol{o}(t) \tag{14}$$

where

$$\eta = \mathbf{r}_6 cos(t/T) \tag{15}$$

$\mathbf{r}_6$ is a vector of random numbers coming from 0 to 1. Finally, the algorithms clip the solutions to ensure they remain within the set bounds.

In the extraction of the PV panel parameters, setting the $K_p = 1.2$, $K_i = 2$, and $K_d = 0.75$ helped the algorithm balance between exploration and exploitation for stable and efficient convergence. The selected moderate K_p value ensures responsiveness without oscillations, while the higher K_i value corrects residual errors, and K_d value provides damping to prevent overshooting.

Additionally, the Lévy flight parameter was set to $\beta = 2$ to allow controlled long-distance jumps, preventing premature convergence while maintaining stability. Empirical tests showed that $\beta < 2$ led to stagnation, while $\beta > 2.5$ caused excessive randomness. This configuration provided optimal accuracy and efficiency, obtaining the best solutions within a few runs.

Although a full grid search or sensitivity analysis was not conducted due to computational constraints, the selected parameters were determined through iterative empirical testing and consistently yielded optimal or near-optimal results across multiple test cases. This parameter set achieved low RMSE and MBE values with efficient convergence over multiple runs.

After running through all the generations, the code returns the best solution and its fitness value, signifying the optimal parameters and the minimum RMSE achieved by the algorithm.

Algorithm 1 and Figure 3 represent the algorithm's pseudo-code and flowchart.

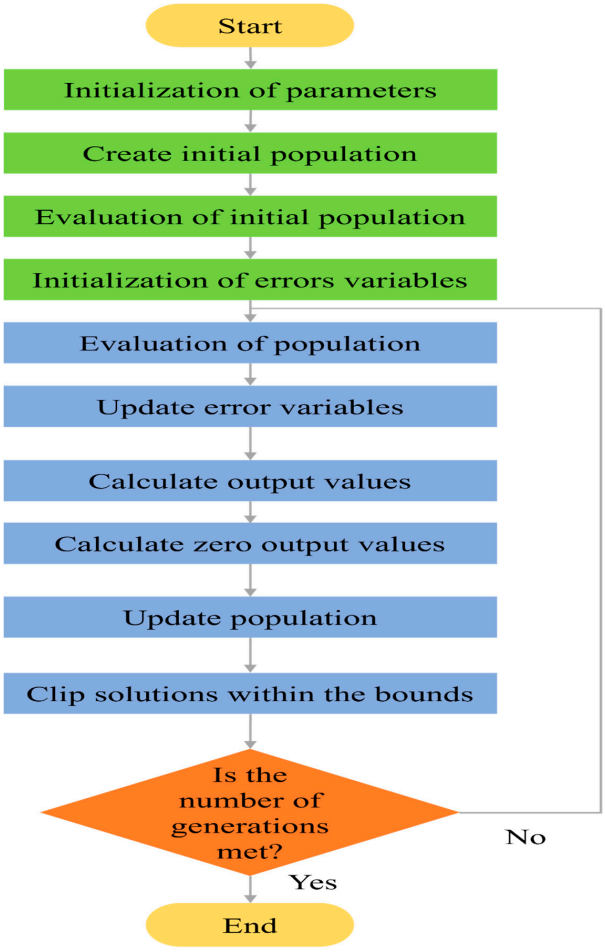

Figure 3. PSA flowchart.

Algorithm 1 PID-Based Search Algorithm

Initialize bounds, population size, number of generations, beta factor, and proportion, integral, and derivative coefficients
Create the initial population using uniform random numbers within the bounds
For every solution in initial population
    Calculate and store fitness value
End For
Identify best solution and its fitness value
Initialize current errors $\boldsymbol{e}_k(1)$ using Equation (8)
Initialize previous errors $\boldsymbol{e}_{k-1}(1)$ as $\boldsymbol{e}_k(1)$ (case for first generation)
Initialize two-step previous $\boldsymbol{e}_{k-2}(1)$ errors as $\boldsymbol{e}_k(1)$ (case for first generation)
For t = 1 to number of generations
    For every solution in population
        Calculate and store fitness value
    End For
    Update best solution and its fitness value
    $\boldsymbol{e}_{k-2}(t) = \boldsymbol{e}_{k-1}(t)$
    Update previous $\boldsymbol{e}_{k-1}(t)$ errors using Equation (9)
    Update current errors $\boldsymbol{e}_k(t)$ using Equation (8)
    Calculate vector of output value $\Delta\boldsymbol{u}(t)$ using Equation (10)
    Calculate λ using Equation (11)
    Calculate Lévy flight function $\boldsymbol{L}$ using Equation (12)
    Calculate zero output value vector $\boldsymbol{o}(t)$ using Equation (11)
    Calculate η matrix using Equation (15)
    Update population using Equation (14)
    Clip set of solutions within the bounds
End For
Return best solution and its fitness value

Due to the PSA's simplicity and ease of implementation, the implementation is built into vs. Code using the Python language version 3.11.9 and runs on a PC with an Intel Core i9 CPU and Windows 11 64-bit operating system. It is straightforward to use and offers stable performance.

3. Results and Discussion

The strength, efficiency, and accuracy of the algorithm are major factors in the growth and evolution of green energy sources. The extraction of exact parameters from photovoltaic cells and panels is a powerful resource that may help advance the renewable energy sector and lead us to a more sustainable future for the world.

To achieve accurate parameter extraction for PV cells and modules, this section determines how successfully the suggested PSA works. RTC France, amorphous silicon (aSi) and monocrystalline silicon (mSi) photovoltaic cells and panels, multijunction cells (MJS), PVM 752 GaAs, and STM6-40 photovoltaic panels were all used in the experiment. Based on a measured dataset, these PV devices were evaluated using the PSA for SDMs and DDMs. The PSA shows better or at least equivalent results compared to twenty-six of the best-performing algorithms.

Four metrics, including RMSE, MBE, STD, and AE, have been evaluated by comparing them with other top metaheuristic algorithms that have been used to extract the parameters of PV cells and panels to evaluate the performance of the PSA. In terms of improving

the manufacturing process for manufacturers, it is crucial to achieve ideal solutions for each device while decreasing computation time. The performance results of the twenty-six comparison algorithms were taken directly from their original studies, using the same datasets and evaluation metrics. While our PSA was fine-tuned for optimal performance, the benchmark results also reflect tuning performed in their respective original papers.

The iteration and population size used for all the tests are 1100 iterations and 100 population sizes. It is significant to know that the PSA leads to better RMSE solutions as the number of iterations increases. This, however, leads to increased computational time. As a result, there is a trade-off between producing better outcomes and requiring more processing resources. In the end, the best combination that led to finding the optimal RMSE value from the first run was 100 population and 1100 iterations. With this iteration and the population size, the PSA provided positive results from its first run. The algorithm's exploration step was much improved and effectively prevented convergence to local minima, which would otherwise result in less-than-ideal results. All execution times reported in this study include both the initialization phase and the full optimization loop to reflect the complete computational cost of the PSA.

The parameters of the PV cells are provided with their upper and lower bounds in Table 1, and in Table 2, the lower and upper bounds applied to the parameters of the PV modules are shown.

Table 1. PV cells lower and upper bounds.

Parameter	RTC Cell		mSi Cell		aSi Cell		MJSC Cell	
	LB	UB	LB	UB	LB	UB	LB	UB
I_{ph} (A)	0	1	0	1	0	1	0.001	0.2
I_{sd1}, I_{sd2}(A)	10^{-12}	10^{-5}	10^{-12}	10^{-5}	10^{-12}	10^{-5}	10^{-40}	10^{-5}
$R_s(\Omega)$	0	1	0	1	0	0.5	0.001	3
$R_{sh}(\Omega)$	0	100	0	200	0	1000	0	100,000
n_1, n_2	1	3	1	3	1	5	1	4

Table 2. PV modules lower and upper bounds.

Parameter	PVM		STM6-40	
	LB	UB	LB	UB
I_{ph} (A)	0	1	0	2
I_{sd1}, I_{sd2}(A)	10^{-12}	10^{-5}	10^{-12}	10^{-5}
$R_s(\Omega)$	0	1	0	1
$R_{sh}(\Omega)$	0	1000	0	1000
n_1, n_2	1	10	1	2×36

3.1. RTC France Cell

The first solar cell being examined is the 57 mm diameter RTC France silicon commercial cell. The majority of studies that extract photovoltaic cell parameters and apply metaheuristic algorithms for the SDMs and DDMs include this specific cell, which has been extensively examined. Although the changes could be slight, they are an essential way to show the new algorithm's effectiveness and accuracy. At 1000 W/m^2 and 33 °C, the RTC silicon photovoltaic cell's I–V characteristic is measured. The associated dataset [34] is shown in Table S1, and it contains values for the current calculated using the SDMs and DDMs, along with their respective errors.

Both the SDM's and DDM's execution times of 3.3512 and 6.5292 s, respectively, are shown in Table 3, along with the RTC photovoltaic cell parameters retrieved using the PSA. The performance of the proposed algorithm was compared using RMSE, MBE, and STD with BWOA, ISHTS, NSGNDO, ISOA, AOS, HSDA, GSK, DIWJAYA, memory-based improved gorilla troops optimizer (MIGTO) [32], growth optimizer (GO) [41], improved marine predators algorithm (IMPA) [22], improved moth flame algorithms with local escape operators (IMFOL) [42], modified stochastic fractal search algorithm (MSFS) [43], a logistic chaotic JAYA algorithm (LCJAYA) [44], hybrid backtracking search optimization algorithm with differential evolution (DE/BSA) [45], leveraging the opposition-based exponential distribution optimizer (OBEDO) [46], and improved differential evolution algorithm (DE) [47]. As all of them produce the same RMSE value for the SDM, there is almost no difference in RMSE between the PSA and other top-performing algorithms. The RMSE value for the DDM, however, has improved to 0.67% when compared to BWOA and 1.15% when compared to ISOA, which demonstrates that PSA obtained the best RMSE value. Despite this, the RMSE for the DDM produced with the ASO, MSFS, MIGTO, IMFOL, and GSK algorithms is higher than that obtained with the PSA.

In the case of the SDM, because the RMSE values derived from the majority of algorithms are essentially identical, the difference in the retrieved parameters is negligible. This is due to the use of the best algorithms from the specialized literature. Algorithms such as DE and LCJAYA, on the other hand, give limited results because of their tendency to converge to local minima, preventing them from discovering optimal solutions and giving inferior results when compared to other top-performing algorithms for both models. This means that not all the algorithms can obtain the minimum value for RMSE. In this instance, however, the PSA surpasses other algorithms across all statistical tests, as shown in Table 3. When all retrieved parameters are analyzed for both models using the PSA method, considerable differences can be detected when compared to other algorithms.

In comparison to BWOA, the reverse saturation alteration for the diffusion mechanism is less in the DDM, varying by around 0.28%, but the difference for generation–recombination is high. As a result, it can be concluded that the DDM can accurately match the SDM in terms of parameter extraction and accuracy. Furthermore, the evaluation of all RTC photovoltaic cell parameters and the RMSE, MBE, and STD demonstrates the PSA's capability to find the optimal solution and avoid hitting local minima.

In comparison to ISOA, both algorithms achieved the same RMSE of 9.8602×10^{-4}; however, PSA yielded a significantly lower mean bias error (MBE) of 9.58×10^{-11} compared to ISOA's -1.57×10^{-8}, indicating greater numerical stability. For the DDM, PSA achieved a better RMSE (9.7078×10^{-4}) than ISOA (9.8215×10^{-4}), as well as a notably reduced MBE (2.31×10^{-9} vs. -3.62×10^{-8}), confirming its consistency and precision.

For both the SDMs and DDMs, Figure 4 shows how the PSA's estimated and observed I–V line up, and Figure 5 displays the P–V characteristics.

The absolute errors that were calculated by the algorithm, which are the differences between the measured current and the calculated current for both models, are displayed in Figure 6.

Table 3. RMSE of the RTC photovoltaic cell and the extracted parameters for different algorithms.

Algorithm	Model	I_{ph} (A)	I_{sd1}(μA)	n_1	R_s(Ω)	R_{sh}(Ω)	I_{sd2}(μA)	n_2	RMSE	MBE	STD
PSA	**SDM**	**0.760775530**	**0.323020790**	**1.48118358**	**0.0363770929**	**53.7185235**	**-**	**-**	$\mathbf{9.8602 \times 10^{-4}}$	$\mathbf{9.58307692 \times 10^{-11}}$	$\mathbf{1.005550 \times 10^{-3}}$
BWOA	SDM	0.760775530	0.323020844	1.48118360	0.0363770922	53.7185277	-	-	9.8602×10^{-4}	$9.6050008 \times 10^{-11}$	1.005550×10^{-3}
ISOA	SDM	0.7607751	0.3236678	1.481385	0.0363688	53.76822	-	-	9.8602×10^{-4}	$-1.57692307674 \times 10^{-8}$	1.005557×10^{-3}
ISHTS	SDM	0.76078	0.32302	1.48118	0.03638	53.71853	-	-	9.8602×10^{-4}	$-3.71615384615 \times 10^{-6}$	1.005600×10^{-3}
NSGNDO	SDM	0.76078	0.32302	1.48118	0.03638	53.71852	-	-	9.8602×10^{-4}	$-3.71769230769 \times 10^{-6}$	1.005599×10^{-3}
DE/BSA	SDM	0.760775	0.32302080	1.48118359	0.0363770	53.71852464	-	-	9.8602×10^{-4}	$-4.71538461539 \times 10^{-7}$	1.005549×10^{-3}
DIWJAYA	SDM	0.760775	0.323021	1.481183	0.036377	53.718525	-	-	9.8602×10^{-4}	$-1.75576923076 \times 10^{-6}$	1.005551×10^{-3}
OBEDO	SDM	0.7608	0.323	1.4812	0.0364	53.7185	-	-	9.8602×10^{-4}	$5.67684615384 \times 10^{-5}$	1.009594×10^{-3}
GO	SDM	0.760776	0.323	1.481183	0.036377	53.718745	-	-	9.8602×10^{-4}	$1.24500000000 \times 10^{-5}$	1.005708×10^{-3}
IMPA	SDM	0.7607755303	0.323020816	1.48118359	0.03637709258	53.71852391	-	-	9.8602×10^{-4}	$-3.84615385140 \times 10^{-9}$	1.005549×10^{-3}
AOS	SDM	0.760776	0.322689	1.48108	0.036379	53.69001	-	-	9.8603×10^{-4}	$9.05384615393 \times 10^{-7}$	1.005559×10^{-3}
IMFOL	SDM	0.760775530	0.323020843	1.48118360	0.0363770924	53.7185307	-	-	9.8602×10^{-4}	$-4.73712949209 \times 10^{-8}$	1.005549×10^{-3}
HSDA	SDM	0.7607758	0.32301653	1.4811823	0.03637708	53.71452	-	-	9.8602×10^{-4}	1.42971×10^{-8}	1.005550×10^{-3}
MSFS	SDM	0.76077553	0.32302082	1.48118359	0.03637709	53.71852461	-	-	9.8602×10^{-4}	$-5.38461538498 \times 10^{-9}$	1.005549×10^{-3}
GSK	SDM	0.7608	0.3231	1.4812	0.0364	53.7227	-	-	9.8602×10^{-4}	$-5.47846153845 \times 10^{-6}$	1.006553×10^{-3}
MIGTO	SDM	0.76077552	0.32302083	1.4811835	0.03637709	53.7185298	-	-	9.8602×10^{-4}	$-1.96923076912 \times 10^{-7}$	1.005548×10^{-3}
LCJAYA	SDM	0.7608	0.3230	1.4819	0.0364	53.7185	-	-	2.7630×10^{-3}	$1.42552076923 \times 10^{-3}$	2.413762×10^{-3}
DE	SDM	0.7607	0.3209	1.4709	0.0363	54.1134	-	-	3.6249×10^{-2}	$-1.98973134615 \times 10^{-2}$	3.089350×10^{-2}
PSA	**DDM**	**0.760800404**	**0.244983250**	**1.45544096**	**0.0368773038**	**58.5598710**	**10.0**	**3.00000000**	$\mathbf{9.7078 \times 10^{-4}}$	$\mathbf{2.30769229980 \times 10^{-9}}$	$\mathbf{9.900088 \times 10^{-4}}$
BWOA	DDM	0.760788874	0.231811940	1.45130774	0.0368625221	56.9731667	2.8436171	2.41335462	9.7738×10^{-4}	2.61366×10^{-8}	9.967384×10^{-4}
ISOA	DDM	0.7607645	0.29116895	1.47189999	0.0365028	55.016889	0.7207339	2.30889999	9.8215×10^{-4}	$-3.61538461445 \times 10^{-8}$	1.001607×10^{-3}
GO	DDM	0.760791	0.238	1.455712	0.036658	54.977810	0.574	1.968428	9.830×10^{-4}	$2.05047692307 \times 10^{-4}$	1.048820×10^{-3}
ISHTS	DDM	0.76078	0.74472	1.99998	0.03674	55.48374	0.22659	1.45125	9.8248×10^{-4}	$-5.16153846145 \times 10^{-7}$	1.001974×10^{-3}
NSGNDO	DDM	0.76078	0.74935	2.00000	0.03674	55.48544	0.22597	1.45102	9.8248×10^{-4}	$8.98538461538 \times 10^{-6}$	1.002053×10^{-3}
DE/BSA	DDM	0.760781	0.749350	1.4510	0.036740	55.485447	0.225973	1.9999	9.8248×10^{-4}	$-4.35068028846 \times 10^{-1}$	6.597302×10^{-1}
DIWJAYA	DDM	0.760781	0.225973	1.451016	0.0367404	55.485448	0.749355	1.999999	9.8248×10^{-4}	$-6.45384615380 \times 10^{-7}$	1.001943×10^{-3}
IMPA	DDM	0.7607810791	0.22597417046	1.451016729	0.036740430739	55.485442507	0.7493481	1.99999999	9.8248×10^{-4}	$3.84615394821 \times 10^{-10}$	1.001941×10^{-3}
OBEDO	DDM	0.7608	0.231	1.4529	0.0367	55.3995	0.708	2.0000	9.8250×10^{-4}	$9.80126923077 \times 10^{-5}$	1.007371×10^{-3}

Table 3. *Cont.*

Algorithm	Model	I_{ph} (A)	I_{sd1}(μA)	n_1	R_s(Ω)	R_{sh}(Ω)	I_{sd2}(μA)	n_2	RMSE	MBE	STD
AOS	DDM	0.76078	0.22732	1.45151	0.036717	55.3951	0.72895	1.99879	9.8745×10^{-4}	$6.28773076923 \times 10^{-5}$	1.004967×10^{-3}
IMFOL	DDM	0.760779169	0.766320100	2.00000000	0.0367305550	55.6567344	0.2251486	1.45077880	9.8252×10^{-4}	$-2.61538461624 \times 10^{-8}$	1.001984×10^{-3}
HSDA	DDM	0.76079412	0.2267032	1.451	0.03676287	55.398385	0.8267308	2.0303627	9.8217×10^{-4}	$3.30230769231 \times 10^{-6}$	1.001620×10^{-3}
MSFS	DDM	0.76078108	0.74934896	2.00000000	0.03674043	55.48543892	0.2259740	1.45101668	9.8248×10^{-4}	$3.80769230855 \times 10^{-8}$	1.001942×10^{-3}
GSK	DDM	0.7608	0.2595	1.4627	0.0366	54.9330	0.4791	1.9983	9.8248×10^{-4}	$4.06253846153 \times 10^{-5}$	1.004134×10^{-3}
MIGTO	DDM	0.7607810	0.225974	1.4510167	0.0367404	55.485441	0.7493493	1.9999999	9.8248×10^{-4}	$2.69230769009 \times 10^{-9}$	1.001942×10^{-3}
LCJAYA	DDM	0.7608	0.22596	1.4518	0.0367	55.4815	0.74640	2.000	3.0123×10^{-3}	$1.60520961538 \times 10^{-3}$	2.599509×10^{-3}
DE	DDM	0.7605	0.423220	1.87579	0.02061	51.9345	0.18726	1.43602	3.0866×10^{-2}	$1.18194761538 \times 10^{-2}$	2.907886×10^{-2}

The results of PSA algorithm are highlighted in bold.

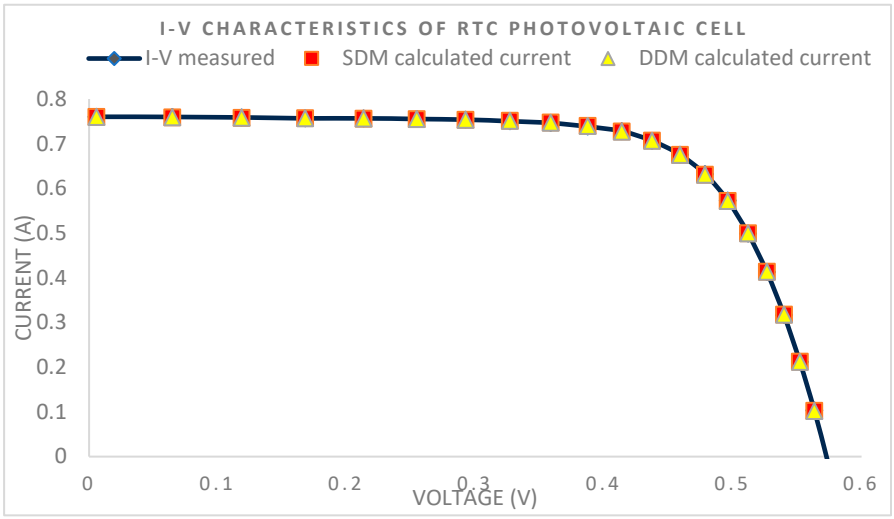

Figure 4. RTC SDM and DDM I–V characteristics.

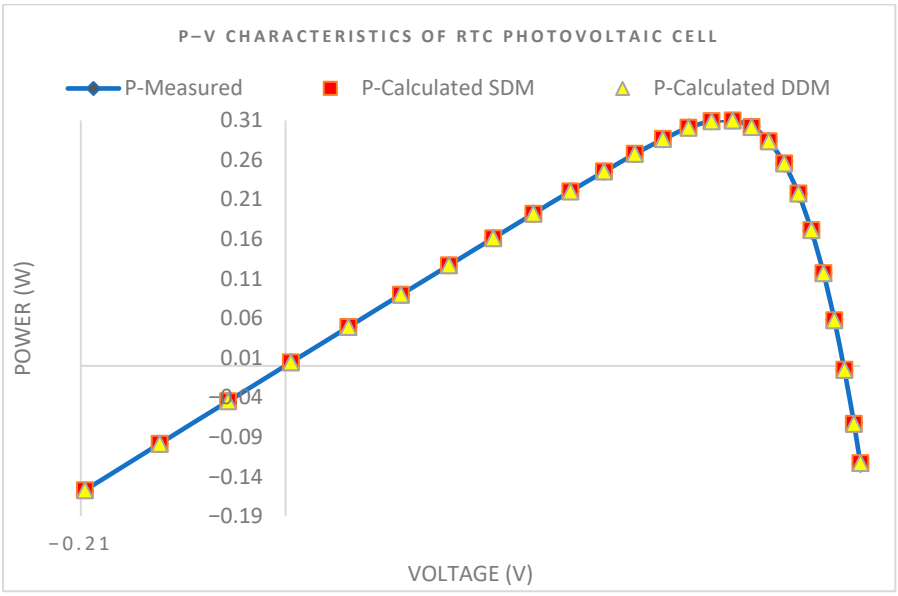

Figure 5. RTC SDM and DDM P–V characteristics.

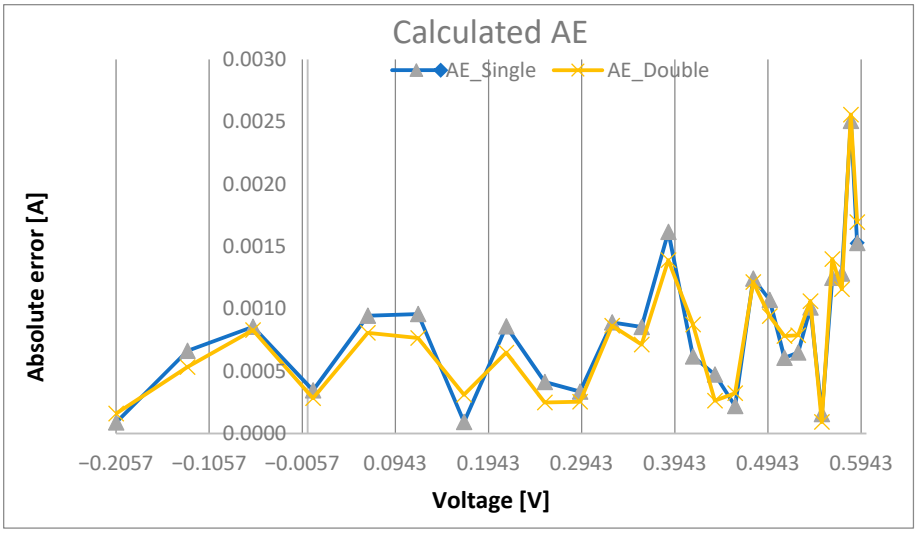

Figure 6. RTC SDM and DDM calculated absolute error.

The PSA's convergence curves for both models are shown in Figure 7, where the maximum iteration was 1500 and the population size was 100. The figure displays the iteration at which we began obtaining the least RMSE value, which is around 250 iterations at population 45 in the case of the SDM and population 93 in the case of the DDM. Additionally, shaded regions representing ± standard deviation across the iterations are included to illustrate the algorithm's stability and convergence behavior over time. These bands reflect the variation in RMSE values and help assess the degree of fluctuation in the optimization process. Narrower bands indicate more consistent performance, while wider bands suggest increased variability.

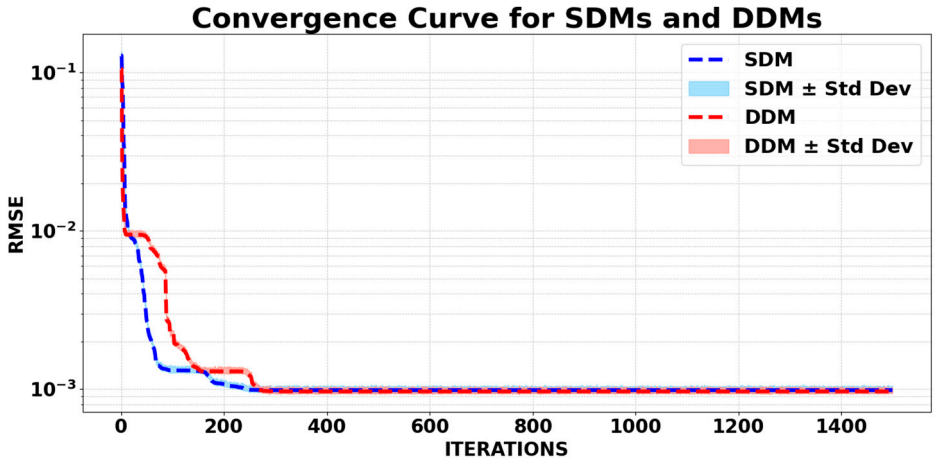

Figure 7. RTC SDM and DDM convergence curve.

Another study focuses on how the RTC photovoltaic cell's RMSE and computation time change with iteration and population size. To find the best population size and iteration rate combination for minimizing the RMSE value, two studies were carried out.

The population size ranged from 20 to 100, and both models were examined in the first study with a set number of iterations of 100. However, this method did not produce the best results because the smallest RMSE value was reached when the population size and iteration were both set to 100. The results of this analysis produced a SDM with an RMSE value of 2.0060×10^{-3} and a compute time of 0.3747 s. With a computation time of 0.5862 s, the DDM's minimal RMSE value was 4.5336×10^{-3}.

In the second study, the number of iterations was changed by a step of 200 while keeping the population size fixed. This method made it possible to determine the algorithm's lowest possible RMSE value. The optimum RMSE solution for the SDM needed a population size of 50 and 1100 iterations, which took around 2.056 s to compute. With a population size of 50 and 1500 iterations, the DDM took around 4.632 s to compute. Table 4 displays the analysis for the SDM, whereas Table 5 displays the analysis for the DDM.

Another experiment was conducted to evaluate the PSA's effect on CPU execution time across different processors, as presented in Table 6. The SDM and DDM were tested on Intel Core i9 and AMD Ryzen 5 5600H CPUs. The Intel Core i9 outperformed the AMD Ryzen 5, with execution times of 1.9865 s (SDM) and 4.5221 s (DDM) compared to 3.5441 s (SDM) and 7.6632 s (DDM) for the AMD. The differences are due to the Intel CPU's better single-threaded performance, optimized instruction sets, and superior thermal management, particularly for the more computationally demanding DDM.

Table 4. RTC iteration vs. population size SDM case.

Number of Runs	Population Size	Number of Iterations	RMSE	Run Time (s)
5	20	100	9.1922×10^{-3}	0.0890
	30	100	5.4496×10^{-3}	0.1280
	50	100	4.4394×10^{-3}	0.1901
	60	100	7.3220×10^{-3}	0.2262
	80	100	6.6887×10^{-3}	0.3345
	100	100	2.0060×10^{-3}	0.3747
	50	300	6.3211×10^{-3}	0.5690
	50	500	3.4615×10^{-3}	0.9510
	50	700	1.3510×10^{-3}	1.3144
	50	900	9.8710×10^{-4}	1.6930
	50	1100	9.8602×10^{-4}	2.0560
	50	1300	9.8602×10^{-4}	2.4502
	50	1500	9.8602×10^{-4}	2.9132

Table 5. RTC iteration vs. population size DDM case.

Number of Runs	Population Size	Number of Iterations	RMSE	Run Time (s)
5	20	100	6.6931×10^{-3}	0.1221
	30	100	8.6845×10^{-3}	0.1806
	50	100	8.7694×10^{-3}	0.2942
	60	100	6.9107×10^{-3}	0.3561
	80	100	8.6446×10^{-3}	0.4728
	100	100	4.5336×10^{-3}	0.5862
	50	300	4.2502×10^{-3}	0.8779
	50	500	1.7329×10^{-3}	1.4524
	50	700	1.4385×10^{-3}	2.0343
	50	900	9.8620×10^{-4}	2.6404
	50	1100	9.8348×10^{-4}	3.1974
	50	1300	9.7632×10^{-4}	3.7938
	50	1500	9.7078×10^{-4}	4.6326

Table 6. RTC CPU execution time.

Number of Runs	Model	Population Size/ Number of Iterations	RMSE	Intel Core i9 CPU Execution Time (s)	AMD Ryzen 5 5600H CPU Execution Time (s)
5	SDM	100/1100	9.8602×10^{-4}	1.9865	3.5441
	DDM	100/1100	9.7078×10^{-4}	4.5221	7.6632

3.2. aSi Cell

The amorphous silicon photovoltaic is the topic of discussion in the second analysis. It was evaluated at a specified 1000 W/m^2 irradiance and 25 °C temperature, and so the I–V characteristic was measured. The result observed [29], which includes the current values derived using the SDMs and DDMs, including their associated errors, can be found in Table S2.

Table 7 displays the characteristics of the aSi photovoltaic cell determined from the PSA for both the SDMs and DDMs. The SDM analysis took 3.3512 s to execute, whereas the DDM took 6.5292 s. Statistical tests, RMSE, MBE, and STD were used to evaluate algorithm performance, which included algorithms such as BWOA, ISOA, and HSDA.

The PSA and BWOA algorithms produced identical RMSE values for both models, showing only minor variations. A comparison of the RMSE value with ISOA for the SDM revealed a small improvement of roughly 0.01%, whereas the improvement for the DDM was more considerable at 3.88%. For both models, the RMSE produced by the HSDA was higher than that obtained by the PSA.

When compared to BWOA, the RMSE findings for the SDMs and DDMs are essentially equal, showing insignificant changes in the obtained parameters. However, substantial differences arise when comparing the DDM to other algorithms such as ISOA and HSDA. Table 7 shows how the PSA beats alternative techniques for the DDM across all statistical tests. As all collected parameters for both models are analyzed using the PSA technique, substantial differences are shown when compared to other algorithms.

Compared to ISOA, the DDM has more changes in reverse saturation for recombination and generation processes as well as changes in series and shunt resistance. This shows that the DDM excels at extracting parameters reliably and with accuracy. Furthermore, the evaluation of all aSi solar cell parameters and the RMSE demonstrates the PSA's capacity to find the optimal solution and avoid entering local minima.

Using the PSA approach, the I–V in Figure 8 and the P–V in Figure 9 shows features of the SDMs and DDMs aligned with the actual values.

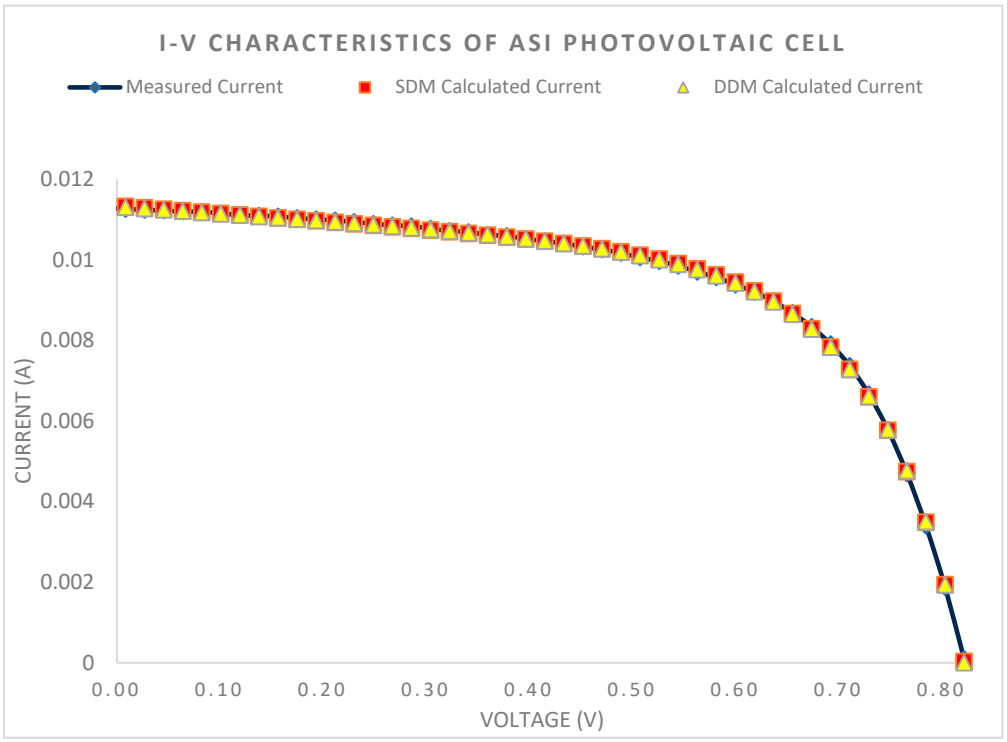

Figure 8. aSi SDM and DDM I–V characteristics.

Table 7. RMSE of the aSi photovoltaic cell and the extracted parameters for different algorithms.

Algorithm	Model	I_{ph} (A)	I_{sd1}(μA)	n_1	R_s(Ω)	R_{sh}(Ω)	I_{sd2}(μA)	n_2	RMSE	MBE	STD
PSA	**SDM**	**0.0113447220**	**0.734937291**	**3.368417**	$\mathbf{1.82123 \times 10^{-14}}$	**523.153423**	**-**	**-**	$\mathbf{4.6123 \times 10^{-5}}$	$\mathbf{-2.159066441 \times 10^{-12}}$	$\mathbf{4.604524 \times 10^{-5}}$
BWOA	SDM	0.0113447220	0.734937259	3.368417	0.01	523.153416	**-**	**-**	4.6123×10^{-5}	$2.159066442 \times 10^{-12}$	4.604524×10^{-5}
ISOA	SDM	0.011344	0.741543	3.37156	1.1657×10^{-9}	524.336	**-**	**-**	4.6127×10^{-5}	$-3.669565217 \times 10^{-7}$	4.605594×10^{-5}
HSDA	SDM	0.011347	0.7047542	3.353834	0.040283	520.0651	**-**	**-**	4.6194×10^{-5}	4.13577×10^{-7}	4.610007×10^{-5}
PSA	**DDM**	**0.0113317349**	**5.50742804**	**5.0000**	**0.500000**	**600.667268**	**0.0502610**	**2.71088634**	$\mathbf{4.0943 \times 10^{-5}}$	$\mathbf{3.6782608695 \times 10^{-9}}$	$\mathbf{4.095979 \times 10^{-5}}$
BWOA	DDM	0.0113439013	3.499752108	4.999279	0.3800093610	554.641471	0.1561147	2.95719676	4.0943×10^{-5}	$-9.393959725 \times 10^{-9}$	4.248453×10^{-5}
ISOA	DDM	0.0113334	0.164276	2.97444	0.347809	570.845	3.66266	4.99379	4.2596×10^{-5}	2.63015×10^{-8}	4.258692×10^{-5}
HSDA	DDM	0.01134914	0.222195	3.079462	0.1987938	528.1008	1.0605304	4.0991152	4.4973×10^{-5}	$-4.004347826 \times 10^{-7}$	4.486736×10^{-5}

The results of PSA algorithm are highlighted in bold.

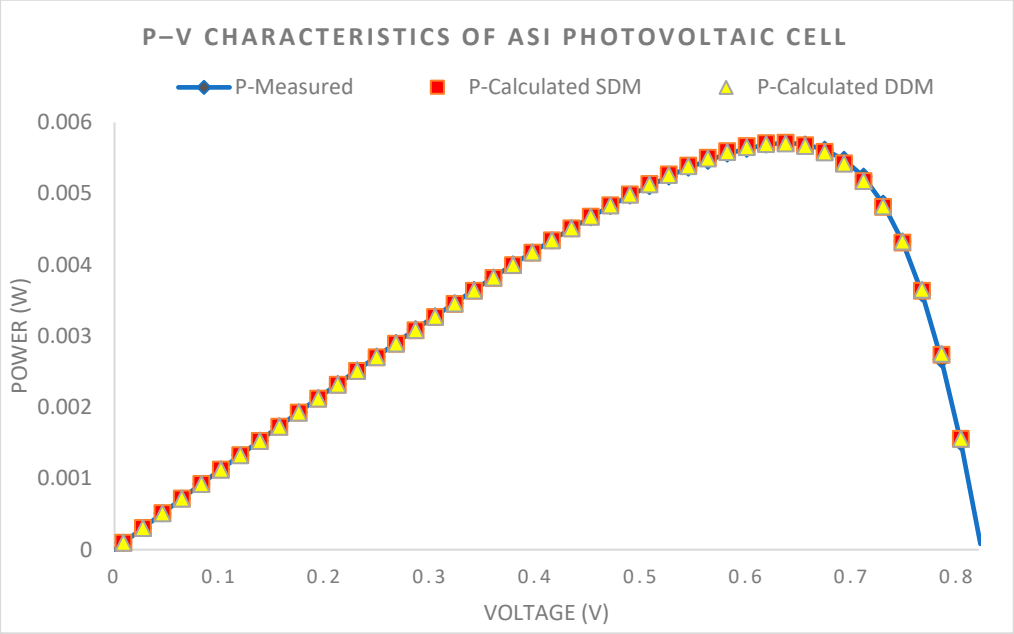

Figure 9. aSi SDM and DDM P–V characteristics.

Disparities between the observed current and the estimated current for both models are shown in Figure 10 as absolute errors, determined by the algorithm.

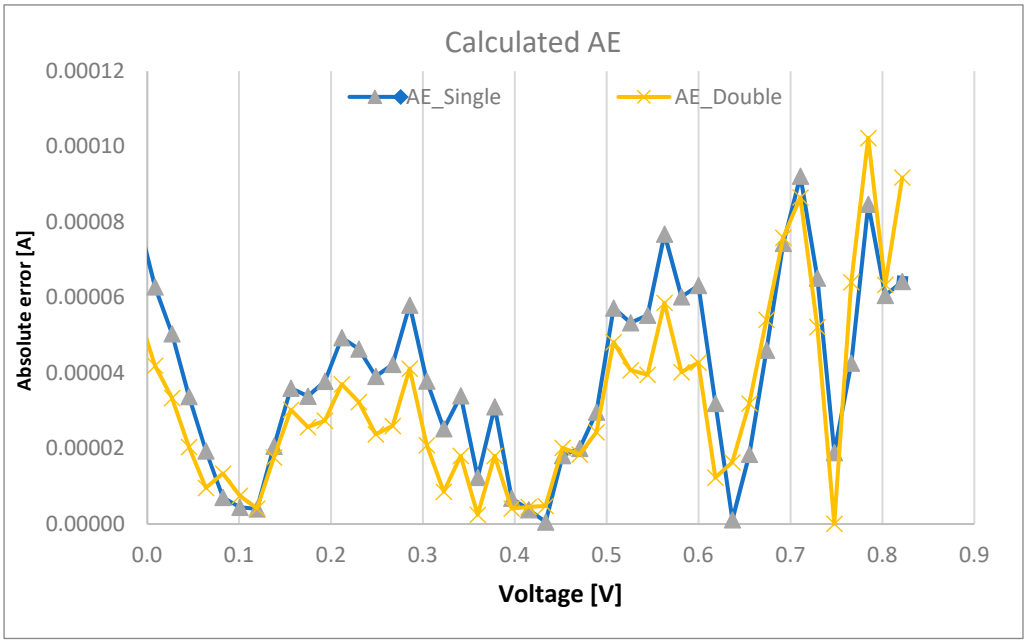

Figure 10. aSi SDM and DDM calculated absolute error.

The convergence curves of the PSA with a maximum of 1500 iterations and a population size of 100 are shown in Figure 11. On the graph, the minimum RMSE value is seen to have been reached at roughly 150 iterations at population 37. Additionally, the figure includes shaded regions representing the ± standard deviation of RMSE values across the iterations. These bands serve to highlight the consistency and stability of the PSA throughout the optimization process. A narrow-shaded area indicates that the RMSE values remained closely clustered, reflecting robust convergence behavior, whereas wider regions would signify greater variability and less consistent performance.

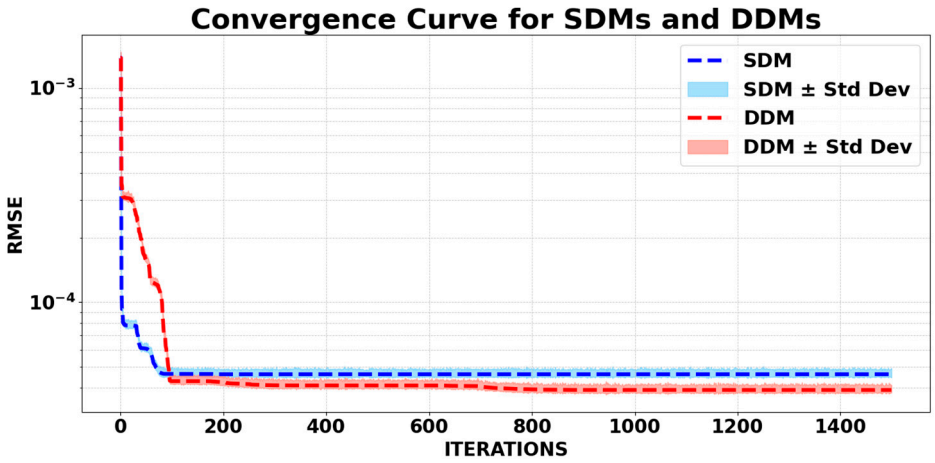

Figure 11. aSi SDM and DDM convergence curve.

In addition to the RTC, the aSi photovoltaic cell benefited from the research that was performed. The original study looked at both models with a fixed number of iterations set to 100, and the population size was between 20 and 100. However, the highest results were obtained by setting both the population size and the number of iterations to 100; therefore, this strategy was not optimal. The investigation showed that the SDM took 0.6133 s to compute and had an RMSE of 4.72×10^{-5}. The DDM, on the other hand, minimized the RMSE to 1.52113×10^{-4} in only 0.9482 s.

In the next experiment, we kept the sample size the same while increasing the number of iterations by a factor of 200. By making this modification, we were able to find the algorithm's minimum RMSE. With 900 iterations and a population size of 50, the optimum RMSE solution for the SDM was found, and the resultant calculation time was around 2.6704 s. For the DDM, the optimal solution took 50 population sizes and 1500 iterations, with an execution time of around 7.082 s. Table 8 displays the results of the experiment with the SDM, whereas Table 9 displays the experiment results with the DDM.

Table 8. aSi iteration vs. population size SDM case.

Number of Runs	Population Size	Number of Iterations	RMSE	Run Time (s)
5	20	100	2.6833×10^{-4}	0.1297
	30	100	2.6592×10^{-4}	0.1885
	50	100	2.3619×10^{-4}	0.3064
	60	100	4.7753×10^{-5}	0.3708
	80	100	9.9107×10^{-5}	0.4944
	100	100	4.7163×10^{-5}	0.6134
	50	300	4.8050×10^{-5}	0.9202
	50	500	4.7721×10^{-5}	1.4856
	50	700	4.6683×10^{-5}	2.0660
	50	900	4.6123×10^{-5}	2.6705
	50	1100	4.6123×10^{-5}	3.2334
	50	1300	4.6123×10^{-5}	3.8267
	50	1500	4.6123×10^{-5}	4.3954

Table 9. aSi iteration vs. population size DDM case.

Number of Runs	Population Size	Number of Iterations	RMSE	Run Time (s)
5	20	100	2.1202×10^{-4}	0.1945
	30	100	3.4918×10^{-4}	0.2907
	50	100	2.4165×10^{-4}	0.4771
	60	100	9.3571×10^{-5}	0.5735
	80	100	1.6082×10^{-4}	0.7634
	100	100	1.5211×10^{-4}	0.9482
	50	300	6.7630×10^{-5}	1.4260
	50	500	4.6719×10^{-5}	2.3688
	50	700	4.2545×10^{-5}	3.3233
	50	900	4.5706×10^{-5}	4.2179
	50	1100	4.5534×10^{-5}	5.4857
	50	1300	4.1202×10^{-5}	6.4356
	50	1500	4.0943×10^{-5}	7.0822

In this experiment, the CPU execution times for the SDM and DDM were evaluated across two different processors: Intel Core i9 and AMD Ryzen 5 5600H. Table 10 presents the results, highlighting that the Intel Core i9 consistently outperformed the AMD Ryzen 5 5600H for both models in terms of execution speed. The SDM required less execution time than the DDM on both processors, indicating a simpler computational complexity. Specifically, the Intel Core i9 achieved 3.2214 s for the SDM and 6.2822 s for the DDM, while the AMD Ryzen 5 took 5.9455 s and 9.9611 s, respectively. These differences can be attributed to the Intel Core i9's superior architecture, clock speeds, and optimizations for single-threaded performance, which the algorithm likely benefits from. Conversely, the AMD Ryzen 5, while powerful, may not be as efficient for workloads with such specific computational demands.

Table 10. aSi CPU execution time.

Number of Runs	Model	Population Size/ Number of Iterations	RMSE	Intel Core i9 CPU Execution Time (s)	AMD Ryzen 5 5600H CPU Execution Time (s)
5	SDM	100/1100	4.6123×10^{-5}	3.2214	5.9455
	DDM	100/1100	4.0943×10^{-5}	6.2822	9.9611

3.3. mSi Cell

The monocrystalline silicon photovoltaic is the subject of this third solar cell analysis. At 1000 W/m^2 irradiance and 27 °C temperature, the I–V characteristic of mSi silicon is evaluated. Current values derived using the SDMs and DDMs, together with their corresponding errors, are reported in Table S3 of the final dataset [29].

The mSi photovoltaic cell's PSA-extracted values for the SDMs and DDMs are shown in Table 11. The SDM took 3.9440 s to execute, whereas the DDM version took 5.5715 s. Algorithms' effectiveness is measured using RMSE, MBE, and STD to evaluate solutions obtained by ISOA, HSDA, and the Barnacles Mating Optimizer Algorithm (BMOA) [48].

Table 11. mSi photovoltaic cell RMSE and extracted parameters for various algorithms.

Algorithm	Model	I_{ph} (A)	I_{sd1}(μA)	n_1	R_s(Ω)	R_{sh}(Ω)	I_{sd2}(μA)	n_2	RMSE	MBE	STD
PSA	**SDM**	**0.425752032**	**0.517217783**	**1.67956503**	**0.0913092173**	**99.1799271**	**-**	**-**	$\mathbf{5.63096 \times 10^{-4}}$	$\mathbf{1.025410758621 \times 10^{-7}}$	$\mathbf{5.533625 \times 10^{-2}}$
ISOA	SDM	0.425745	0.523619	1.68107	0.0911642	99.8127	-	-	5.5854×10^{-4}	$7.025715034483 \times 10^{-8}$	5.533602×10^{-2}
HSDA	SDM	0.425753	0.516241	1.679334	0.0913289	99.07598	-	-	5.63098×10^{-4}	3.50364×10^{-7}	5.533616×10^{-2}
BMOA	SDM	0.42660	0.54453	1.4429	0.11186	50.305	-	-	1.8456×10^{-3}	5.1804×10^{-1}	$1.290548 \times 10^{+0}$
PSA	**DDM**	**0.425311171**	**2.59254768**	**2.00693446**	**0.113483601**	**200.0**	$\mathbf{2.29084278 \times 10^{-5}}$	**1.000**	$\mathbf{2.6216 \times 10^{-4}}$	$\mathbf{2.028312551724 \times 10^{-8}}$	$\mathbf{5.527109 \times 10^{-2}}$
ISOA	DDM	0.425579	0.131003	1.53882	0.0960953	130.184	7.18119	2.57289	4.4383×10^{-4}	$1.025653137931 \times 10^{-7}$	5.535608×10^{-2}
HSDA	DDM	0.425748	0.2402586	1.679334	0.0913289	105.075	1.0301352	2.091868	5.3057×10^{-4}	-5.75799×10^{-7}	7.357780×10^{-2}
BMOA	DDM	0.42731	0.056095	1.4532	0.10822	53.106	0.53674	2.0685	1.3754×10^{-3}	2.4771×10^{-4}	5.535128×10^{-2}

The results of PSA algorithm are highlighted in bold.

In terms of the SDM, the RMSE value achieved by the PSA is not exceeded by ISOA; however, it does vary slightly from HSDA. In the instance of the DDM, the RMSE value improves by 40.9% when compared to ISOA and 50.58% when compared to HSDA. Despite these gains, the RMSE for both models generated by the BMOW algorithm is still greater than that achieved by the PSA.

The SDM's RMSE findings show that the PSA was below ISOA's RMSE value, with the discrepancies in the extracted parameters being very small. But there are notable differences when compared to other algorithms like BMOA and HSDA. In contrast, the PSA surpasses all other algorithms in the DDM with a high improvement, as seen in Table 11.

The performance of the BMOA was poor, since it became stuck in a local minimum and could not achieve the optimal results. When comparing the PSA findings to those of other algorithms, significant discrepancies were found after assessing all of the obtained parameters for both models. The PSA's parameters, specifically in the SDM, exhibited only minor variations in values compared to ISOA, resulting in a greater RMSE value than that attained by ISOA, where ISOA obtained 5.5854×10^{-4} while PSA obtained 5.63096×10^{-4}. The PSA, however, produced significant improvements in the DDM, notably in reverse saturation for the recombination and generation processes, as well as changes in series and shunt resistance. In terms of RMSE, PSA obtained 2.6216×10^{-4} while ISOA obtained 4.4383×10^{-4}, representing an improvement of around 40.9% by PSA in this case. When compared to ISOA and HSDA, these improvements produced a greater overall improvement. Importantly, the PSA's capability to locate the ideal solution and avoid being trapped in local minima is shown by the assessment of all mSi photovoltaic cell characteristics and the RMSE.

The alignment of the estimated and observed I–V and P–V characteristics generated by the PSA is shown in Figures 12 and 13, allowing a comparison of the measured and computed currents for both the SDMs and DDMs.

Figure 14 shows the absolute errors, which are the differences between the observed and estimated currents for both models as determined by the algorithm. In the case of the DDM, the absolute errors are very low in comparison with the ones calculated with the SDM.

The convergence curves of the PSA method for both models are shown in Figure 15, with a maximum iteration of 1500 and a population size of 100. The graphic depicts the iteration at which the algorithm starts producing the lowest RMSE result.

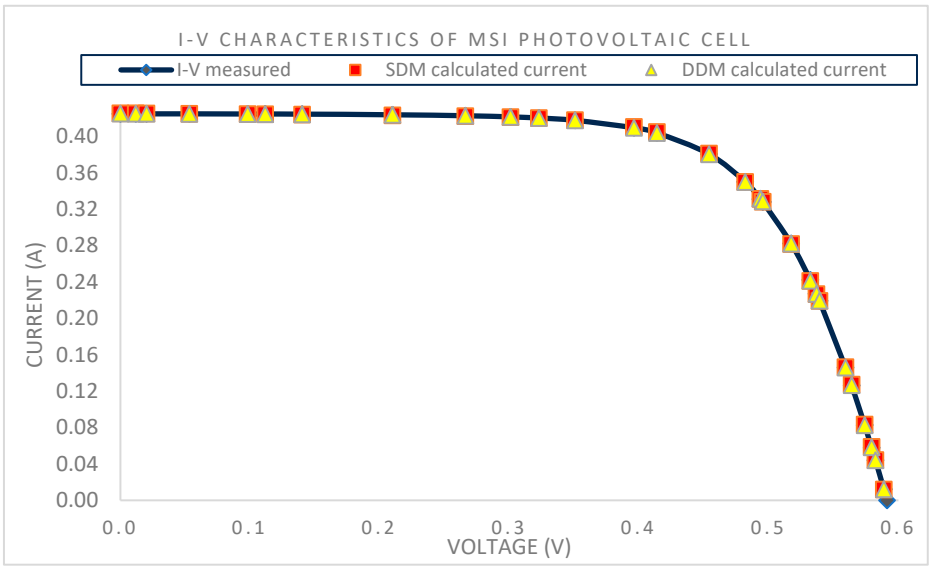

Figure 12. mSi SDM and DDM I–V characteristics.

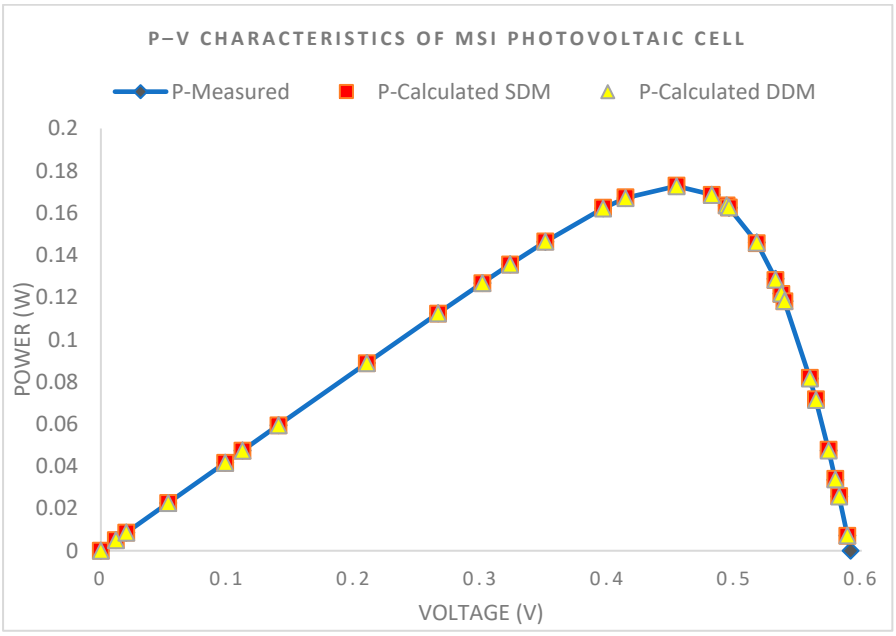

Figure 13. mSi SDM and DDM P–V characteristics.

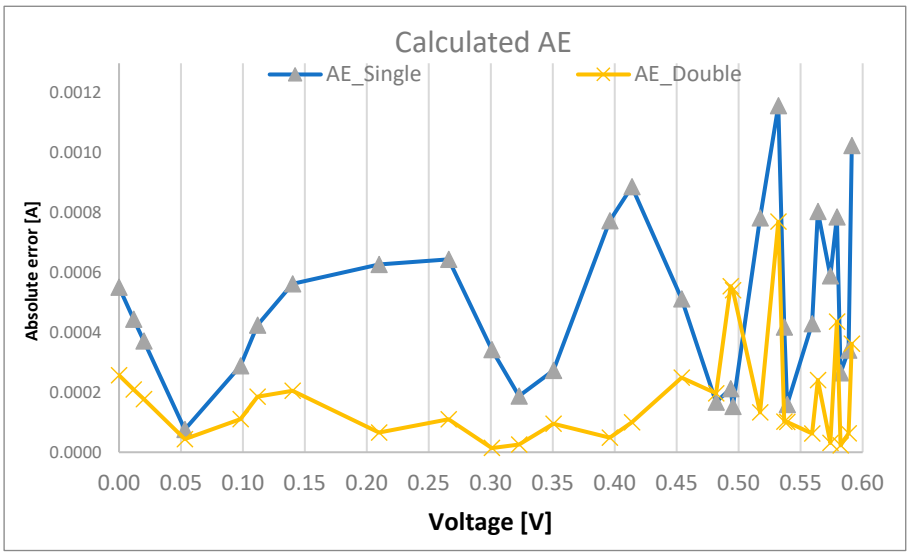

Figure 14. mSi SDM and DDM calculated absolute error.

Starting with iteration 220 at 66 population for the SDM, the PSA showed evidence of escaping the local minimum and nearing the ideal RMSE value. For the DDM, the algorithm began this process at iteration 100, although it took a significant jump to obtain the optimum RMSE at iteration 780 and 70 population.

To provide additional insight into the algorithm's reliability, shaded regions corresponding to ± standard deviation across iterations are included. These bands demonstrate the level of variability in RMSE values at each iteration, offering a visual measure of algorithm stability. Narrower bands indicate consistent behavior across optimization steps, affirming PSA's robustness in both SDMs and DDMs.

An initial investigation was performed on both models with a fixed iteration number set at 100 and the size of the population ranging between 20 and 100 to determine the most effective combination of iteration size and population size for reaching the best RMSE. The lowest RMSE value was attained when the population size and the number of iterations were both adjusted to 100; therefore, this method clearly did not work. The study showed

that the RMSE for the SDM was 1.61×10^{-3} and the calculation time was 0.4073 s, whereas the RMSE for the DDM was 1.3605×10^{-3} and the computation time was 0.6340 s.

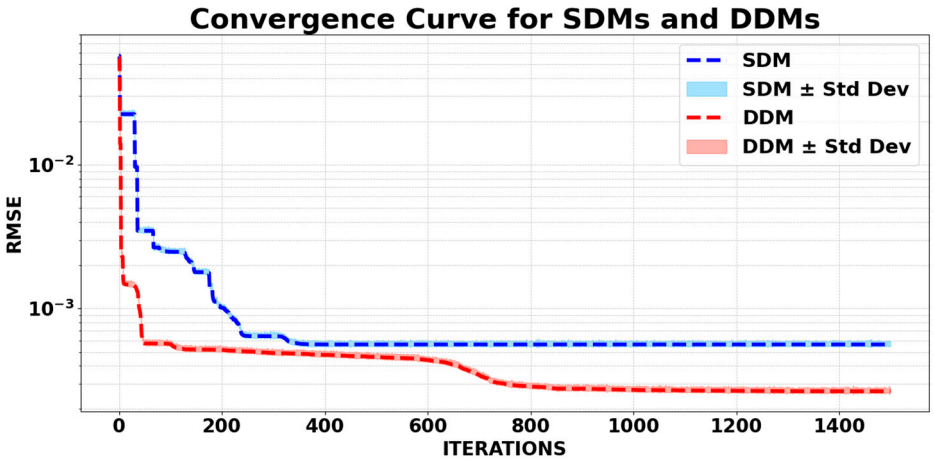

Figure 15. mSi SDM and DDM convergence curve.

In the next experiment, the population size was held constant, but the number of iterations was increased by 200. The lowest RMSE value achievable by the method was thus determined. It took around 2.2343 s to obtain the optimum RMSE solution for the SDM with a population size of 50 and 1100 iterations. The DDM took around 7.082 s to calculate after 1500 iterations with a population size of 50. Since DDM requires the extraction of seven parameters, the search for the best RMSE takes more time, resulting in a higher number of iterations.

Table 12 presents the results of the SDM study, while Table 13 presents the findings of the DDM analysis.

Table 12. mSi iteration vs. population size SDM case.

Number of Runs	Population Size	Number of Iterations	RMSE	Run Time(s)
5	20	100	3.3469×10^{-3}	0.0888
	30	100	2.4630×10^{-3}	0.1277
	50	100	8.4297×10^{-4}	0.2075
	60	100	3.2550×10^{-3}	0.2474
	80	100	1.9829×10^{-3}	0.3272
	100	100	1.6111×10^{-3}	0.4074
	50	300	9.8442×10^{-4}	0.6108
	50	500	7.4310×10^{-4}	1.0200
	50	700	6.7488×10^{-4}	1.4103
	50	900	5.9296×10^{-4}	1.8384
	50	1100	5.6309×10^{-4}	2.2344
	50	1300	5.6309×10^{-4}	2.6551
	50	1500	5.6309×10^{-4}	3.1299

Table 13. mSi iteration vs. population size DDM case.

Number of Runs	Population Size	Number of Iterations	RMSE	Run Time(s)
5	20	100	1.5243×10^{-3}	0.1945
	30	100	3.4145×10^{-3}	0.2907
	50	100	4.2702×10^{-3}	0.4771
	60	100	3.6405×10^{-3}	0.5735
	80	100	3.4981×10^{-3}	0.7634
	100	100	1.3605×10^{-3}	0.9482
	50	300	2.8906×10^{-3}	1.4260
	50	500	9.9828×10^{-4}	2.3688
	50	700	1.3823×10^{-3}	3.3233
	50	900	5.0116×10^{-4}	4.2179
	50	1100	5.2988×10^{-4}	5.4857
	50	1300	4.7973×10^{-4}	6.4356
	50	1500	2.6216×10^{-4}	7.0822

In this analysis, the CPU execution times for the SDM and DDM were compared on Intel Core i9 and AMD Ryzen 5 5600H processors, as summarized in Table 14. The results indicate that the Intel Core i9 performed consistently faster than the AMD Ryzen 5 for both models, with execution times of 3.7890 s for the SDM and 5.2777 s for the DDM on the Intel processor, compared to 6.1665 s and 8.8891 s, respectively, on the AMD processor. The Intel Core i9's superior execution times can be attributed to its advanced architecture, higher single-threaded performance, and optimizations for computational workloads, while the AMD Ryzen 5, though competitive, lags behind in handling the algorithm's demands.

Table 14. mSi CPU execution time.

Number of Runs	Model	Population Size/ Number of Iterations	RMSE	Intel Core i9 CPU Execution Time (s)	AMD Ryzen 5 5600H CPU Execution Time (s)
5	SDM	100/1100	5.6309×10^{-4}	3.7890	6.1665
	DDM	100/1100	2.6216×10^{-4}	5.2777	8.8891

3.4. PVM Panel

PVM, short for PVM 752 GaAs thin-film photovoltaic panel, is the second photovoltaic panel being examined. The PVM I–V characteristic is being measured at 1000 W/m^2 irradiance and 25 °C. The result is observed [29], containing the measured data and current values derived using the SDMs and DDMs, together with the corresponding errors, and can be found in Table S4.

The characteristics of the PVM photovoltaic panel that were derived using the PSA for the SDMs and DDMs are shown in Table 15. The SDM takes 5.3845 s to execute, whereas the DDM takes 8.5250 s. The performance of the algorithm, comprising BWOA, HSDA, AOS, OBEDO, and Supply-Demand-Based Optimization (SDO) [49], is measured by using RMSE.

Table 15. RMSE of the PVM photovoltaic panel and the extracted parameters for different algorithms.

Algorithm	Model	I_{ph} (A)	I_{sd1}(μA)	n_1	R_s(Ω)	R_{sh}(Ω)	I_{sd2}(μA)	n_2	RMSE	MBE	STD
PSA	**SDM**	**0.100066826**	$\mathbf{3.778895 \times 10^{-6}}$	**1.61567282**	**0.660508880**	**608.00995**	**-**	**-**	$\mathbf{2.2780 \times 10^{-4}}$	$\mathbf{-3.7727 \times 10^{-12}}$	$\mathbf{2.304376 \times 10^{-4}}$
BWOA	SDM	0.10006682	3.7788×10^{-12}	1.6156728	0.660508875	608.00993	-	-	2.2780×10^{-4}	3.82764×10^{-12}	2.304380×10^{-4}
HSDA	SDM	0.1000275091	5.94432×10^{-6}	1.6466902	0.65014344	678.14612	-	-	2.3469×10^{-4}	4.21357×10^{-9}	2.374105×10^{-4}
SDO	SDM	0.1000	5.9440×10^{-6}	1.6467	0.6499	668.5946	-	-	2.3487×10^{-4}	3.283863×10^{-5}	2.374245×10^{-4}
AOS	SDM	0.10001	5.944×10^{-6}	1.646700	0.6479	662.9896	-	-	2.3921×10^{-4}	5.575000×10^{-6}	2.419182×10^{-4}
OBEDO	SDM	0.10016	1.85×10^{-6}	1.5691	0.6777	519.07	-	-	2.4818×10^{-4}	-9.57454×10^{-6}	2.503665×10^{-4}
PSA	**DDM**	**0.100000321**	**3.96450670**	**5.78835276**	**0.676629575**	**1000**	$\mathbf{1.0 \times 10^{-6}}$	**1.53243**	$\mathbf{1.6579 \times 10^{-4}}$	$\mathbf{2.27272 \times 10^{-10}}$	$\mathbf{1.677123 \times 10^{-4}}$
BWOA	DDM	0.1000106735	9.988550426	7.17885667	0.6724961417	983.11164	1.5226×10^{-12}	1.55771	1.67797×10^{-4}	-1.69207×10^{-6}	1.798672×10^{-4}
OBEDO	DDM	0.10009	1.89×10^{-8}	1.3561	0.6786	618.21	2.43×10^{-5}	1.8068	2.0777×10^{-4}	1.687500×10^{-6}	2.138220×10^{-4}
HSDA	DDM	0.1000559745	7.26940×10^{-5}	2.00260802	0.669003416	634.06199	6.222×10^{-7}	1.51479	2.1306×10^{-4}	-1.77149×10^{-7}	2.161703×10^{-4}
SDO	DDM	0.1001	7.2313×10^{-4}	2.0000	0.6684	637.3026	6.23×10^{-5}	1.5152	2.1318×10^{-4}	3.283863×10^{-5}	2.374245×10^{-4}
AOS	DDM	0.103192	1.775×10^{-4}	1.999	0.6547	200	1×10^{-6}	1.57105	1.7804×10^{-3}	-4.38184×10^{-4}	1.984256×10^{-3}

The results of PSA algorithm are highlighted in bold.

Both the PSA and the BWOA provide the same RMSE values for the SDM, indicating that there is little difference between the two approaches. However, when HSDA and SDO are put up against PSA, the proposed algorithm performs much better, yielding a 3% advantage over both algorithms.

On the other hand, the RMSE improvement using the PSA for the DDM is impressive: 1.19% better than BWOA and 22.18% better than HSDA. Despite this, for both models, the RMSE produced using the PSA method is still better than other algorithms.

According to the RMSE values for the SDM, the PSA method achieves the same results as BWOA (2.2780×10^{-4}), with hardly any changes in the parameters that are found. However, as demonstrated in Table 15, when comparing it with other algorithms such as BWOA and HSDA, significant discrepancies are found. The PSA performs better than other algorithms in all statistical tests, as shown in particular by the DDM.

AOS algorithms were unable to locate the lowest RMSE value because they were caught in local minima, which prevented them from obtaining the ideal RMSE. On the other hand, compared to other methods, substantial differences may be found when examining all obtained parameters using the PSA approach for both models.

In the instance of SDM, the PSA produced significantly larger parameter value differences for the I_{sd1} of 3.778895×10^{-6} compared to BWOA with 3.7788×10^{-12}, which resulted in a significantly lower RMSE value. Compared to BWOA and HSDA, the PSA improved the DDM the most by changing the reverse saturation for the recombination and generation processes and showing the most change in series and shunt resistance. The evaluation of all PVM solar module parameters and the RMSE show that the PSA can choose the best solution and not become stuck in local minima.

Figures 16 and 17 show the PSA's estimated and observed I–V and P–V for the SDMs and DDMs to show the difference between the measured current and the calculated current.

The absolute errors acquired by the algorithm are shown in Figure 18, which illustrates the differences between the observed current and the predicted current for both the SDMs and DDMs. In the case of the DDM, the absolute errors are very low in comparison with the ones calculated with the SDM.

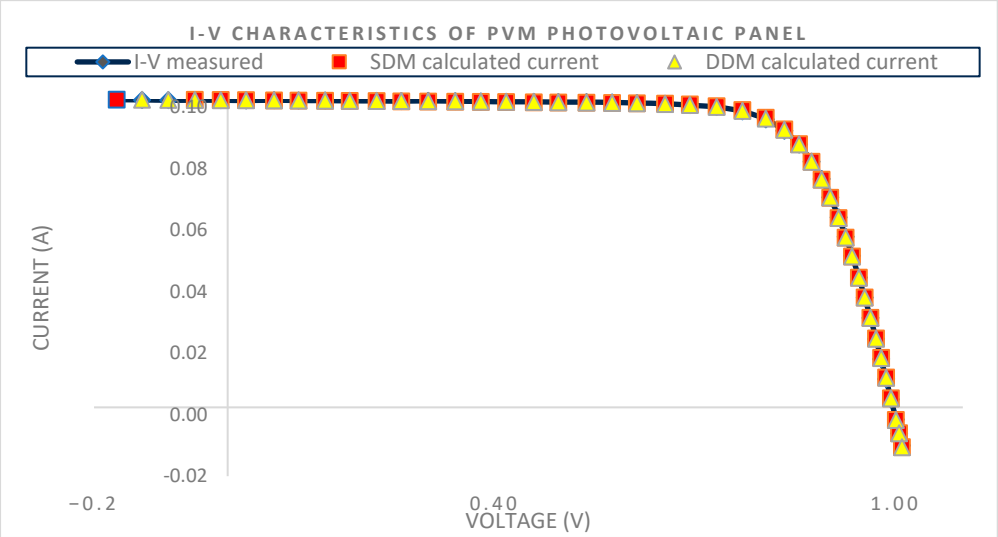

Figure 16. PVM SDM and DDM I–V characteristics.

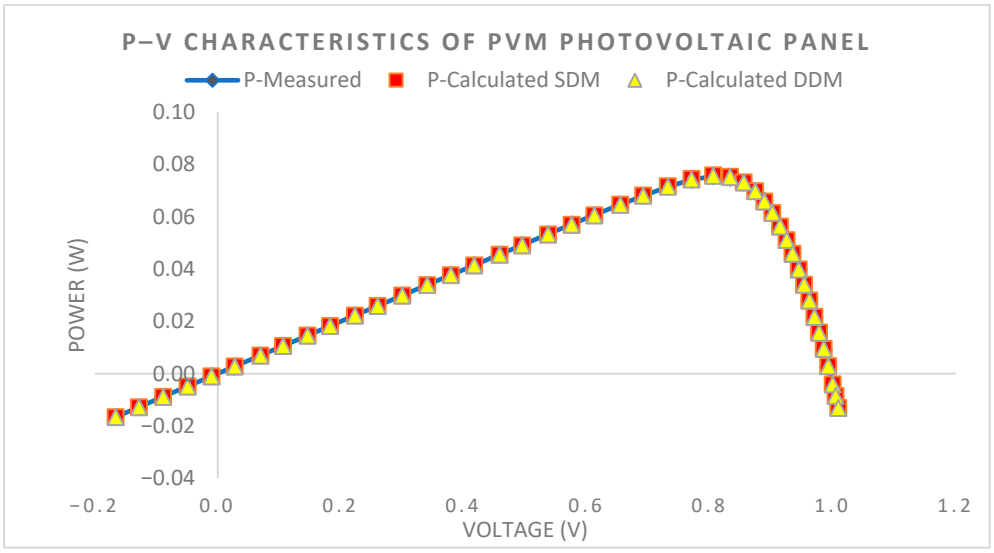

Figure 17. PVM SDM and DDM P–V characteristics.

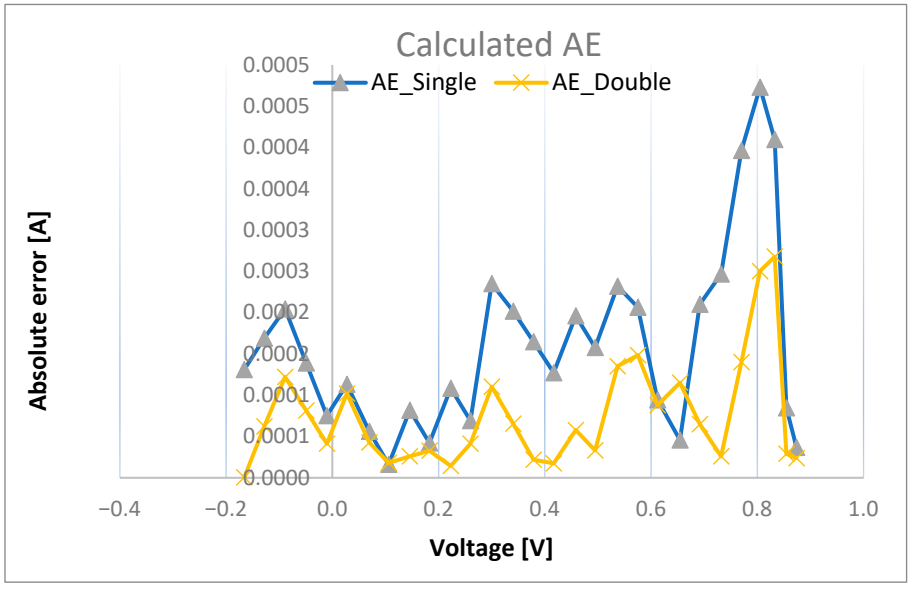

Figure 18. PVM SDM and DDM calculated absolute error.

The PSA's convergence curves for both models are shown in Figure 19. The algorithm's maximum iteration was set at 1500, with a population of 100. The graph shows which iterations the algorithm went through to reach the lowest RMSE value. The PSA began to evade the local minimum for the SDM around iteration 270 at 80 populations, arriving at the ideal RMSE value. For the DDM, on the other hand, this improvement started at iteration 370 at population 65, showing that it required a few more iterations to obtain the ideal RMSE value compared to the SDM.

To provide a clearer view of performance consistency, the convergence curves are accompanied by shaded bands representing ± standard deviation at each iteration. These bands indicate the degree of fluctuation in RMSE values and offer visual confirmation of the PSA's stability and reliability during the optimization process.

Two tests were undertaken for both the SDMs and DDMs to find the appropriate combination of iteration size and population size for achieving the ideal RMSE in the case of PVM.

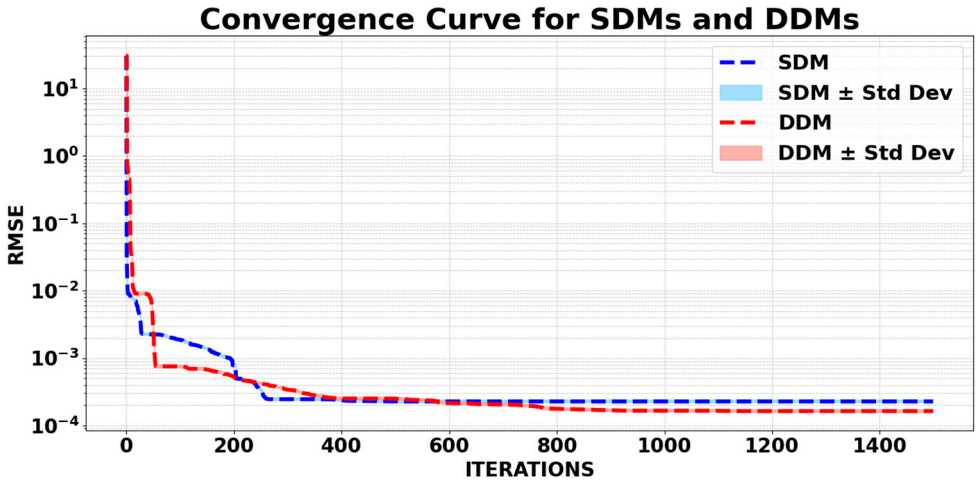

Figure 19. PVM SDM and DDM convergence curve.

The population size in the first experiment varied from 20 to 100, but the number of iterations remained constant at 100. This method, however, did not produce the best results, as the lowest RMSE value was found when both the population size and iteration were set to 100. This resulted in an RMSE of 2.2031×10^{-3} and a computation time of 0.6249 s for the SDM. Meanwhile, the DDM achieved a minimal RMSE of 5.9449×10^{-4} in 0.9250 s of computing time.

In the second experiment, the number of iterations varied by 200 each time while the population size remained the same. The best RMSE value for the algorithm was calculated using this strategy. The SDM's optimum RMSE solution, computed with 1500 iterations and a population size of 50, took about 4.6392 s to complete. With a population size of 50, the optimal solution in the DDM was similarly discovered after 1500 iterations; however, this time the calculation took roughly 6.8158 s. Finding the ideal RMSE requires more iterations for the DDM than it does for the SDM since the DDM requires the extraction of seven parameters. The findings of the SDM analysis are shown in Table 16, while the DDM analysis results are shown in Table 17.

Table 16. PVM iteration vs. population size SDM case.

Number of Runs	Population Size	Number of Iterations	RMSE	Run Time (s)
5	20	100	3.6985×10^{-3}	0.1184
	30	100	1.5959×10^{-2}	0.1756
	50	100	7.4701×10^{-3}	0.2828
	60	100	5.8390×10^{-3}	0.3432
	80	100	2.3133×10^{-3}	0.5087
	100	100	2.2031×10^{-3}	0.6249
	50	300	2.2025×10^{-3}	0.9447
	50	500	9.6592×10^{-4}	1.5563
	50	700	2.7921×10^{-4}	2.1904
	50	900	2.2026×10^{-3}	2.7723
	50	1100	6.3920×10^{-4}	3.3992
	50	1300	2.5767×10^{-4}	3.9922
	50	1500	2.2780×10^{-4}	4.6392

Table 17. PVM iteration vs. population size DDM case.

Number of Runs	Population Size	Number of Iterations	RMSE	Run Time (s)
5	20	100	5.5364×10^{-3}	0.1895
	30	100	7.6733×10^{-3}	0.2784
	50	100	8.9201×10^{-4}	0.4655
	60	100	3.3552×10^{-3}	0.5530
	80	100	2.9206×10^{-3}	0.7356
	100	100	5.9449×10^{-4}	0.9250
	50	300	5.1178×10^{-3}	1.3636
	50	500	4.8415×10^{-4}	2.2742
	50	700	2.3892×10^{-4}	3.1928
	50	900	3.2303×10^{-4}	4.1243
	50	1100	6.6249×10^{-4}	5.0614
	50	1300	1.6835×10^{-4}	5.8600
	50	1500	1.6579×10^{-4}	6.8158

In this experiment, summarized in Table 18, the performance of the SDMs and DDMs was evaluated in terms of CPU execution times across two processors: Intel Core i9 and AMD Ryzen 5 5600H. On the Intel Core i9, the DDM required 8.5112 s, significantly more than the 5.2231 s for the SDM. Similarly, on the AMD Ryzen 5, the DDM took 12.1566 s compared to 9.8966 s for the SDM. The differences in execution times between the processors also highlight the Intel Core i9's better efficiency in managing computational workloads, which is likely due to its optimized architecture and superior single-thread performance.

Table 18. PVM CPU execution time.

Number of Runs	Model	Population Size/ Number of Iterations	RMSE	Intel Core i9 CPU Execution Time (s)	AMD Ryzen 5 5600H CPU Execution Time (s)
5	SDM	100/1100	2.2780×10^{-4}	5.2231	9.8966
	DDM	100/1100	1.6579×10^{-4}	8.5112	12.1566

3.5. STM6-40 Panel

The third solar panel under consideration is an STM6-40 monocrystalline panel manufactured by Schutten Solar. These panels are made up of 36 cells connected in series [50]. The I–V characteristic of the STM6-40 is measured at 1000 W/m^2 irradiance and 51 °C. The dataset of 20 points [35], which is shown in Table S5, used to extract the parameters, includes the current values derived using the SDMs and DDMs, as well as their corresponding errors.

The STM6-40 photovoltaic panel's PSA-obtained parameters are displayed in Table 19 for both the SDM and DDM types. The SDM took 2.56053 s to execute, while the DDM version took 4.4231 s. RMSE, MBE, and STD were used to evaluate the effectiveness of the algorithm.

Table 19. RMSE of the STM6-40 photovoltaic panel and the extracted parameters for different algorithms.

Algorithm	Model	I_{ph} (A)	I_{sd1}(μA)	n_1	R_s(Ω)	R_{sh}(Ω)	I_{sd2}(μA)	n_2	RMSE	MBE	STD
PSA	**SDM**	**1.66390478**	**1.73865701**	**54.7309054**	**0.153855757**	**573.418599**	**-**	**-**	$\mathbf{1.72981 \times 10^{-3}}$	$\mathbf{-2.5000000 \times 10^{-10}}$	$\mathbf{1.774752 \times 10^{-3}}$
ELADE	SDM	1.66390478	1.73865704	54.7309054	0.153855756	573.4186142	-	-	1.72981×10^{-3}	-1.000×10^{-9}	1.774752×10^{-3}
HSDA	SDM	1.663904779	1.7386543978	54.730899	0.1538559327	543.4183498	-	-	1.72981×10^{-3}	1.58×10^{-10}	1.875694×10^{-3}
RLDE	SDM	1.66390478	1.73865695	54.7309054	0.1538557624	573.4185940	-	-	1.72981×10^{-3}	$-7.500000010295 \times 10^{-9}$	1.774751×10^{-3}
GSK	SDM	1.6635	1.9240	55.134	0.144	595.7656	-	-	1.72981×10^{-3}	$1.841645000000 \times 10^{-4}$	1.798958×10^{-3}
DE/BSA	SDM	1.66390477	1.73865688	54.7309051	0.00427377	15.92829377	-	-	1.72981×10^{-3}	$7.514688560000 \times 10^{-1}$	3.788024×10^{-1}
IMPA	SDM	1.66390478	1.738656929	54.7309052	0.153855762	573.418589	-	-	1.72981×10^{-3}	$-1.500000028527 \times 10^{-9}$	1.774750×10^{-3}
ELPSO	SDM	1.666268	4.59614×10^{-1}	50.458643	0.5	497.747315	-	-	2.1803×10^{-3}	$1.457340026790 \times 10^{+4}$	$3.158300 \times 10^{+4}$
PSA	**DDM**	**1.66406725**	$\mathbf{2.42076 \times 10^{-5}}$	**31.6892203**	**0.331459896**	**613.868341**	**2.64345628**	**58.0305558**	$\mathbf{1.6892 \times 10^{-3}}$	$\mathbf{-2.965000000266 \times 10^{-7}}$	$\mathbf{1.733171 \times 10^{-3}}$
HSDA	DDM	1.663877330	1.1006110140	53.3408920	0.1635655746	580.9770182	1.681213674	65.87326495	1.72072×10^{-3}	4.98823×10^{-7}	1.765430×10^{-3}
ELPSO	DDM	1.664843	1.67016×10^{-2}	41.993481	0.50	606.888301	6.210924	67.344124	1.8307×10^{-3}	-8.2272×10^{-5}	3.966815×10^{-2}
CPSO	DDM	1.664746	2.52914×10^{-2}	42.816947	0.50	611.747136	8.779822	70.734863	1.8343×10^{-3}	1.15492×10^{-4}	4.034324×10^{-2}
BSA	DDM	1.661112	1.198529	100	0.50	924.813027	8.9167×10^{-1}	52.874278	4.0335×10^{-3}	-8.48803×10^{-4}	6.383651×10^{-2}
ABC	DDM	1.663472	8.937775	71.464981	1.236435	938.209991	1×10^{-6}	27.790714	2.0538×10^{-3}	-2.42145×10^{-4}	1.639806×10^{-1}

The results of PSA algorithm are highlighted in bold.

The PSA is compared with ELADE, HSDA, GSK, IMPA, DE/BSA, reinforcement learning-based differential evolution (RLDE) [51], rnhanced leader particle swarm optimization (ELPSO) [52], conventional PSO (CPSO) [52], backtracking search algorithm (BSA) [52], and artificial bee colony (ABC) [52]. The PSA and other algorithms produce comparable RMSE values for the SDM, showing no variation in performance. However, when comparing PSA with ELPSO, it is roughly 20.6% better. The DDM's RMSE value is 1.83% better than HSDA's result and 7.72% better than it was when ELPSO was used. For both models, the RMSE performance of the PSA beats the other algorithms.

RMSE for SDM shows that the PSA yielded the same results (1.72981×10^{-3}) as the ELADE, HSDA, RLDE, IMPA, and GSK algorithms, showing that the differences between them were quite small. In the case of the DDM, when compared to other algorithms like HSDA and ELPSO, however, certain key differences become apparent. Table 19 shows that, in contrast, the PSA consistently achieves the highest statistical success rates in the DDM. When compared to other methods, the PSA's study of all obtained parameters reveals significant discrepancies.

The PSA's parameters displayed slightly bigger variations in values as compared to HSDA in the SDM, leading to a slightly better RMSE value than that produced by other algorithms. On the other hand, the PSA was better in the case of the DDM with an RMSE of 1.6892×10^{-3} compared to either HSDA with 1.72072×10^{-3} or ELPSO with 1.8307×10^{-3}. This is because the PSA could make more subtle changes to reverse saturation for the recombination and generation mechanisms, as well as changes to series and shunt resistance. Calculating the RMSE for each parameter of an STM6-40 photovoltaic module shows that the PSA method can find the best solution without becoming stuck in local minima.

Figures 20 and 21 depict the consistency between the estimated and observed I–V and P–V characteristics generated by the PSA, allowing one to visually identify the discrepancies between the measured and calculated current in both the SDMs and DDMs.

Figure 22 displays the absolute errors, which are estimated by the PSA, between the observed and predicted currents for both models.

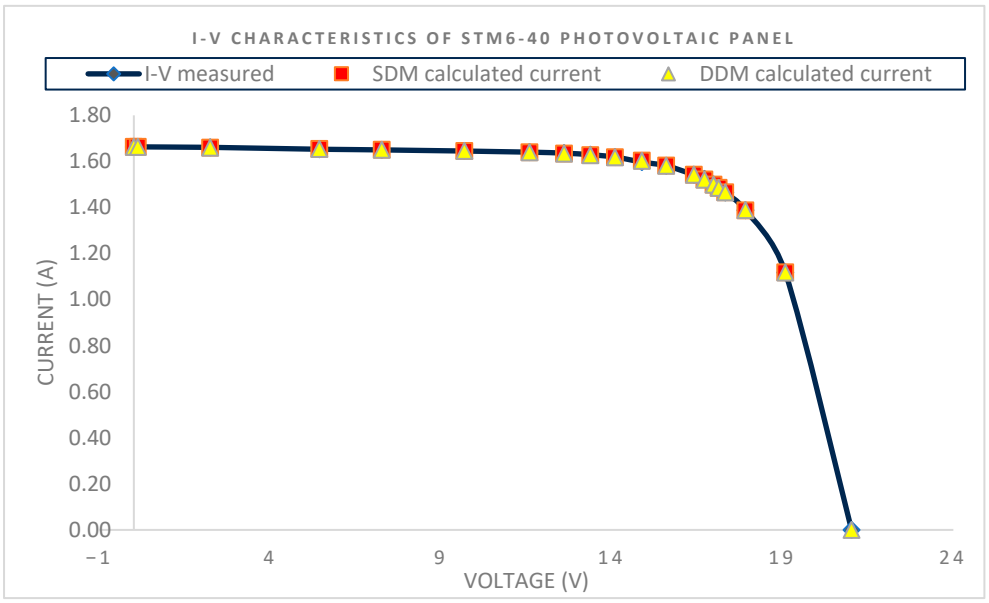

Figure 20. STM6-40 SDM and DDM I–V characteristics.

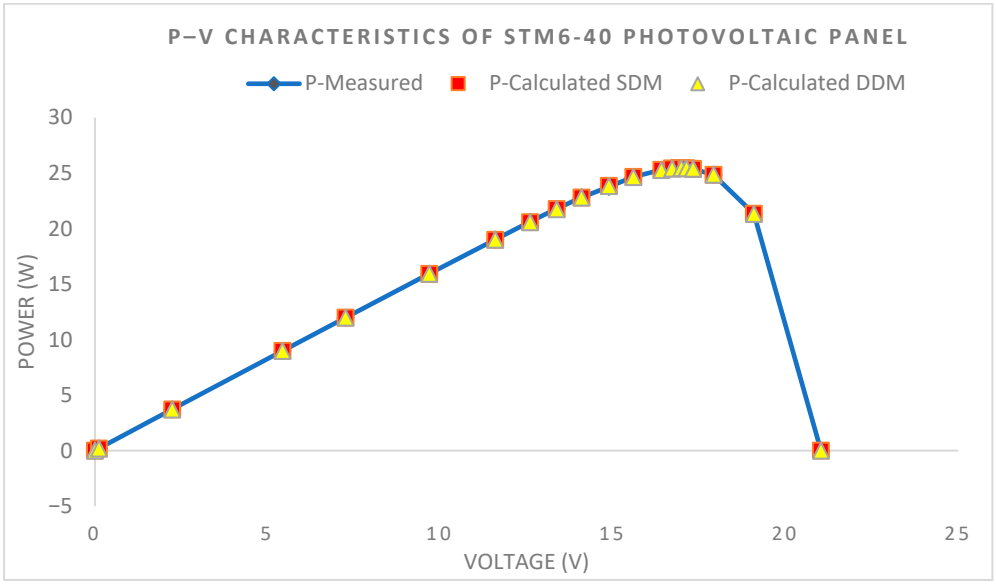

Figure 21. STM6-40 SDM and DDM P–V characteristics.

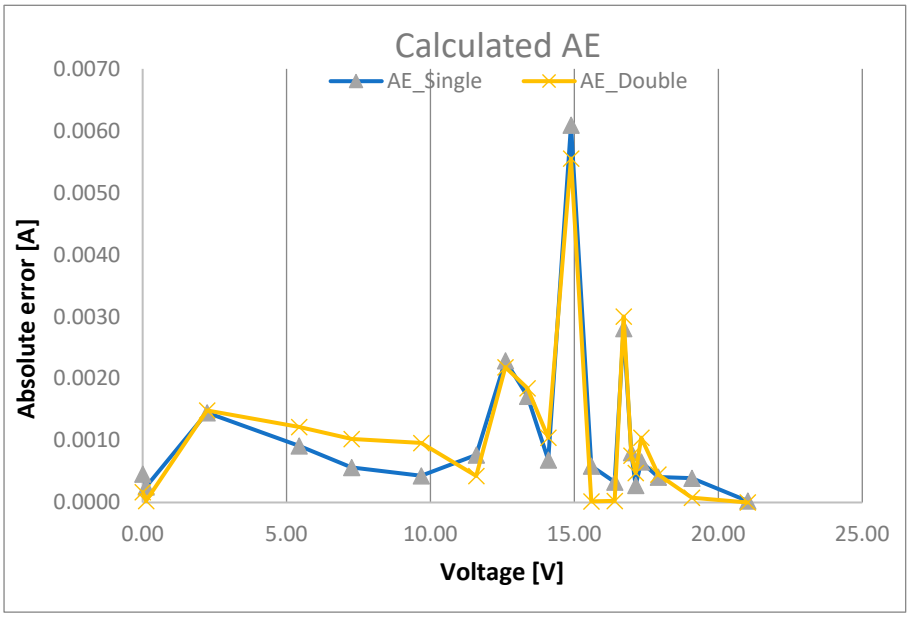

Figure 22. STM6-40 SDM and DDM calculated absolute error.

The PSA's convergence curves are shown in Figure 23 for both models, with a maximum of 1500 iterations and a population size of 100. The figure shows the iteration at which the algorithm started attaining the minimum RMSE. At iteration 50 and a population of 89, the PSA began to escape the local minimum and eventually reached the optimal RMSE for the SDM. However, in the DDM, the PSA started making headway toward the best possible solution around iteration 120 and a population of 33.

To assess the consistency of the algorithm's performance, shaded regions representing ± standard deviation across the iterations have been added to the convergence curves. These bands help visualize the variability in RMSE values throughout the optimization and provide further evidence of the algorithm's robustness and stability across runs.

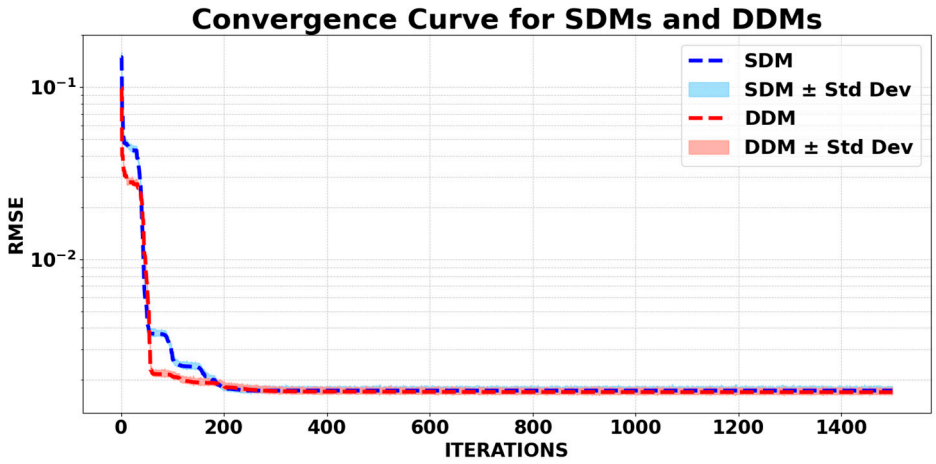

Figure 23. STM6-40 SDM and DDM convergence curve.

Two experiments were performed to establish the ideal iteration size and population size for minimizing the RMSE in the context of STM6-40. While the number of iterations was held constant at 100 for both models, the population size ranged from 20 to 100 in the first investigation. The shortest RMSE values were achieved when the population size and iteration were both set to 100; therefore, this strategy did not end up being the most effective. The resulting RMSE for the SDM was 2.08×10^{-3}, and the computation took 0.3170 s. The best RMSE value for the DDM was 1.80×10^{-3}, and it took 0.486 s to compute.

The second experiment involved keeping the population size the same but increasing the number of iterations by a factor of 200. This was a more efficient means of determining the RMSE of the algorithm. It took about 1.0971 s to calculate the optimal RMSE solution for the SDM with a population size of 50 and 700 iterations. However, the DDM required a larger number of iterations (1500) and a population size of 50, resulting in a computation time of about 3.5337 s due to the need to extract seven parameters. The intricacy of parameter extraction necessitated a longer iteration time in DDM to achieve the best RMSE. Table 20 displays the SDM analysis results, and Table 21 displays the DDM analysis results.

Table 20. STM6-40 iteration vs. population size SDM case.

Number of Runs	Population Size	Number of Iterations	RMSE	Run Time (s)
5	20	100	3.5802×10^{-3}	0.0689
	30	100	3.9856×10^{-3}	0.1013
	50	100	2.1070×10^{-2}	0.1630
	60	100	5.2572×10^{-3}	0.1971
	80	100	6.1873×10^{-3}	0.2573
	100	100	2.0879×10^{-3}	0.3170
	50	300	1.7571×10^{-3}	0.4678
	50	500	2.5284×10^{-3}	0.7864
	50	700	1.7298×10^{-3}	1.0971
	50	900	1.7298×10^{-3}	1.4028
	50	1100	1.7298×10^{-3}	1.8029
	50	1300	1.7298×10^{-3}	2.1277
	50	1500	1.7298×10^{-3}	2.4678

Table 21. STM6-40 iteration vs. population size DDM case.

Number of Runs	Population Size	Number of Iterations	RMSE	Run Time (s)
5	20	100	3.30×10^{-3}	0.1077
	30	100	3.49×10^{-2}	0.1556
	50	100	2.25×10^{-2}	0.2524
	60	100	1.45×10^{-2}	0.2968
	80	100	1.87×10^{-3}	0.4020
	100	100	1.80×10^{-3}	0.4867
	50	300	5.9485×10^{-3}	0.7507
	50	500	2.7450×10^{-3}	1.2540
	50	700	3.3232×10^{-3}	1.7156
	50	900	2.2871×10^{-3}	2.2240
	50	1100	2.7683×10^{-3}	2.5609
	50	1300	1.7029×10^{-3}	3.0451
	50	1500	1.6892×10^{-3}	3.5337

Table 22 examines the performance of both PV models in terms of CPU execution times on two processors: Intel Core i9 and AMD Ryzen 5 5600H. In terms of computational performance, the Intel Core i9 consistently outperformed the AMD Ryzen 5 5600H. For the SDM, the Intel Core i9 required 2.6012 s, significantly faster than the 4.9071 s taken by the AMD Ryzen 5. Similarly, for the DDM, the Intel Core i9 completed the task in 4.4091 s, whereas the AMD Ryzen 5 required 8.0901 s. These results highlight the Intel Core i9's efficiency and better optimization for the given workload.

Table 22. STM6-40 CPU execution time.

Number of Runs	Model	Population Size/ Number of Iterations	RMSE	Intel Core i9 CPU Execution Time (s)	AMD Ryzen 5 5600H CPU Execution Time (s)
5	SDM	100/1100	1.7298×10^{-3}	2.6012	4.9071
	DDM	100/1100	1.6892×10^{-3}	4.4091	8.0901

3.6. Multijunction Solar Cells (MJSC)

Three datasets were measured for an InGaP/InGaAs/Ge multijunction solar cell (MJSC) at three different temperatures. The MJSC, with a 1 cm^2 active area, was tested under one-sun irradiance at three temperature conditions: Case A (41.5 °C), Case B (51.3 °C), and Case C (61.6 °C). These measurements were conducted in natural sunlight using the RELab system. The datasets were utilized to extract either five or seven parameters of the MJSC based on the SDMs and DDMs [33]. The associated errors and current values derived from the three cases are presented in Tables S6–S8.

The PSA was compared with other methods, including BWOA, the Chameleon Swarm Algorithm (CSA) [33], and the 5P method. The results obtained using the PSA for both the SDMs and DDMs are summarized in Table 23. To evaluate algorithm performance, the statistical tests RMSE and MBE were employed.

Table 23. RMSE of the MJSC photovoltaic cell and the extracted parameters for different algorithms.

Algorithm	Model	I_{ph} (A)	I_{sd1}(μA)	n_1	R_s(Ω)	R_{sh}(Ω)	I_{sd2}(μA)	n_2	RMSE	MBE	STD
Case A: MJSC at 41.5 °C											
PSA	**SDM**	**0.0120751156**	$\mathbf{3.78228069 \times 10^{-34}}$	**1.28263417**	**1.56797153**	**65,594.5326**	**-**	**-**	$\mathbf{7.82737990023 \times 10^{-5}}$	$\mathbf{3.448275 \times 10^{-10}}$	$\mathbf{7.965936 \times 10^{-5}}$
BWOA	SDM	0.0121089428	$2.3858024 \times \mathbf{10^{-38}}$	1.131912325	2.066927686	18,039.8683	**-**	**-**	$8.9120260701 \times 10^{-5}$	$2.96206896 \times 10^{-7}$	9.069773×10^{-05}
CSA	SDM	0.011344	$1 \times \mathbf{10^{-40}}$	1.06130028	2.29655935	13,448.8398	**-**	**-**	$1.0186359636 \times 10^{-4}$	$7.80407241 \times 10^{-4}$	1.036667×10^{-4}
5P	SDM	0.011347	$1.671098 \times \mathbf{10^{-35}}$	1.230082	2.54146	22,3038.6	**-**	**-**	$3.8193717771 \times 10^{-4}$	$9.25141724 \times 10^{-4}$	3.306110×10^{-4}
PSA	**DDM**	**0.0120716502**	$\mathbf{1.58366151 \times 10^{-14}}$	**3.9999999**	**1.66262677**	**100,000.0**	$\mathbf{2.4774912 \times 10^{-35}}$	**1.23645540**	$\mathbf{7.80311999761 \times 10^{-5}}$	$\mathbf{3.4482758 \times 10^{-10}}$	$\mathbf{7.941187 \times 10^{-5}}$
BWOA	DDM	0.0121078162	$2.3606317 \times \mathbf{10^{-38}}$	1.1319214	2.062849604	18,690.3364	$2.8573381 \times \mathbf{10^{-40}}$	1.1319214800337	$8.912393251 \times 10^{-5}$	$-1.2862068 \times 10^{-6}$	9.069190×10^{-5}
CSA	DDM	0.0121315693	$1 \times \mathbf{10^{-40}}$	1.0613296	2.293901938	14,059.1135	$1.4088443 \times \mathbf{10^{-22}}$	2.296108251161	$1.0088314434 \times 10^{-4}$	$-2.6548275 \times 10^{-6}$	1.026332×10^{-4}
Case B: MJSC at 51.3°C											
PSA	**SDM**	**0.0123378435**	$\mathbf{1.09880130 \times 10^{-31}}$	**1.3122855**	**1.47116460**	**99,999.9724**	**-**	**-**	$\mathbf{7.53874445507 \times 10^{-5}}$	$\mathbf{6.8965517 \times 10^{-10}}$	$\mathbf{7.672188 \times 10^{-5}}$
BWOA	SDM	0.012382444	$8.6249412 \times \mathbf{10^{-37}}$	1.1163539	2.126612851	15,948.7273133	-	-	$9.4508986452 \times 10^{-5}$	$-5.4137931 \times 10^{-7}$	9.628231×10^{-5}
CSA	SDM	0.012411337	$1 \times \mathbf{10^{-40}}$	1.0011103	2.499590378	9954.65622131	**-**	**-**	$1.0186359636 \times 10^{-4}$	$2.67241379 \times 10^{-7}$	1.253248×10^{-4}
5P	SDM	0.01235383	$6.068981 \times \mathbf{10^{-36}}$	1.144978	2.655795	23,011.52	**-**	**-**	$3.2891868735 \times 10^{-4}$	$1.80126206 \times 10^{-4}$	2.801111×10^{-4}
PSA	**DDM**	**0.0123407585**	$\mathbf{5.51543091 \times 10^{-14}}$	**4.0**	**1.56872922**	**99,995.2907**	$\mathbf{9.3577793 \times 10^{-33}}$	**1.26596321**	$\mathbf{7.51615759370 \times 10^{-5}}$	$\mathbf{2.75862068 \times 10^{-9}}$	$\mathbf{7.649272 \times 10^{-5}}$
BWOA	DDM	0.012293709	$1.0091098 \times \mathbf{10^{-40}}$	1.0091098	2.528407915	100,000	$1.0091098 \times \mathbf{10^{-40}}$	1.1319214800337	$1.3016613407 \times 10^{-4}$	$-1.4912937 \times 10^{-3}$	2.076396×10^{-3}
CSA	DDM	0.012442539	$1 \times \mathbf{10^{-40}}$	1.0011646	2.46267704	9760.66361	$2.6221828 \times \mathbf{10^{-17}}$	2.9999999990733	$1.1884566045 \times 10^{-4}$	$-7.6765517 \times 10^{-6}$	1.208481×10^{-4}
Case C: MJSC at 61.6 °C											
PSA	**SDM**	**0.0125823702**	$\mathbf{8.90988273 \times 10^{-30}}$	**1.32195972**	**1.42076389**	**76848.8449**	**-**	**-**	$\mathbf{6.77540506680 \times 10^{-5}}$	$\mathbf{7.1428571 \times 10^{-10}}$	$\mathbf{6.899791 \times 10^{-5}}$
BWOA	SDM	0.012646374	$1.7564737 \times 10^{-35}$	1.09267609	2.173965484	10,764.6947093	-	-	$9.6816850966 \times 10^{-5}$	$-7.2857142 \times 10^{-7}$	9.859123×10^{-5}
CSA	SDM	0.012666374	$1.8464737 \times 10^{-35}$	1.095016268	2.16696549	10,780.7048	**-**	**-**	$5.6947272673 \times 10^{-4}$	$-3.5050500 \times 10^{-4}$	4.570632×10^{-4}
5P	SDM	0.01258102	1.85812×10^{-30}	1.290024	2.212063	64,141.71	**-**	**-**	$3.2115343257 \times 10^{-4}$	1.774900×10^{-4}	2.725626×10^{-4}
PSA	**DDM**	**0.0125823703**	$\mathbf{2.91443456 \times 10^{-22}}$	**4.0**	**1.42076409**	**76,848.5295**	$\mathbf{8.9098505 \times 10^{-30}}$	**1.32195965**	$\mathbf{6.77540506673 \times 10^{-5}}$	$\mathbf{-3.571428 \times 10^{-10}}$	$\mathbf{6.899659 \times 10^{-5}}$

The results of PSA algorithm are highlighted in bold.

The PSA consistently outperformed BWOA and CSA across all temperatures and models. It achieved significant RMSE reductions, with improvements of up to 42.26% over BWOA and 88.11% over CSA. Additionally, the PSA demonstrated exceptional accuracy and stability in parameter extraction, particularly for I_{ph}, I_{sd1}, R_{sh}, and $n1$, even in complex scenarios and at higher temperatures.

At Case A (41.5 °C), for the SDM, PSA achieved the lowest RMSE of 7.8273×10^{-5}, improving over BWOA and CSA by 12.17% and 23.17%, respectively. The extracted parameters from PSA showed minimal variation, particularly for the photogenerated current (I_{ph}), which differed by only 0.12% compared to other algorithms. Moreover, PSA outperformed its counterparts in accurately estimating reverse saturation current (I_{sd1}) and shunt resistance (R_{sh}), with deviations significantly lower than those produced by CSA. In the DDM, PSA continued to lead with an RMSE of 7.803119×10^{-5}, reflecting a 12.42% improvement over BWOA and a 22.67% improvement over CSA. The extracted parameters from PSA were more consistent, with R_s and $n1$ values remaining stable, while R_{sh} and I_{sd1} exhibited less deviation than those obtained using the CSA.

At a higher temperature of 51.3 °C (Case B), PSA maintained its leading performance. For the SDM, it achieved an RMSE of 7.538744×10^{-5}, marking a significant improvement of 20.23% over BWOA and 25.99% over CSA. The parameters extracted by PSA, particularly I_{ph} and R_s, remained stable and accurate despite the increased thermal stress. The differences in I_{ph} were slightly more pronounced, with PSA differing from BWOA by 0.10% and from CSA by 0.14%. These values reflect a slight increase in variability as temperature rises, which is expected due to the thermal impact on solar cell behavior. Additionally, the ideality factor ($n1$) showed reduced variation, ensuring reliable modeling under these conditions. In the DDM, PSA achieved an RMSE of 7.5161575×10^{-5}, which was dramatically better than the BWOA and CSA, with improvements of 42.26% and 36.74%, respectively. The extracted parameters for I_{sd1} and R_{sh} were notably more consistent, avoiding the exaggerated deviations seen with CSA. Furthermore, PSA managed to align its R_s and $n2$ values with expected trends, demonstrating its robustness in handling the complexity of the DDM.

Under the most challenging conditions at 61.6 °C in Case C, PSA's ability to maintain accuracy was evident. For the SDM, it achieved an RMSE 6.7754050×10^{-5}, a 30.01% improvement over BWOA and an impressive 88.11% over CSA. The extracted parameters, including I_{ph} and I_{sd1}, remained highly accurate, and the differences in I_{ph} increased further, with PSA differing from BWOA by 0.20% and from CSA by 0.30%. These results highlight the challenges posed by high-temperature conditions, where thermal noise and variability in the dataset can affect parameter extraction, reflecting PSA's capability to handle the noise and variability introduced by higher temperatures. Additionally, R_{sh} and $n1$ were extracted with significantly reduced deviations compared to CSA, ensuring a more precise representation of the solar cell's behavior.

The best results obtained to extract the parameters of the MJSC for three temperatures are obtained using the PSA for both models by comparing the obtained results with the other algorithms in terms of RMSE, MBE, and STD.

The reduction in RMSE between the temperatures is significant (Figure 24). At 51.3 °C, the RMSE improves by 3.68% compared to 41.5 °C. As the temperature increases to 61.6 °C, the RMSE further improves by 10.12% compared to 51.3 °C. Overall, the RMSE improves by 13.43% when comparing the highest temperature (61.6 °C) to the lowest (41.5 °C).

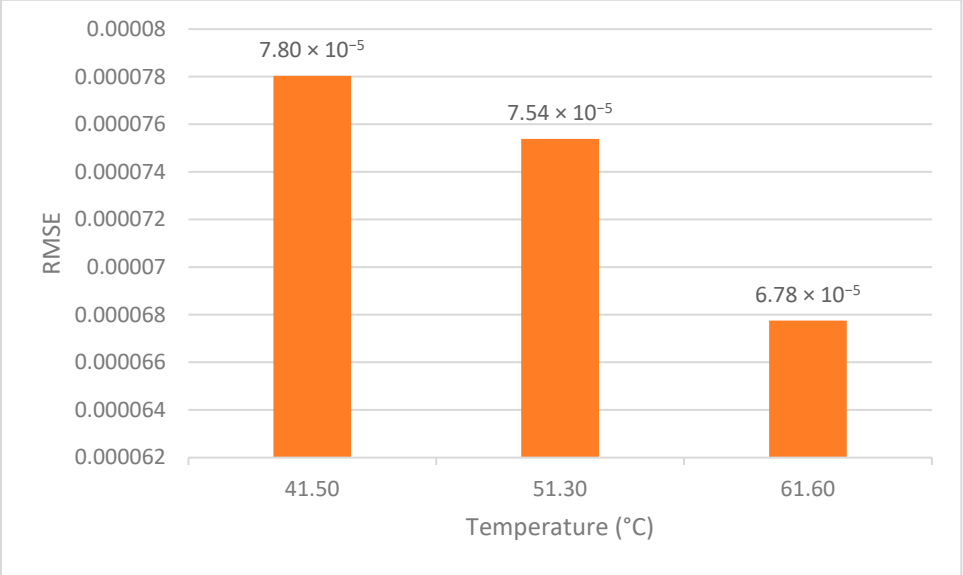

Figure 24. RMSE for PSA with regard to temperature.

Table 24 compares the performance of the SDM and DDM across three temperature scenarios—Case A (41.5 °C), Case B (51.3 °C), and Case C (61.6 °C)—on Intel Core i9 and AMD Ryzen 5 5600H processors. As temperature increases, run times vary slightly, with the SDM consistently requiring less time than the DDM. For example, in Case A, the SDM's run time is 3.1843 s compared to 4.099 s for the DDM, with similar trends in Cases B and C. The Intel Core i9 consistently outperforms the AMD Ryzen 5 in CPU execution times, completing tasks in as little as 2.9817 s (SDM, Case C) compared to Ryzen's 4.0331 s. Notably, the Intel Core i9's CPU execution times are slightly lower than the total run times due to its efficient handling of task overheads, while the AMD Ryzen 5's results are closer to the total run times, highlighting the i9's superior computational efficiency.

Table 24. MJSC CPU execution time for all cases.

Number of Runs	Model	Population Size/ Number of Iterations	RMSE	Run Time (s)	Intel Core i9 CPU Execution Time (s)	AMD Ryzen 5 5600H CPU Execution Time (s)
5	Case A: MJSC at 41.5 °C					
	SDM	100/1100	7.8273×10^{-5}	3.1843	3.0124	4.9071
	DDM	100/1100	7.8031×10^{-5}	4.099	3.8558	8.0901
	Case B: MJSC at 51.3 °C					
	SDM	100/1100	7.5387×10^{-5}	3.3770	3.1109	4.5467
	DDM	100/1100	7.5161×10^{-5}	4.6077	4.441	8.2144
	Case C: MJSC at 61.6 °C					
	SDM	100/1100	6.7754×10^{-5}	3.0998	2.9817	4.0331
	DDM	100/1100	6.7754×10^{-5}	4.1012	3.9912	8.5455

4. Conclusions

This research introduced a novel metaheuristic algorithm, the PID-based search algorithm (PSA), for extracting parameters of PV cells and panels. By optimizing the proportion, integral, and differential parameters and relocating initial error calculations outside the iterative loop, the PSA demonstrated enhanced exploration, exploitation, and computational

efficiency. The algorithm's performance was validated using both SDMs and DDMs across datasets, including RTC France, aSi, mSi, and various photovoltaic panels. Compared to 26 state-of-the-art algorithms using four statistical tests, the PSA consistently achieved superior or comparable accuracy.

In particular, the RTC France DDM yielded RMSE improvements of 0.67–2.10%, while the mSi cell demonstrated up to 40.9% RMSE improvement compared with the ISOA, which obtained 4.4383×10^{-4}, while PSA obtained 2.6216×10^{-4} with execution times of 3.94 s for the SDM and 5.57 s for the DDM. Additionally, the PSA required smaller population sizes and fewer iterations, underscoring its practicality. Testing the PSA under diverse conditions, including the multijunction solar cell (MJSC) at three different temperatures (41.5 °C, 51.3 °C, and 61.6 °C), demonstrated its reliability and effectiveness. It achieved average RMSE improvements of 12.17–12.42% across BWOA and CSA and execution times as low as 2.98 s for SDMs and 4.10 s for DDMs. The PSA was also tested using two CPUs—Intel Core i9 and AMD Ryzen 5 5600H—to evaluate CPU efficiency, with Intel Core i9 consistently achieving faster results. The extracted parameters, such as I_{ph}, I_{sd1}, R_{sh}, and n_1, were highly accurate and stable, even under high thermal stress.

To enhance the robustness and practical applicability of the PID-based Search Algorithm (PSA) in real-world environments, future work will integrate strategies to handle uncertainties such as sensor noise, environmental variability, and data imperfections commonly encountered in field conditions. One promising direction is the use of stochastic perturbation models that simulate noisy I–V measurements, enabling PSA to be tested and calibrated for resilience under such conditions. Moreover, the control-based nature of PSA allows for adaptive adjustment of the PID coefficients (K_p, K_i, K_d) in response to the algorithm's convergence behavior, which can improve stability and prevent premature convergence. Building on these enhancements, future work will apply PSA to emerging challenges, including perovskite solar cell modeling, triple-diode models, and PV panel power output prediction, further demonstrating its accuracy, flexibility, and potential for advancing complex photovoltaic systems.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app15137403/s1, Table S1. RTC silicon PV cell dataset and current results; Table S2. Amorphous silicon photovoltaic cell dataset and current results; Table S3. Monocrystalline PV cell dataset and current results; Table S4. PVM 752 GaAs thin-film PV panel dataset and current results; Table S5. STM4-60 monocrystalline PV panel dataset and current results; Table S6. Multi-junction solar cell dataset at 41.5 °C and current results; Table S7. Multi-junction solar cell dataset at 51.3 °C and current results; Table S8. Multi-junction solar cell dataset at 61.1 °C and current results.

Author Contributions: Conceptualization, A.B. and D.T.C.; methodology, A.B., O.A., D.T.C., and P.A.C.; software, A.B.; validation, A.B. and O.A.; investigation, A.B.; data curation, A.B.; writing—original draft preparation, A.B. and O.A.; writing—review and editing, A.B., O.A., D.T.C., and P.A.C.; visualization, A.B.; supervision, D.T.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We present the data in the Supplementary Materials.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tooryan, F.; HassanzadehFard, H.; Collins, E.R.; Jin, S.; Ramezani, B. Smart integration of renewable energy resources, electrical, and thermal energy storage in microgrid applications. *Energy* **2020**, *212*, 118716. [CrossRef]
2. Piliougine, M.; Guejia-Burbano, R.; Petrone, G.; Sánchez-Pacheco, F.; Mora-López, L.; Sidrach-de-Cardona, M. Parameters extraction of single diode model for degraded photovoltaic modules. *Renew Energy* **2021**, *164*, 674–686. [CrossRef]
3. EU Commission. *Committing to Climate-Neutrality by 2050: Commission Proposes European Climate Law and Consults on the European Climate Pact*; EU Commission: Brussels, Belgium, 2020.
4. Cotfas, D.; Cotfas, P.; Mahmoudinezhad, S.; Louzazni, M. Critical factors and parameters for hybrid photovoltaic-thermoelectric systems; review. *Appl. Therm. Eng.* **2022**, *215*, 118977. [CrossRef]
5. Yang, B.; Wang, J.; Zhang, X.; Yu, T.; Yao, W.; Shu, H.; Zeng, F.; Sun, L. Comprehensive overview of meta-heuristic algorithm applications on PV cell parameter identification. *Energy Convers. Manag.* **2020**, *208*, 112595. [CrossRef]
6. Elkholy, A.; Abou El-Ela, A. Optimal parameters estimation and modelling of photovoltaic modules using analytical method. *Heliyon* **2019**, *5*, e02137. [CrossRef]
7. Li, S.; Gong, W.; Gu, Q. A comprehensive survey on meta-heuristic algorithms for parameter extraction of photovoltaic models. *Renew. Sustain. Energy Rev.* **2021**, *141*, 110828. [CrossRef]
8. Le, T.M.C.; Le, X.C.; Huynh, N.N.P.; Doan, A.T.; Dinh, T.V.; Duong, M.Q. Optimal power flow solutions to power systems with wind energy using a highly effective meta-heuristic algorithm. *Int. J. Renew. Energy Dev.* **2023**, *12*, 467–477. [CrossRef]
9. Navarro, M.A.; Oliva, D.; Ramos-Michel, A.; Haro, E.H. An analysis on the performance of metaheuristic algorithms for the estimation of parameters in solar cell models. *Energy Convers. Manag.* **2023**, *276*, 116523. [CrossRef]
10. Cotfas, D.; Cotfas, P.; Kaplanis, S. Methods to determine the dc parameters of solar cells: A critical review. *Renew. Sustain. Energy Rev.* **2013**, *28*, 588–596. [CrossRef]
11. Gu, Z.; Xiong, G.; Fu, X. Parameter Extraction of Solar Photovoltaic Cell and Module Models with Metaheuristic Algorithms: A Review. *Sustainability* **2023**, *15*, 3312. [CrossRef]
12. Sharma, A.; Sharma, A.; Averbukh, M.; Jately, V.; Rajput, S.; Azzopardi, B.; Lim, W.H. Performance investigation of state-of-the-art metaheuristic techniques for parameter extraction of solar cells/module. *Sci. Rep.* **2023**, *13*, 11134. [CrossRef]
13. Zhao, L.; Bi, S.; Li, J.; Wen, Y.; Zhang, H.; Zhang, D.; Lu, S.; Yin, P.; Shi, F.; Yan, J. Prussian blue analogues for advanced non-aqueous sodium ion batteries: Redox mechanisms, key challenges and modification strategies. *Energy Storage Mater.* **2025**, *78*, 104256. [CrossRef]
14. Yadav, D.; Singh, N.; Bhadoria, V.S.; Giri, N.C.; Cherukuri, M. A Novel Metaheuristic Jellyfish Optimization Algorithm for Parameter Extraction of Solar Module. *Int. Trans. Electr. Energy Syst.* **2023**, *2023*, 5589859. [CrossRef]
15. Vandrasi, R.K.; Sravana Kumar, B.; Devarapalli, R. Solar photo voltaic module parameter extraction using a novel Hybrid Chimp-Sine Cosine Algorithm. *Energy Sources Part A* **2022**, 1–20. [CrossRef]
16. Long, W.; Cai, S.; Jiao, J.; Xu, M.; Wu, T. A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models. *Energy Convers. Manag.* **2020**, *203*, 112243. [CrossRef]
17. Chen, X.; Yu, K. Hybridizing cuckoo search algorithm with biogeography-based optimization for estimating photovoltaic model parameters. *Sol. Energy* **2019**, *180*, 192–206. [CrossRef]
18. Ayyarao, T.S.; Kumar, P.P. Parameter estimation of solar PV models with a new proposed war strategy optimization algorithm. *Int. J. Energy Res.* **2022**, *46*, 7215–7238. [CrossRef]
19. Gao, Y. PID-based search algorithm: A novel metaheuristic algorithm based on PID algorithm. *Expert Syst. Appl.* **2023**, *232*, 120886. [CrossRef]
20. He, P.; Xi, X.; Li, S.; Qin, W.; Xing, C.; Yang, B. Radial Basis Function Based Meta-Heuristic Algorithms for Parameter Extraction of Photovoltaic Cell. *Processes* **2023**, *11*, 1606. [CrossRef]
21. Tajjour, S.; Chandel, S.S.; Malik, H.; Alotaibi, M.A.; Ustun, T.S. A Novel Metaheuristic Approach for Solar Photovoltaic Parameter Extraction Using Manufacturer Data. *Photonics* **2022**, *9*, 858. [CrossRef]
22. Abdel-Basset, M.; El-Shahat, D.; Chakrabortty, R.K.; Ryan, M. Parameter estimation of photovoltaic models using an improved marine predators algorithm. *Energy Convers. Manag.* **2021**, *227*, 113491. [CrossRef]
23. Belabbes, F.; Cotfas, D.T.; Cotfas, P.A.; Medles, M. Using the snake optimization metaheuristic algorithms to extract the photovoltaic cells parameters. *Energy Convers. Manag.* **2023**, *292*, 117373. [CrossRef]
24. Chen, X.; Wang, S.; He, K. Parameter estimation of various PV cells and modules using an improved simultaneous heat transfer search algorithm. *J. Comput. Electron.* **2024**, *23*, 584–599. [CrossRef]
25. Madhiarasan, M.; Cotfas, D.T.; Cotfas, P.A. Black Widow Optimization Algorithm Used to Extract the Parameters of Photovoltaic Cells and Panels. *Mathematics* **2023**, *11*, 967. [CrossRef]

26. Ghetas, M.; Elshourbagy, M. Parameters extraction of photovoltaic models using enhanced generalized normal distribution optimization with neighborhood search. *Neural Comput. Appl.* **2024**, *36*, 14035–14052. [CrossRef]
27. Xiong, G.; Li, L.; Mohamed, A.W.; Yuan, X.; Zhang, J. A new method for parameter extraction of solar photovoltaic models using gaining–sharing knowledge based algorithm. *Energy Rep.* **2021**, *7*, 3286–3301. [CrossRef]
28. Gao, X.; Cui, Y.; Hu, J.; Xu, G.; Wang, Z.; Qu, J.; Wang, H. Parameter extraction of solar cell models using improved shuffled complex evolution algorithm. *Energy Convers. Manag.* **2018**, *157*, 460–479. [CrossRef]
29. Cotfas, D.T.; Deaconu, A.M.; Cotfas, P.A. Hybrid successive discretisation algorithm used to calculate parameters of the photovoltaic cells and panels for existing datasets. *IET Renew. Power Gener.* **2021**, *15*, 3661–3687. [CrossRef]
30. Choulli, I.; Elyaqouti, M.; Saadaoui, D.; Lidaighbi, S.; Elhammoudy, A.; Abazine, I. DIWJAYA: JAYA driven by individual weights for enhanced photovoltaic model parameter estimation. *Energy Convers. Manag.* **2024**, *305*, 118258. [CrossRef]
31. Ali, F.; Sarwar, A.; Bakhsh, F.I.; Ahmad, S.; Shah, A.A.; Ahmed, H. Parameter extraction of photovoltaic models using atomic orbital search algorithm on a decent basis for novel accurate RMSE calculation. *Energy Convers. Manag.* **2023**, *277*, 116613. [CrossRef]
32. Abdel-Basset, M.; El-Shahat, D.; Sallam, K.M.; Munasinghe, K. Parameter extraction of photovoltaic models using a memory-based improved gorilla troops optimizer. *Energy Convers. Manag.* **2022**, *252*, 115134. [CrossRef]
33. Cotfas, D.T.; Madhiarasan, M.; Cotfas, P.A. Extraction of the multijunction solar cell parameters using two metaheuristic algorithms. *IEEE Access* **2024**, *12*, 109634–109656. [CrossRef]
34. Easwarakhanthan, T.; Bottin, J.; Bouhouch, I.; Boutrit, C. Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers. *Int. J. Sol. Energy* **1986**, *4*, 1–12. [CrossRef]
35. Oliva, D.; Abd El Aziz, M.; Hassanien, A.E. Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm. *Appl. Energy* **2017**, *200*, 141–154. [CrossRef]
36. Arabshahi, M.; Torkaman, H.; Keyhani, A. A method for hybrid extraction of single-diode model parameters of photovoltaics. *Renew Energy* **2020**, *158*, 236–252. [CrossRef]
37. Jordehi, A.R. Parameter estimation of solar photovoltaic (PV) cells: A review. *Renew. Sustain. Energy Rev.* **2016**, *61*, 354–371. [CrossRef]
38. Ahmadianfar, I.; Gong, W.; Heidari, A.A.; Golilarz, N.A.; Samadi-Koucheksaraee, A.; Chen, H. Gradient-based optimization with ranking mechanisms for parameter identification of photovoltaic systems. *Energy Rep.* **2021**, *7*, 3979–3997. [CrossRef]
39. Singla, M.K.; Gupta, J.; Alsharif, M.H.; Kim, M.-K.; Aljaidi, M.; Safaraliev, M. A robust multi-objective optimization algorithm for accurate parameter estimation for solar cell models. *Soft Comput.* **2024**, *28*, 11265–11277. [CrossRef]
40. Ayyarao, T.S.; Kishore, G.I. Parameter estimation of solar PV models with artificial humming bird optimization algorithm using various objective functions. *Soft Comput.* **2024**, *28*, 3371–3392. [CrossRef]
41. Aribia, H.B.; El-Rifaie, A.M.; Tolba, M.A.; Shaheen, A.; Moustafa, G.; Elsayed, F.; Elshahed, M. Growth Optimizer for Parameter Identification of Solar Photovoltaic Cells and Modules. *Sustainability* **2023**, *15*, 7896. [CrossRef]
42. Qaraad, M.; Amjad, S.; Hussein, N.K.; Badawy, M.; Mirjalili, S.; Elhosseini, M.A. Photovoltaic parameter estimation using improved moth flame algorithms with local escape operators. *Comput. Electr. Eng.* **2023**, *106*, 108603. [CrossRef]
43. Xu, S.; Qiu, H. A modified stochastic fractal search algorithm for parameter estimation of solar cells and PV modules. *Energy Rep.* **2022**, *8*, 1853–1866. [CrossRef]
44. Jian, X.; Weng, Z. A logistic chaotic JAYA algorithm for parameters identification of photovoltaic cell and module models. *Optik* **2020**, *203*, 164041. [CrossRef]
45. Saadaoui, D.; Elyaqouti, M.; Assalaou, K.; Lidaighbi, S.; Arjdal, E.; Choulli, I.; Elhammoudy, A.; Abazine, I. A hybrid optimization algorithm to identify unknown parameters of photovoltaic models under varying operating conditions. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108544. [CrossRef]
46. Kullampalayam Murugaiyan, N.; Chandrasekaran, K.; Manoharan, P.; Derebew, B. Leveraging opposition-based learning for solar photovoltaic model parameter estimation with exponential distribution optimization algorithm. *Sci. Rep.* **2024**, *14*, 528. [CrossRef] [PubMed]
47. Kharchouf, Y.; Herbazi, R.; Chahboun, A. Parameter's extraction of solar photovoltaic models using an improved differential evolution algorithm. *Energy Convers. Manag.* **2022**, *251*, 114972. [CrossRef]
48. Madhiarasan, M.; Cotfas, D.T.; Cotfas, P.A. Barnacles Mating Optimizer Algorithm to Extract the Parameters of the Photovoltaic Cells and Panels. *Sensors* **2022**, *22*, 6989. [CrossRef]
49. Xiong, G.; Zhang, J.; Shi, D.; Yuan, X. Application of supply-demand-based optimization for parameter extraction of solar photovoltaic models. *Complexity* **2019**, *2019*, 1–22. [CrossRef]
50. Tong, N.T.; Pora, W. A parameter extraction technique exploiting intrinsic properties of solar cells. *Appl. Energy* **2016**, *176*, 104–115. [CrossRef]

51. Hu, Z.; Gong, W.; Li, S. Reinforcement learning-based differential evolution for parameters extraction of photovoltaic models. *Energy Rep.* **2021**, *7*, 916–928. [CrossRef]
52. Jordehi, A.R. Enhanced leader particle swarm optimisation (ELPSO): An efficient algorithm for parameter estimation of photovoltaic (PV) cells and modules. *Sol. Energy* **2018**, *159*, 78–87. [CrossRef]