Artificial Intelligence Topics and Applications **1**(1), 2025, 59–72

Hybrid Atom Search and Salp Swarm Optimization for Parameter Estimation in Solar Photovoltaic Modules

Mohamed Eisa[*,1], **Yousif A. Alhaj**[2,3,4], **Mijahed Aljober**[3], and **Hamzah A. Qasem**[2,5]

[1]Faculty of Management Technology and Information Systems, Portsaid University, Egypt.
[2]Artificial Intelligence Department, 21 September University for Medical and Applied Sciences, Sana'a, Yemen.
[3]Artificial Intelligence Department, Modern Specialized university, Sana'a, Yemen.
[4]Cybersecurity Department, Al-Saeeda University, Sana'a, Yemen.
[5]Computer Science and Information Technology, International University of Technology Twintech, Yemen.

Abstract The efficient design of solar photovoltaic (PV) modules relies on accurately estimating the internal parameters of their equivalent circuit models. This task involves solving highly nonlinear and multimodal optimization problems. To address this challenge, this paper proposes a hybrid metaheuristic approach (ASOSSA) that integrates Atom Search Optimization (ASO) with Salp Swarm Algorithm (SSA) to improve parameter estimation for single and double diode PV models. By combining ASO's global search capabilities with SSA's dynamic position updates, the proposed method enhances convergence and avoids local optima. Simulation results, benchmarked against various state-of-the-art algorithms, demonstrate ASOSSA's effectiveness and robustness in producing precise parameter estimates, even under noisy measurement conditions.

Keywords: Solar Cells; Photovoltaic Modules; Atom Search Optimization; Salp Swarm Algorithm; Parameter Estimation

1 Introduction

The rising global energy demand, coupled with the depletion of fossil fuel resources, has intensified the pursuit of clean and sustainable alternatives. Among these, photovoltaic (PV) energy has gained significant attention due to its global availability, emission-free nature, and ease of deployment [1]. PV systems convert solar radiation into electrical power using solar cells (SCs), but their performance is strongly influenced by external environmental factors such as irradiance and temperature. Furthermore, the high cost

*Corresponding author: mohamed_eisa@himc.psu.edu.eg

Received: 25 March 2025; Revised: 1 May 2025; Accepted: 7 May 2025
Available Online: 12 May 2025
Artificial Intelligence Topics and Applications

of installation and frequent maintenance, primarily due to technological immaturity and environmental degradation, remains a major barrier to large-scale adoption [2].

Optimizing PV module performance requires accurate modeling of the electrical behavior of solar cells. This is typically achieved through equivalent circuit models, most commonly the Single Diode (SD) and Double Diode (DD) models [3]. These models rely on parameter estimation to simulate the nonlinear current-voltage (I–V) relationship, involving variables such as photo-generated current, diode saturation current, ideality factor, and parasitic resistances. The precision of this parameter identification is crucial to the effectiveness of PV system design and simulation [4].

Mathematically, estimating these parameters can be formulated as an optimization problem, where the objective is to minimize the Root Mean Square Error (RMSE) between measured and simulated current values [5]. However, this optimization landscape is complex—highly nonlinear and multimodal—often leading traditional deterministic methods such as Newton-Raphson, Lambert W-based models, and iterative curve fitting to converge prematurely or become trapped in local optima [6–9].

Metaheuristic algorithms provide a robust alternative by enabling global search over rugged landscapes. Approaches like Particle Swarm Optimization (PSO) [10], Harmony Search (HS) [11], Genetic Algorithms (GA) [12], firefly algorithm [13], and artificial gorilla troops optimizer [14] have all demonstrated promising results in PV parameter estimation. Nevertheless, according to the No-Free-Lunch theorem [15], no single optimization technique can universally solve all classes of problems optimally.

To address this, we introduce a hybrid metaheuristic called ASOSSA, which combines Atom Search Optimization (ASO) [16] with the Salp Swarm Algorithm (SSA) [17]. ASO contributes strong global exploration capabilities, while SSA enhances convergence through swarm-based position updates. The integration aims to balance exploration and exploitation, improving solution accuracy and avoiding local stagnation.

We validate the ASOSSA method on benchmark datasets using both SD and DD models. Experimental results show that ASOSSA outperforms several existing techniques in terms of accuracy, robustness, and resistance to noisy data, making it a strong candidate for real-world PV modeling and parameter estimation tasks.

The rest of this paper is organized as follows: Section 2 introduces the fundamental concepts of photovoltaic modeling and optimization, including the SD and DD models, ASO, and SSA. Section 3 presents the proposed ASOSSA algorithm and details its integration mechanism. Section 4 discusses the experimental setup, benchmark datasets, and performance metrics, followed by a comprehensive analysis of the results. Finally, Section 5 concludes the study.

2 Background

2.1 Photovoltaic Models

Accurate modeling of photovoltaic (PV) cells is fundamental in optimizing their design and performance. A mathematical representation of the internal parameters of solar cells (SCs) is essential to this task. Among the most widely accepted models are those based on equivalent electronic circuits, notably the Single Diode (SD) and Double Diode (DD) models [18–22]. These models serve not only for electrical characterization but also enable the formulation of parameter identification as an optimization problem. In the following subsections, we detail both models, highlighting their structure, governing equations, and

relevance to optimization-based parameter estimation.

2.1.1 Single Diode Model

The SD model represents one of the simplest yet most commonly used configurations for PV cell modeling. It consists of a current source representing the photo-generated current, a single diode modeling the p-n junction, and two resistive elements: a series resistor R_s and a shunt resistor R_{sh}, to account for internal losses. The diode incorporates a non-ideal behavior parameterized by an ideality factor n.

The output current I of the SD model is expressed as:

$$I = I_{ph} - I_d - I_{sh} \tag{1}$$

where I_{ph} is the photo-generated current, I_d is the diode current, and I_{sh} is the current through the shunt resistor. Using the Shockley diode equation, the diode current I_d is given by:

$$I = I_{ph} - I_s \left[\exp\left(\frac{q(V + IR_s)}{nkT}\right) - 1\right] - \frac{V + IR_s}{R_{sh}} \tag{2}$$

In this expression, I_s denotes the diode saturation current, V is the terminal voltage, q is the electron charge, k is Boltzmann's constant, and T is the absolute temperature of the cell in Kelvin. The five parameters to be estimated in this model—I_{ph}, I_s, n, R_s, and R_{sh}—significantly influence the model's accuracy and the overall performance prediction of the PV system.

2.1.2 Double Diode Model

The DD model extends the SD model by incorporating an additional diode. This second diode is introduced to account for recombination effects and other non-ideal behaviors not captured by the SD model. Like the SD model, it includes a photo-generated current source, two series diodes, and resistive losses through R_s and R_{sh}.

The output current for the DD model is defined as:

$$I = I_{ph} - I_{d1} - I_{d2} - I_{sh} \tag{3}$$

where I_{d1} and I_{d2} represent the currents through the first and second diodes, respectively. These components are modeled using modified Shockley equations:

$$I = I_{ph} - I_{s1}\left[\exp\left(\frac{q(V + IR_s)}{n_1 kT}\right) - 1\right] - I_{s2}\left[\exp\left(\frac{q(V + IR_s)}{n_2 kT}\right) - 1\right] - \frac{V + IR_s}{R_{sh}} \tag{4}$$

In this context, I_{s1} and I_{s2} denote the saturation currents of the two diodes, while n_1 and n_2 are their corresponding ideality factors. The remaining parameters are defined as in the SD model. The DD model requires the estimation of seven parameters: I_{ph}, I_{s1}, I_{s2}, n_1, n_2, R_s, and R_{sh}—making it more comprehensive and accurate for high-fidelity modeling of real solar cells.

The parameter bounds outlined in Table 1 are chosen based on existing literature [13,38–41], with slight tolerances to maintain a feasible search space while ensuring physical relevance. This is essential for accurate optimization results that reflect real-world solar cell behavior. For instance, Laudani et al. [23] analyzed valid physical ranges for each parameter, providing further insight into appropriate boundary selections.

Table 1: Parameter bounds for the SD and DD photovoltaic models

Parameter	Lower Bound	Upper Bound
I_{ph} (A)	0	1
I_{sd} / I_{sd1} / I_{sd2} (A)	0	0.5
n, n_1, n_2	1	2
R_s (Ω)	0	0.5
R_{sh} (Ω)	0	100

2.2 Solar Cell Parameter Estimation as an Optimization Task

The SD and DD photovoltaic models rely on several internal parameters that must be accurately estimated to ensure precise performance simulation. This estimation process can be framed as an optimization problem, where the objective is to minimize the error between the modeled and actual output current of a real solar cell.

For both models, an error-based objective function can be derived from Equations (2) and (4), evaluating how closely the simulated current aligns with measured data. Let I_m and V_m represent the measured current and voltage values from experimental or manufacturer-provided datasets. For a candidate solution x (i.e., the parameter vector), the error is quantified by comparing the simulated current $I_{sim}(x, V_m)$ against the real measurements I_m.

To capture this deviation, we define the objective function using the Root Mean Square Error (RMSE):

$$\text{RMSE}(x) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(I_m^{(i)} - I_{sim}(x, V_m^{(i)})\right)^2} \tag{5}$$

Here, N is the number of data points in the dataset. The optimization algorithm aims to minimize the RMSE by iteratively refining the parameter vector x, seeking the best match between the modeled and actual current outputs.

In practice, the input dataset may be affected by noise or measurement inaccuracies, which introduce multi-modality and ruggedness to the objective function landscape. These characteristics pose challenges for traditional optimization techniques, making robust metaheuristic methods particularly suitable for this parameter identification problem [23, 24].

2.3 Atom Search Optimization

Atom Search Optimization (ASO) is a physics-inspired metaheuristic algorithm introduced by [16], which simulates the motion and interaction of atoms based on molecular dynamics. In ASO, each atom represents a potential solution within the search space, and its quality is evaluated using a fitness function. Atoms with better fitness (heavier mass) exert stronger attraction on others, guiding the population toward optimal regions. Conversely, lighter atoms exhibit higher acceleration, which promotes exploration of new areas and prevents premature convergence.

The algorithm begins by initializing a population of N atoms, each with a position X_i in a D_X-dimensional search space. The fitness of each atom, Fit_i, is calculated and compared to the current global best fitness Fit_b. If $Fit_i < Fit_b$, the corresponding atom is selected as the new global best solution X_b.

The mass m_i of each atom is computed using the following normalized exponential function:

$$m_i(t) = \frac{M_i(t)}{\sum_{j=1}^{N} M_j(t)}, \quad M_i(t) = \exp\left(-\frac{Fit_i(t) - Fit_b(t)}{Fit_w(t) - Fit_b(t)}\right) \tag{6}$$

where $Fit_w(t)$ is the worst fitness in the current population and t is the current iteration.

The number of neighbors K_N considered for each atom decreases over time to shift the focus from exploration to exploitation:

$$K_N(t) = N - (N-2)\sqrt{\frac{t}{t_{\max}}} \tag{7}$$

To compute the interaction force F_i and constraint force G_i, the following expressions are used:

$$F_i^d(t) = \sum_{j \in K_{\text{best}}} \eta_j F_{ij}^d(t) \tag{8}$$

$$G_i^d(t) = \lambda(t)(X_b^d(t) - X_i^d(t)), \quad \lambda(t) = \beta \exp\left(-20\frac{t}{t_{\max}}\right) \tag{9}$$

Here, K_{best} is a subset of the best-performing atoms, η_j is a uniformly distributed random number in $[0, 1]$, and β is a control parameter for constraint force scaling.

The net acceleration for atom i is calculated using:

$$\begin{aligned} acc_i^d(t) &= \frac{F_i^d(t)}{m_i(t)} + \frac{G_i^d(t)}{m_i(t)} \\ &= -\alpha\left(1 - \frac{t-1}{t_{\max}}\right)^3 \exp\left(-20\frac{t}{t_{\max}}\right) \times \\ &\sum_{j \in K_{\text{best}}} \frac{r_i\left[2(h_{ij}(t))^{13} - (h_{ij}(t))^7\right]}{m_i(t)} \frac{(X_j^d(t) - X_i^d(t))}{\|X_i(t) - X_j(t)\|_2} \\ &+ \frac{\beta \exp\left(-20\frac{t}{t_{\max}}\right)(X_b^d(t) - X_i^d(t))}{m_i(t)} \end{aligned} \tag{10}$$

where r_i is a random value and $h_{ij}(t)$ denotes the distance between atoms i and j.

The position and velocity of each atom are updated as follows:

$$v_i^d(t+1) = \gamma v_i^d(t) + acc_i^d(t) \tag{11}$$

$$X_i^d(t+1) = X_i^d(t) + v_i^d(t+1) \tag{12}$$

The complete ASO procedure is summarized in Algorithm 1.

Algorithm 1 Atom Search Optimization (ASO)

Initialize a population X of N atoms in D_X dimensions
Set initial velocities and evaluate fitness values
repeat
  for each atom $X_i \in X$ **do**
    Evaluate fitness Fit_i
    Update best fitness Fit_b and corresponding solution X_b if improved
    Compute atom mass m_i using Eq. (6)
    Identify K_N neighbors using Eq. (7)
    Compute interaction force F_i using Eq. (8)
    Compute constraint force G_i using Eq. (9)
    Calculate acceleration acc_i using Eq. (10)
    Update velocity v_i using Eq. (11)
    Update position X_i using Eq. (12)
  end for
until convergence or $t \geq t_{\max}$
Return best solution X_b

2.3.1 Salp Swarm Algorithm (SSA)

SSA is an optimization technique proposed by [17] to solve different types of optimization problems. It emulates the salps' behavior in nature. Salps are a type of the Salpidae's family. Their moving behavior and weights have a high-water percentage. They use a contracting mechanism to move, pumping water through their bodies to change position. In the ocean, salps form salp chains, which improve foraging and enable efficient movement through coordinated changes in their arrangement [25].
The SSA algorithm begins by generating a random population then dividing this population into two groups (i.e. the leader and followers) based on their position in the chain, the front ones are called the leader, and the other ones are the followers. The position of the salps' is determined in n-dimensions that represent the problem's search domain and n denotes the problem's variables. The food source of the salps is considered as the target of the swarm. The position of the salps should be updated frequently, therefore, the salp leader using the following equation to do this update:

$$x_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j) \times c_2 + lb_j) & c_3 \leq 0 \\ F_j - c_1((ub_j - lb_j) \times c_2 + lb_j) & c_3 > 0 \end{cases} \tag{13}$$

where x_j^1 represents the position of the leader in j-th dimension. F_j is the food source in this dimension. The upper and lower bounds are represented by ub_j and lb_j, respectively. c_2 and c_3 are random variable in $[0, 1]$ to maintain the search domain. The parameter c_1 balances the exploration and exploitation phases. It is computed as follows:

$$c_1 = 2e^{-(\frac{4t}{t_{max}})^2}, \tag{14}$$

where t and t_{max} represent the current iteration and the total iterations' number, respectively. The followers' position is also updated by the following equation:

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \tag{15}$$

where, x_j^i denotes the i-th follower position in j-th dimension and $i > 1$.
Algorithm 2 show the entire steps of the SSA algorithm.

Algorithm 2 Salp Swarm Algorithm (SSA)

```
Generate a random population X.
repeat
    Calculate the objective function for solutions.
    Update the best salps (F = X^b).
    Update c_1 by Eq. (14).
    for i = 1 : N do
        if i == 1 then
            Update the salps' position by Eq. (13)
        else
            Update the salps' position by Eq. (15)
        end if
    end for
until (t < t_max)
Save the best solution F.
```

3 Proposed Method

This section introduces the proposed hybrid optimization algorithm, ASOSSA, developed to enhance parameter estimation in photovoltaic (PV) models. ASOSSA integrates Atom Search Optimization (ASO) with the Salp Swarm Algorithm (SSA), leveraging the strengths of both to improve performance in highly nonlinear and multimodal search spaces. While ASO provides efficient global exploration through mass-based attraction mechanisms, SSA contributes local exploitation using swarm-inspired positional adjustments.

The algorithm begins by initializing a random population of candidate solutions, each representing a potential parameter set for the PV model. ASO is first applied to evaluate fitness, update atom velocities, and reposition individuals in the search space based on acceleration and attraction forces derived from the best-performing solutions.

To prevent stagnation in local minima, the SSA mechanism is intermittently activated. In this phase, salp-like movements refine the position of solutions, with a leading member moving toward the global best and subsequent members following in a linked pattern. This swarm-based adaptation helps improve convergence and solution diversity.

ASOSSA alternates between ASO and SSA updates based on a probabilistic condition during each iteration. When ASO is active, population updates involve velocity-driven adjustments and mass-guided movement. In SSA phases, a subset of the population adapts using mid-point and chaotic-position strategies to fine-tune local exploration around promising regions. Throughout the process, the algorithm retains and updates the best global solution based on the RMSE between the model output and the experimental dataset.

The algorithm, as illustrated in Algorithm 3, is implemented in MATLAB, based on the original ASO [16] and SSA [17] frameworks, with enhancements for stability, boundary control, and hybrid switching. Performance is validated on both single diode (SD) and double diode (DD) solar cell models.

Algorithm 3 ASOSSA Algorithm

```
Initialize atoms with random positions and velocities
Evaluate fitness and set global best X_b
for t = 1 to t_max do
    Compute masses and accelerations
    if rand() > 0.5 then
        for each atom do
            Update velocity and position
            Apply bounds and re-evaluate fitness
        end for
    else
        Update leader using adaptive coefficient c_1
        for each follower do
            Update position using midpoint rule
        end for
        Evaluate fitness and update X_b if improved
    end if
end for
Return best solution X_b
```

4 Results and Discussion

This section presents the performance evaluation of the proposed ASOSSA algorithm for estimating the parameters of single-diode (SD) and double-diode (DD) photovoltaic models. The evaluation dataset is derived from a commercial monocrystalline silicon solar cell manufactured by R.T.C. Company (Paris, France), with a 57 mm diameter tested under standard conditions (1 Sun, 1000 W/m^2, and $T = 33°C$). The dataset comprises 23 widely cited I–V measurement records.

The experiments were implemented in MATLAB 2016b on a Windows 8 machine with an Intel Core i5 processor and 4 GB RAM. The ASOSSA algorithm was executed with a population size of 300 and a maximum of 3000 iterations. It is benchmarked against a wide set of algorithms: SSA, ASO, PSO, GA, ILCOA [26], LCOA [26], DE [27], Newton, HS [11], SCA [28], GGHS [29], BSA [30], and simulated annealing (SA) [31].

4.1 Performance Measures

The Root Mean Square Error (RMSE) is used as the main metric to evaluate the quality of parameter estimation:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(I_{tm}^{(i)} - I_{te}^{(i)})^2} \tag{16}$$

where I_{tm} is the measured current, I_{te} is the estimated current from the model, and $N = 37$ is the number of data points in the extended dataset.

4.2 Single Diode (SD) Model Results

Table 2 presents the predicted parameters and RMSE values obtained using ASOSSA and other comparative algorithms. ASOSSA achieved the lowest RMSE value of 9.860×10^{-4}, indicating its high accuracy. The estimated values for key parameters such as I_{ph}, n, R_s, and R_{sh} were consistent across the top-performing algorithms (ASOSSA, SSA, ASO), although ASOSSA produced a more precise estimate for I_{sd}.

Methods such as GGHS and ILCOA yielded competitive RMSE values; however, ILCOA introduced inconsistencies in I_{sd} that could affect the long-term reliability of the model. In contrast, Newton-based and other classical techniques resulted in significantly higher RMSEs (e.g., 9.7×10^{-3} for Newton, and 4.102×10^{-3} for GA), highlighting their limited effectiveness in nonlinear, multimodal optimization problems.

Algorithms such as PSO and SA showed RMSEs exceeding 10^{-2} and exhibited substantial deviations in parameters like R_{sh} and n, suggesting a tendency toward premature convergence or instability. These results emphasized the robustness of ASOSSA, which not only minimized estimation errors but also maintained physical plausibility across all parameters. Figure 1 shows the convergence of ASOSSA on SD model compared to ASO and SSA methods.

Table 2: Predicted values of SD circuit model and the corresponding RMSE for the proposed method and the compared algorithms

Parameter	ASOSSA	SSA	ASO	GGHS	ILCOA	LCOA	DE
I_{ph} (A)	0.7608	0.7606	0.7607	0.7609	0.7608	0.7608	0.7608
I_{sd} (A)	3.2E-07	2.9E-07	3.1E-07	3.3E-07	3.2E-01	3.2E-01	3.230E-7
n	1.4810	1.4705	1.4780	1.4820	1.4811	1.4812	1.4806
R_s (Ω)	0.0364	0.0368	0.0365	0.0363	0.0364	0.0364	0.0364
R_{sh} (Ω)	53.700	54.145	53.955	53.060	53.719	53.902	53.710
RMSE	**9.860E-04**	1.029E-03	9.900E-04	9.900E-04	**9.860E-04**	9.861E-04	2.340E-02

Parameter	Newton	HS	PSO	SCA	GA	BSA
I_{ph} (A)	0.7608	0.7607	0.7080	0.7670	0.7665	0.7610
I_{sd} (A)	3.223E-07	3.049E-07	2.1EE-07	2.3EE-07	7.45EE-07	4.8EE-07
n	1.4837	1.4753	1.5100	1.5000	1.5702	1.5200
R_s (Ω)	0.0364	0.0366	0.0363	0.0370	0.0314	0.0340
R_{sh} (Ω)	53.763	53.590	49.500	19.300	29.483	79.590
RMSE	9.700E-03	9.90E-04	0.082	5.40E-03	4.102E-3	1.44E-03

4.3 Double Diode (DD) Circuit Model Results

For the more complex DD model, the results shown in Table 3 further supported the effectiveness of ASOSSA. It again achieved the lowest RMSE of 9.826×10^{-4}, confirming its ability to generalize to models with additional complexity. The algorithm consistently estimated the values of I_{ph}, n_1, n_2, and both diode saturation currents more accurately than the other methods.

Although ILCOA and LCOA obtained similar RMSEs, their estimates for I_{sd1} and I_{sd2} were notably less accurate and exceeded expected physical ranges—e.g., 0.23 and 0.75—whereas ASOSSA provided more realistic and constrained results. This highlighted the better trade-off that ASOSSA achieved between convergence accuracy and parameter realism.

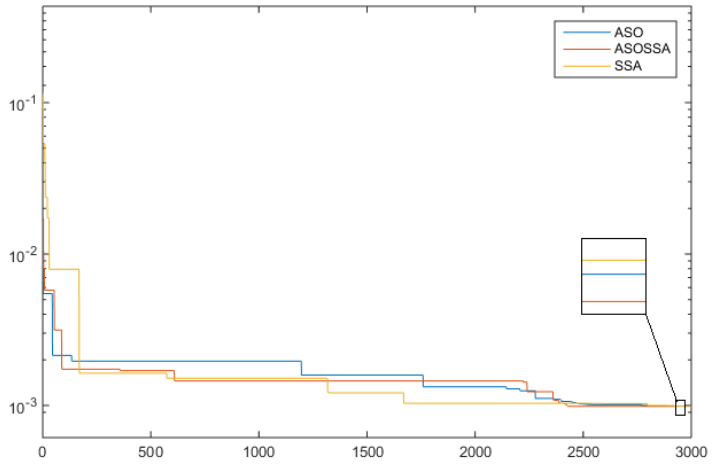

Figure 1: Convergence of ASOSSA on SD model

Other metaheuristic methods such as BSA and GGHS provided moderately acceptable results but lacked consistency across all estimated parameters. PSO and SCA, despite their popularity, failed to preserve physical validity in key values such as n_2 and R_{sh}, leading to higher RMSEs (e.g., 0.1985 for PSO), and thus, lower overall reliability. Figure 2 shows the convergence of ASOSSA on DD model compared to ASO and SSA methods.

Table 3: Predicted values of DD circuit model and the corresponding RMSE for the proposed method and the compared algorithms

Parameter	ASOSSA	ASO	SSA	ILCOA[2019]	LCOA[2019]	SA
I_{ph} (A)	0.76078	0.76085	0.76089	0.76078	0.76077	0.76230
I_{sd1} (A)	2.3E-07	2.1E-07	2.3E-07	2.3E-01	2.7E-01	4.8E-07
I_{sd2} (A)	6.8E-07	6.0E-07	6.0E-07	7.5E-01	3.8E-01	1.0E-08
n_1	1.45214	1.44630	1.45573	1.45101	1.46205	1.51700
n_2	1.98799	1.86681	1.91774	2.00000	1.99380	2.00000
R_s (Ω)	0.03673	0.03652	0.03641	0.03674	0.36670	0.03450
R_{sh} (Ω)	55.2881	55.8621	55.3055	55.5320	54.6314	43.1000
RMSE	**9.826E-04**	9.928E-04	9.981E-04	**9.826E-04**	9.842E-04	1.660E-02

Parameter	HS	GGHS	PSO	SCA	BSA
I_{ph} (A)	0.76170	0.76050	0.88148	0.76155	0.76700
I_{sd1}	1.3E-07	3.7E-07	4.5E-07	3.1E-07	4.2E-07
I_{sd2} (A)	2.6E-07	1.4E-07	7.8E-08	5.2E-08	1.0E-12
n_1	1.49400	1.49600	1.87886	1.83399	1.47000
n_2	1.49900	1.92900	1.34062	1.32096	2.00000
R_s (Ω)	0.03540	0.03560	0.04729	0.04432	0.03530
R_{sh} (Ω)	46.8200	62.7800	58.6496	58.2099	54.4500
RMSE	1.260E-03	0.00107	1.985E-01	0.0094	1.10E-02

In general, the proposed ASOSSA algorithm showed good reliability among the tested methods. It provided accurate results, maintained consistent performance across both SD and DD models, and yielded parameter estimates that adhered to known physical constraints.

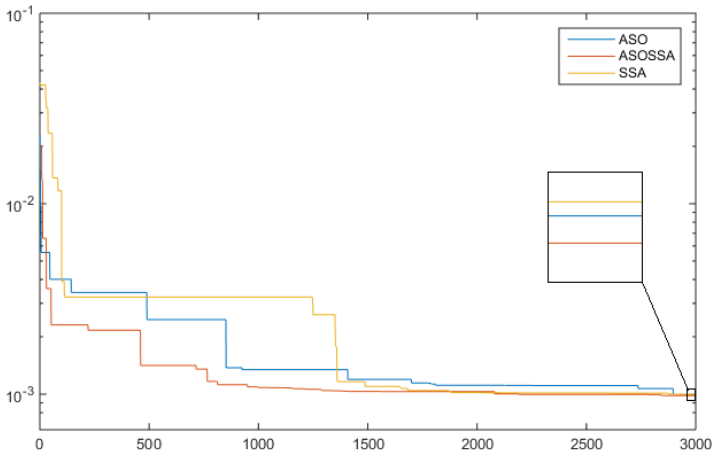

Figure 2: Convergence of ASOSSA on DD model

5 Conclusion

This study introduced ASOSSA, a hybrid metaheuristic optimization algorithm that integrates Atom Search Optimization (ASO) with the Salp Swarm Algorithm (SSA), for accurate parameter estimation in solar photovoltaic (PV) module modeling. Experimental results demonstrated that ASOSSA achieved the lowest RMSE of 9.860×10^{-4} for the single-diode (SD) model and 9.826×10^{-4} for the double-diode (DD) model. Compared to the nearest competing method (SSA), ASOSSA reduced RMSE by approximately 16.4% and 1.6% for the SD and DD models, respectively. The integration of ASO's global exploration with SSA's local exploitation enabled ASOSSA to effectively navigate the complex and multimodal search space of PV parameter estimation. The algorithm produced physically plausible and numerically stable solutions, and showed good reliability and convergence behavior across both tested models. Future research will explore the extension of ASOSSA to broader renewable energy optimization tasks, improve its convergence rate, and incorporate adaptive control mechanisms. These enhancements aim to support the development of intelligent, accurate, and efficient optimization tools for sustainable energy system design.

Conflicts of Interest

The authors declare that there is no conflict of interest.

References

[1] Mahmood Swadi, Dheyaa Jasim Kadhim, Mohamed Salem, Firas Mohammed Tuaimah, Ammar Sabri Majeed, and Ali Jawad Alrubaie. Investigating and predicting the role of photovoltaic, wind, and hydrogen energies in sustainable global energy evolution. *Global Energy Interconnection*, 7(4):429–445, 2024.

[2] Lina M Shaker, Ahmed A Al-Amiery, Mahdi M Hanoon, Waleed K Al-Azzawi, and Abdul Amir H Kadhum. Examining the influence of thermal effects on solar cells: a comprehensive review. *Sustainable Energy Research*, 11(1):6, 2024.

[3] Dalia Yousri, Mohamed Abd Elaziz, Diego Oliva, Laith Abualigah, Mohammed AA Al-Qaness, and Ahmed A Ewees. Reliable applied objective for identifying simple and detailed photovoltaic models using modern metaheuristics: Comparative study. *Energy Conversion and Management*, 223:113279, 2020.

[4] Zhentao Tang, Kaiyu Wang, Mingxin Zhu, Yongxuan Yao, Lan Zhuang, Huiqin Chen, Jing Li, Li Yan, and Shangce Gao. A deep-layered water flow optimizer for global continuous optimization problems and parameter estimation of solar photovoltaic models. *IEEE Access*, 2025.

[5] Diego Oliva, Ahmed A Ewees, Mohamed Abd El Aziz, Aboul Ella Hassanien, and Marco Peréz Cisneros. A chaotic improved artificial bee colony for parameter estimation of photovoltaic cells. *Energies*, 10(7):865, 2017.

[6] Óscar Dávalos-Orozco, Pedro M Rodrigo, Fernando Brambila-Paz, and Julio C Rosas-Caro. An accurate explicit six-parameter solar cell model based on single-diode and its parameter extraction for seven photovoltaic technologies. *Journal of Solar Energy Engineering*, 146(2), 2024.

[7] Jordan Nafack Nihako, Elie Simo, Derrick Duclos Abada Essouma, Maëlle Nanmegne Leutchouang, Christel Roseny Atteutsia Tsakem, Christelle Yolande Tchienou Tchienou, Jimy Synclair Kenhago Watia, Pierre-Olivier Logerais, and Joseph Marae Djouda. Realistic modeling of photovoltaic solar cell: A simple and accurate two-diode model. *Applied Research*, 4(2):e70010, 2025.

[8] Hanaa Fathi, Deema Mohammed Alsekait, Arar Al Tawil, Israa Wahbi Kamal, Mohammad Sameer Aloun, and Ibrahim IM Manhrawy. Enhancing sustainability in renewable energy: Comparative analysis of optimization algorithms for accurate pv parameter estimation. *Sustainability*, 17(6):2718, 2025.

[9] Jun Qian, Hui Zhang, and Shun Wang. Parameter identification of the pv systems based on an adapted version of human evolutionary optimizer. *Scientific Reports*, 15(1):7375, 2025.

[10] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, pages 1942–1948. IEEE, 1995.

[11] Zong Woo Geem, Joong Hoon Kim, and Gobichettipalayam Vasudevan Loganathan. A new heuristic optimization algorithm: harmony search. *simulation*, 76(2):60–68, 2001.

[12] Jonathan Shapiro. Genetic algorithms in machine learning. In *Advanced course on artificial intelligence*, pages 146–168. Springer, 1999.

[13] Rehab Ibrahim, Marwa Gaheen, and Hashim Jarar. Boosted pathfinder algorithm and firefly algorithm for multi-level thresholding image segmentation using fuzzy entropy. *Artificial Intelligence Topics and Applications*, 1(1):1–21, 2025.

[14] Aykut Fatih Güven, Salah Kamel, and Mohamed H Hassan. Optimization of grid-connected photovoltaic/wind/battery/supercapacitor systems using a hybrid artificial gorilla troops optimizer with a quadratic interpolation algorithm. *Neural Computing and Applications*, 37(4):2497–2535, 2025.

[15] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[16] Weiguo Zhao, Liying Wang, and Zhenxing Zhang. A novel atom search optimization for dispersion coefficient estimation in groundwater. *Future Generation Computer Systems*, 91:601–610, 2019.

[17] Seyedali Mirjalili, Amir H Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, and Seyed Mohammad Mirjalili. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 2017.

[18] Imade Choulli, Mustapha Elyaqouti, Driss Saadaoui, Souad Lidaighbi, Abdelfattah Elhammoudy, Ismail Abazine, et al. Diwjaya: Jaya driven by individual weights for enhanced photovoltaic model parameter estimation. *Energy Conversion and Management*, 305:118258, 2024.

[19] Nandhini Kullampalayam Murugaiyan, Kumar Chandrasekaran, Premkumar Manoharan, and Bizuwork Derebew. Leveraging opposition-based learning for solar photovoltaic model parameter estimation with exponential distribution optimization algorithm. *Scientific Reports*, 14(1):528, 2024.

[20] Morad Ali Kh Almansuri, Ziyodulla Yusupov, Javad Rahebi, and Raheleh Ghadami. Parameter estimation of pv solar cells and modules using deep learning-based white shark optimizer algorithm. *Symmetry*, 17(4):533, 2025.

[21] Zhentao Tang, Kaiyu Wang, Yongxuan Yao, Mingxin Zhu, Lan Zhuang, Huiqin Chen, Jing Li, Li Yan, and Shangce Gao. Parameter adaptive manta ray foraging optimization for global continuous optimization problems and parameter estimation of solar photovoltaic models. *International Journal of Computational Intelligence Systems*, 18(1):29, 2025.

[22] Diego Oliva, Mohamed Abd Elaziz, Ammar H Elsheikh, and Ahmed A Ewees. A review on meta-heuristics methods for estimating parameters of solar cells. *Journal of Power Sources*, 435:126683, 2019.

[23] A. Laudani, F. Riganti Fulginei, and A. Salvini. Identification of the one-diode model for photovoltaic modules from datasheet values. *Solar Energy*, 108:432–446, 2014.

[24] Wei Zhang, Xiding Zeng, Yuan Huang, Juan Liang, Xinyu Wang, Jiahong Guo, Zhangyu Li, Kun Yang, and Jing Zhang. Based on machine learning: Energy consumption optimization and energy efficiency evaluation for photovoltaic electro-fenton technology in wastewater treatment plant. *Renewable Energy*, page 122626, 2025.

[25] Rehab Ali Ibrahim, Ahmed A Ewees, Diego Oliva, Mohamed Abd Elaziz, and Songfeng Lu. Improved salp swarm algorithm based on particle swarm optimization for feature selection. *Journal of Ambient Intelligence and Humanized Computing*, 10(8):3155–3169, 2019.

[26] Nafiseh Pourmousa, S Mohammadreza Ebrahimi, Milad Malekzadeh, and Mojtaba Alizadeh. Parameter estimation of photovoltaic cells using improved lozi map based chaotic optimization algorithm. *Solar Energy*, 180:180–191, 2019.

[27] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[28] Seyedali Mirjalili. Sca: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96:120–133, 2016.

[29] Behrooz Keshtegar and Mahmoud Oukati Sadeq. Gaussian global-best harmony search algorithm for optimization problems. *Soft Computing*, 21:7337–7349, 2017.

[30] Xian-Bing Meng, Xiao Zhi Gao, Lihua Lu, Yu Liu, and Hengzhen Zhang. A new bio-inspired optimisation algorithm: Bird swarm algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*, 28(4):673–687, 2016.

[31] Khaled M El-Naggar, MR AlRashidi, MF AlHajri, and AK Al-Othman. Simulated annealing algorithm for photovoltaic parameters identification. *Solar Energy*, 86(1):266–274, 2012.