Mathematical Biosciences and Engineering

MBE, 19(6): 5638–5670.
DOI: 10.3934/mbe.2022264
Received: 12 February 2022
Revised: 17 March 2022
Accepted: 18 March 2022
Published: 31 March 2022

http://www.aimspress.com/journal/MBE

Research article

A chaotic self-adaptive JAYA algorithm for parameter extraction of photovoltaic models

Juan Zhao[1,*], Yujun Zhang[1], Shuijia Li[2], Yufei Wang[1], Yuxin Yan[3] and Zhengming Gao[4,5]

[1] School of electronics and information engineering, Jingchu University of Technology, Jingmen 448000, China
[2] School of Computer Science, China University of Geosciences, Wuhan 430074, China
[3] Academy of arts, Jingchu University of Technology, Jingmen 448000, China
[4] School of computer engineering, Jingchu University of Technology, Jingmen 448000, China
[5] Institute of intelligent information technology, Hubei Jingmen industrial technology research institute, Jingmen 448000, China

* **Correspondence:** Email: ajuan323@jcut.edu.cn.

Abstract: In order to have the highest efficiency in real-life photovoltaic power generation systems, how to model, optimize and control photovoltaic systems has become a challenge. The photovoltaic power generation systems are dominated by photovoltaic models, and its performance depends on its unknown parameters. However, the modeling equation of the photovoltaic model is nonlinear, leading to the difficulty in parameter extraction. To extract the parameters of the photovoltaic model more accurately and efficiently, a chaotic self-adaptive JAYA algorithm, called AHJAYA, was proposed, where various improvement strategies are introduced. First, self-adaptive coefficients are introduced to change the priority of information from the best search agent and the worst search agent. Second, by combining the linear population reduction strategy with the chaotic opposition-based learning strategy, the convergence speed of the algorithm is improved as well as avoid falling into local optimum. To verify the performance of the AHJAYA, four photovoltaic models are selected. The experimental results prove that the proposed AHJAYA has superior performance and strong competitiveness.

Keywords: photovoltaic model; parameter extraction; AHJAYA; self-adaptive; Linear population reduction; chaotic opposition-based learning

1. Introduction

As resources continue to be depleted, finding high-quality renewable energy is a very important task. Among the many renewable energy sources, photovoltaic energy [1] is called the most potential renewable energy. It has a series of advantages that traditional energy cannot compare with, such as clean, pollution-free, renewable, and daily usable. However, the performance of a photovoltaic system depends on the chosen photovoltaic model and unknown parameters [2] in the model. At present, a variety of photovoltaic models have been developed, including single-diode model [3] (SDM), double-diode model [4] (DDM), three-diode model [5] (TDM), etc., but the most widely and most commonly [6] used are still the SDM and DDM. However, various uncertain factors will directly affect the changes of parameters, thereby reducing the performance of photovoltaic systems. Therefore, it is necessary to extract unknown parameters from the photovoltaic model before the photovoltaic system is used.

To accurately extract these unknown parameters, many methods have been proposed. These methods can be roughly divided into three categories, including analytical methods, deterministic methods, and meta-heuristic methods. Analytical methods and deterministic methods rely on the initial values of the model and the necessary assumptions, so it is easy to lead to a decrease in the accuracy of the solution, and even the latter is easy to fall into the local optimum. Therefore, in order to address these difficulties, many meta-heuristic methods are used for parameter extraction of photovoltaic models, due to its simple structure, clear concept, few parameters and high efficiency. For example, Whale Optimization Algorithm [7] (WOA), Differential Evolution Algorithm [8] (DE), Harmony Search Algorithm [9] (HS), Cuckoo Search Algorithm [10] (CS), Genetic Algorithm [11] (GA), Artificial Bee Colony Algorithm [12] (ABC), Simulated Annealing Algorithm [13] (SA), Teaching-Learning-Based Optimization Algorithm [14] (TLBO), Ant Lion Optimizer [15] (ALO), Arithmetic Optimization Algorithm [16] (AOA), Bonobo Optimizer [17] (BO), Sine Cosine Algorithm [18] (SCA), Rao-1 Algorithm [19], Empire Competition Algorithm [20] (ICA), Marine predators algorithm [21] (MPA), Harris Hawk optimization algorithm [22] (HHO), etc. Yu et al. [23] proposed a new variant of differential evolution (PDcDE) to extract the parameters of several solar photovoltaic models. Gao et al. [24] proposed a directed permutation differential evolution algorithm (DPDE) to solve the parameter estimation problem of several solar photovoltaic models. Although these algorithms have obtained satisfactory results in the extraction of photovoltaic parameters, in order to further reduce the complexity of the algorithm and improve the efficiency of the algorithm, it is still necessary to find better algorithm for further reducing the complexity of the algorithm and improving the efficiency of the algorithm.

The JAYA algorithm [25] is a very potential swarm-based optimization algorithm. Compared with the traditional swarm optimization algorithm, the JAYA algorithm has two advantages. First, the JAYA algorithm has no additional control parameters, only a common parameter, the initial population size. This means that the algorithm runs faster. Second, the algorithm has only one evolutionary strategy. This shows that the structure of the algorithm becomes very simple, on the other hand, the resources used for computation are reduced. Because of these advantages, many improved JAYA algorithms are used to solve various high-dimensional complex problems [26–30]. Tefek et al. [31] proposed Jaya Linear (Jaya-L) and Jaya Quadratic (Jaya-Q) models for estimating the future number of road accidents in Turkey. Gholami et al. [32] proposed a Powerful Enhanced Jaya (PEJAYA) to solve numerical and engineering problems, and the experimental results show that the improvement is very effective. Jian et al. [33] proposed a chaotic second order oscillation JAYA algorithm

(CSOOJAYA) for parameter extraction of photovoltaic models. The experimental results show that CSOOJAYA performs well in all aspects. Belagoune et al. [34] proposed a discrete chaotic Jaya optimization (DCJO) algorithm to perform preventive maintenance scheduling of power system generators. The experimental results show that the proposed DCJO is more effective than other optimization algorithms to solve the GMS problem.

In this paper, a chaotic self-adaptive JAYA algorithm, called AHJAYA, is proposed. In the AHJAYA algorithm, various improvement strategies are introduced. A self-adaptive coefficient strategy is introduced with the aim of changing the priority of utilizing the best search agent and the worst search agent in the update formula. This means that the overall evolution is biased towards the optimal search agent. The exploration ability of the algorithm is improved. On the other hand, the linear population reduction strategy and the chaotic opposition-based learning strategy are introduced by the proposed AHJAYA, the convergence speed of the algorithm is further improved and the local optimum can be avoided. It should be noted that there are already many efficient self-adaptive coefficient strategies [35,36], and corresponding adaptive strategies need to be selected for different problems. On the other hand, linear population reduction strategy is also often used to improve the performance of the algorithm [37,38]. To verify the performance of the AHJAYA, several different photovoltaic models are chosen, including the single-diode model, the double-diode model, the STM6-40/36 model, and the STP6-120/36 model. Finally, the AHJAYA is compared with other mature algorithms. The experimental results show that the proposed AHJAYA has superior performance and is in a leading position among the algorithms used for photovoltaic parameter extraction.

The main contributions of this paper are as follows:

1) In order to extract the parameters of the photovoltaic model more accurately and efficiently, an improved AHJAYA is proposed.

2) The self-adaptive coefficient strategy is introduced to change the priority of the optimal search agent and the worst search agent in the evolution strategy.

3) The chaotic opposition-based learning strategy is proposed to prevent the AHJAYA from falling into local optimum.

4) The linear population reduction strategy is introduced so that the algorithm converges faster and uses less computational resources.

The rest of this paper is arranged as follows: In Section 2, the definition of photovoltaic (PV) model and objective function is introduced. In Section 3, the original JAYA algorithm is introduced. In Section 4, the improved AHJAYA algorithm is introduced in detail. In Section 5, simulation experiments and result analysis are carried out by the AHJAYA and the comparison algorithms. In Section 6, summarized the article and looked forward to future work.

2. Definition of photovoltaic (PV) model

In this section, three photovoltaic models (single-diode model, double-diode model, PV module model) are introduced and their objective functions are defined.

2.1. Single-diode model

The characteristics of solar cells can be accurately described by SDM, which can be expressed by the following formula.

$$I_L = I_{pv} - I_d - I_p \tag{1}$$

$$I_d = I_{sd}\left[\exp(\frac{(V_L + I_L R_s) \times q}{nkT}) - 1\right] \tag{2}$$

$$I_p = \frac{V_L + I_L R_s}{R_P} \tag{3}$$

where, I_p is the shunt resistor current, I_d is the diode current, I_{pv} is the current generated by solar irradiation, I_{sd} is the diode saturation current, V_L is the output voltage, R_s and R_P are the series and shunt resistances respectively, n is the diode characteristic factor, $k = 1.3806503 \times 10^{-23} J/K$ and $q = 1.60217646 \times 10^{-19} C$ are both constants.

Therefore, the output current of the SDM can be expressed by the following formula.

$$I_L = I_{pv} - I_{sd}\left[\exp(\frac{(V_L + I_L R_s) \times q}{nkT}) - 1\right] - \frac{V_L + I_L R_s}{R_P} \tag{4}$$

It can be seen from the formula that this model needs to extract five unknown parameters including I_{pv}, I_{sd}, R_s, R_P and n.

2.2. Double-diode model

DDM adds a diode on the basis of SDM, so the effect of loss of recombination current is considered. This model can be expressed by the following formula.

$$I_L = I_{pv} - I_{d1} - I_{d2} - I_p \tag{5}$$

$$I_{d_1} = I_{sd_1}\left[\exp(\frac{(V_L + I_L R_s) \times q}{n_1 kT}) - 1\right] \tag{6}$$

$$I_{d_2} = I_{sd_2}\left[\exp(\frac{(V_L + I_L R_s) \times q}{n_2 kT}) - 1\right] \tag{7}$$

where, I_{sd_1} and I_{sd_2} are the diffusion current and saturation current, and n_1 and n_2 are the ideality factors of the two diodes, respectively.

Therefore, the output current of the DDM can be expressed by the following formula.

$$I_L = I_{pv} - I_{sd_1}\left[\exp(\frac{(V_L + I_L R_s) \times q}{n_1 kT}) - 1\right] - I_{sd_2}\left[\exp(\frac{(V_L + I_L R_s) \times q}{n_2 kT}) - 1\right] - \frac{V_L + I_L R_s}{R_P} \tag{8}$$

It can be seen from the formula that this model needs to extract seven unknown parameters including I_{pv}, I_{sd_1}, I_{sd_2}, R_s, R_P, n_1 and n_2.

2.3. PV module model

The PV module model is built on multiple PV cells connected in parallel and in series. Therefore, it can be expressed by the following formula.

$$I_L = I_{pv}N_p - I_{sd}N_p\left[\exp(\frac{(V_LN_p + I_LR_sN_s)\times q}{nN_sN_pkT}) - 1\right] - \frac{V_LN_p + I_LR_sN_s}{R_PN_s} \tag{9}$$

where, N_s represents the number of photovoltaic cells in series, and N_p represents the number of photovoltaic cells in parallel.

It can be seen from the formula that this model needs to extract five unknown parameters.

2.4. Objective function of PV model

The parameters of the above models are estimated when using data provided by the supplier. Usually, an objective function is needed to estimate the error of the experiment. In this paper, the root mean square error (RMSE) is adopted as the objective function for optimization. Because it can reflect the degree of error between the measured data and the real data.

$$min \qquad RMSE(x) = \sqrt{\frac{\sum_{i=1}^{N}(I_i - I_L)^2}{N}} \tag{10}$$

where, N is the number of datasets, I_L is the calculated current, and I_i is the data provided by the supplier.

It can be seen from formula (10) that when the value of RMSE is smaller, the extracted parameters are more accurate.

3. JAYA algorithm

The JAYA algorithm is based on the idea of approaching the best solution and moving away from the worst solution in the process of computation. Different from the traditional differential evolution (DE) algorithm, the JAYA algorithm has only one common parameter, which is the population size. In addition, there is only one evolution strategy in the algorithm. All individuals evolve through this strategy, which can be expressed by the following formula.

$$Y_{i,j} = X_{i,j} + rand_1 \times \left(X_{best,j} - \left|X_{i,j}\right|\right) - rand_2 \times \left(X_{worst,j} - \left|X_{I,j}\right|\right) \tag{11}$$

where, $X_{best,j}$ represents the best solution, $X_{worst,j}$ represents the worst solution, and $rand_1$ and $rand_2$ are random numbers between 0 and 1.

If the fitness value of the updated solution is better than the previous solution, then the updated solution can be accepted, otherwise, the previous solution is kept.

$$X_i = \begin{cases} Y_i, & if \ \ f(Y_i) < (X_i) \\ X_i, & otherwise \end{cases} \tag{12}$$

4. The proposed AHJAYA algorithm

For the JAYA algorithm, its main no-parameter feature is the most attractive. The best and worst individuals in the population are used by the JAYA algorithm to exploration in the global scope. However, the single exploration strategy can easily lead to incomplete exploration or even local optimum. To improve these shortcomings, some improvement strategies are introduced.

4.1. Self-adaptive coefficient strategy

In the process of exploration, the best and worst individuals with the same priority are selected by the original JAYA algorithm, but this approach cannot have efficient exploration ability. Therefore, the priority of individual assignment needs to be changed, so that all search agents are searched in the direction of the optimal individual, and self-adaptive coefficient strategy [39] is introduced.

$$A_1 = \begin{cases} \dfrac{mean(f(X))}{f(X_{best})}, f(X_{best}) \neq 0 \\ 1, f(X_{best}) = 0 \end{cases} \tag{13}$$

$$A_2 = \begin{cases} \dfrac{mean(f(X))}{f(X_{worst})}, f(X_{worst}) \neq 0 \\ 1, f(X_{worst}) = 0 \end{cases} \tag{14}$$

where, X_{best} and X_{worst} are the global best solution and the worst solution, respectively. In the process of optimization, the value of A_1 is greater than 1, and the value of A_2 is less than 1. As the iteration continues to increase, they eventually approach 1.

These two coefficients are introduced into the update formula. It can be expressed by the following formula.

$$Y_{i,j} = X_{i,j} + A_1 \times rand_1 \times \left(X_{best,j} - \left|X_{i,j}\right|\right) - A_2 \times rand_2 \times \left(X_{worst,j} - \left|X_{I,j}\right|\right) \tag{15}$$

At the beginning of the iteration, the difference between the two coefficients is large, so all search agents are moved to the position of the global optimum, which improves the performance of the exploration. In late iterations, two coefficients approach 1 and local exploration is implemented, improving exploitation performance.

4.2. Linear population reduction strategy

The performance of the JAYA algorithm is directly affected by the population size. In order to improve the optimization efficiency of the algorithm, a linear population reduction strategy is adopted, therefore, the convergence speed is further improved. It can be expressed by the following formula.

$$NP_{G+1} = round\left[\left(\frac{NP_{min} - NP_{init}}{MaxNFES}\right) \times NFES + NP_{init}\right] \tag{16}$$

where NP_{min} is the population size at the end of the algorithm iteration, which is set to 3, NP_{init} is the initial population size, $NFES$ is the current number of evaluations, $MaxNFES$ is the maximum number of evaluations, NP_G is the population size of the current generation, and NP_{G+1} is the population size of the next generation. On the other hand, as the population continues to decrease, the algorithm is gradually biased towards exploitation, so there is good transition between exploration and exploitation.

4.3. The chaotic opposition-based learning strategy

In the original JAYA algorithm, the case for local optimum cannot be handled. Therefore, a chaotic map is introduced, which is used to jump out of local optimum in the algorithm. The specific discussion of the chaotic map is placed in Section 5.2, and this chaotic map is tentatively used here. It can be expressed by the following formula.

$$Z_{i,j} = \left(\left(UB_j + LB_j\right) - X_{i,j}\right) \times |P_k| \quad (17)$$

$$P_{k+1} = \begin{cases} \vartheta \times (1 - \tau_1 \times P_k^2) & P_k < 0 \\ 1 - \tau_2 \times P_k & otherwise \end{cases} \quad (18)$$

where, $\vartheta = 0.85$, $\tau_1 = 1.8$, $\tau_2 = 2.0$, UB_j and LB_j represent the maximum and minimum values in the current population in the *j*th dimension, respectively. The initial value of P_k is 0.7.

It is worth noting that this chaotic mapping fluctuates between -1 and 1, so the absolute function must be added before it can be used, as shown in Figure 1.

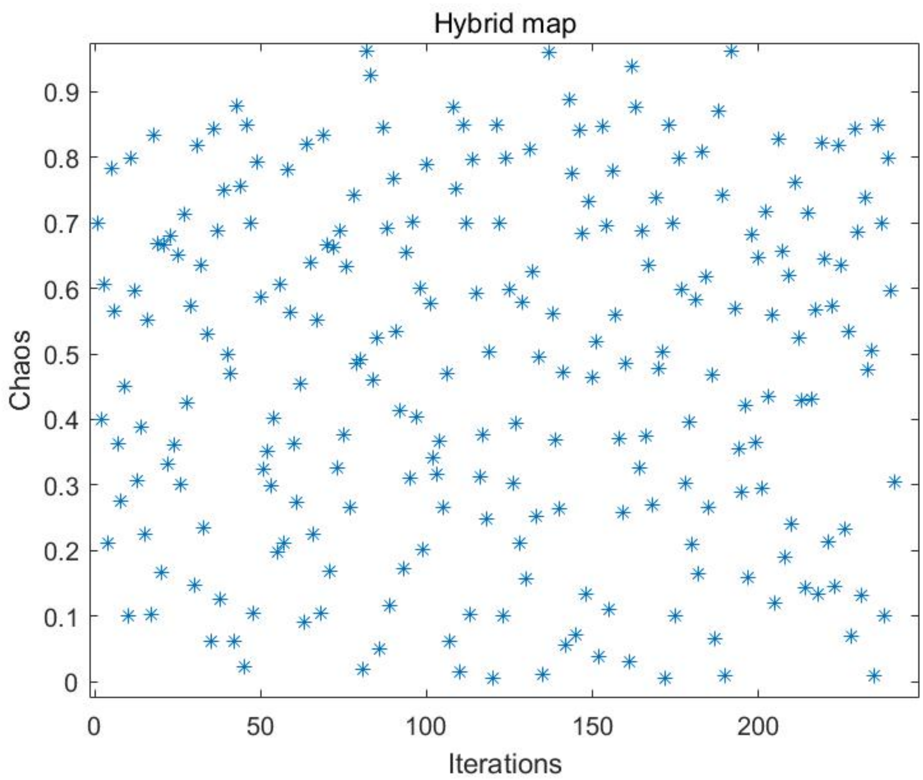


Figure 1. The chaotic mapping fluctuations.

4.4. Overview of the proposed AHJAYA algorithm

After the original JAYA algorithm is introduced by the above three improvement strategies, both the exploration and exploitation capabilities of the algorithm have been greatly improved. In the exploration phase, the exploration capability is enhanced by two coefficients, because the optimal search agent is assigned a higher priority by the two coefficients, and therefore, the overall evolution towards the optimal search agent. In the exploitation phase, the population gradually decreases, on the other hand, the two coefficients approach 1, so the exploitation capacity is greatly improved. Finally, the chaotic opposition-based learning strategy is introduced, which avoids the algorithm from falling into local optimum. The overall algorithm structure has not changed and still has a simple structure.

In the proposed AHJAYA, since the evolutionary strategy is not changed, the increased complexity comes from sorting after removing individuals from the population and the chaotic opposition-based learning strategy. The complexity of sorting is $O(NP \times log(NP)$, and the complexity of chaotic opposition-based learning is $O(NP \times Dim)$. Therefore, the complexity of the proposed JAYA algorithm is $O(G_{max} \times NP \times (log(NP) + Dim)$. Where, G_{max} is the maximum number of iterations, and Dim is the population dimension. The pseudo code of the proposed AHJAYA is shown in Algorithm 1. It can be seen from Algorithm 1 that two coefficients are updated before each iteration, and linear population reduction strategy will be used after the population is updated through the evolution strategy. In addition, for the chaotic opposition-based learning strategy, when the random number is less than 0.3, it means that the new position needs to be updated through the strategy at this time.

Algorithm 1: The pseudo-code of the proposed AHJAYA

```
Set population size NP, the maximum number of evaluations MaxNFES, dimension Dim
Initialize the positions of Individuals X_i (i = 1,2, ..., NP)
Set NFES = 0, NP = NP_max = 50, NP_min = 3.
Set NFES = NP.
While (NFES ≤ MaxNFES)
    Calculate coefficients A_1 and A_2 using Eqs (13) and (14)
    For i = 1 : NP
        Update the new position X' using Eq (15)
        If f(X') < f(X) then
          X=X'
        End if
    End For
    Calculate the new population NP_{G+1} using Eq (16)
    If rand < Q then
        Calculate the new position X_COBL using Eq (17).
    End If
    Memory saving
End While
Return X_best
```

It should be noted that the threshold setting of Q also affects the overall performance of the algorithm, so the specific discussion is set in Section 5.1.

5. Experiments and results

5.1. Analysis of the threshold value of parameter Q

The threshold of parameter Q affects the overall performance of the algorithm, so it is necessary to analyze this threshold. It is worth noting that the algorithm as a whole is only affected by this factor, so it is only necessary to set different Q values, and then statistically analyze the relevant values to draw conclusions. For the convenience of the experiment, the CEC2020 competition is selected, because this competition has a large number of complex functions to test the performance of the algorithm, so slight changes in the performance of the algorithm can be reflected numerically. In order to facilitate the experiment, the values of Q are 0.1, 0.3, 0.5, 0.7 and 0.9, respectively. The experimental results are shown in Table 1. All experiments were run 30 times, and the maximum number of evaluations is set to 15,000 times.

It can be seen from Table 1 that when Q = 3, AHJAYA performs the best, accounting for 9 of the 20 best values. Therefore, the threshold of the most suitable Q is set to 0.3.

Table 1. Results of the CEC2020 competition with different values of Q.

F	Item	AHJAYA (Q = 0.1)	AHJAYA (Q = 0.3)	AHJAYA (Q = 0.5)	AHJAYA (Q = 0.7)	AHJAYA (Q = 0.9)
CEC2020F1	Mean	**7.163×10^6**	7.482×10^7	2.775×10^8	2.048×10^8	7.859×10^8
	Std	**1.019×10^7**	1.016×10^8	4.013×10^8	2.357×10^8	7.927×10^8
CEC2020F2	Mean	1.982×10^3	**1.830×10^3**	1.913×10^3	1.884×10^3	2.020×10^3
	Std	**2.790×10^2**	2.867×10^2	3.297×10^2	3.036×10^2	3.783×10^2
CEC2020F3	Mean	7.396×10^2	**7.391×10^2**	7.433×10^2	7.430×10^2	7.457×10^2
	Std	9.554	9.474	10.53	10.02	**9.154**
CEC2020F4	Mean	1.903×10^3	**1.901×10^3**	2.405×10^3	3.067×10^3	4.221×10^3
	Std	1.764	**16.16**	1.305×10^3	3.327×10^3	7.701×10^3
CEC2020F5	Mean	1.091×10^4	**6.713×10^3**	7.471×10^3	1.818×10^4	2.847×10^4
	Std	6.329×10^3	4.880×10^3	**3.707×10^3**	3.799×10^4	6.844×10^4
CEC2020F6	Mean	1.602×10^3	**1.601×10^3**	1.602×10^3	**1.601×10^3**	1.603×10^3
	Std	3.301	**0.1688**	3.471	0.2548	4.793
CEC2020F7	Mean	4.017×10^3	**3.288×10^3**	4.196×10^3	4.598×10^3	4.783×10^3
	Std	1.129×10^3	**7.755×10^2**	1.420×10^3	1.787×10^3	1.876×10^3
CEC2020F8	Mean	**2.304×10^3**	2.310×10^3	2.317×10^3	2.325×10^3	2.342×10^3
	Std	**17.73**	21.07	29.60	28.66	50.97
CEC2020F9	Mean	2.727×10^3	2.710×10^3	**2.674×10^3**	2.722×10^3	2.711×10^3
	Std	76.83	89.78	1.101×10^2	**75.50**	85.39
CEC2020F10	Mean	**2.924×10^3**	2.938×10^3	2.935×10^3	2.950×10^3	2.941×10^3
	Std	**20.93**	23.32	25.92	30.74	21.81

5.2. Analysis of chaotic map

Many chaotic maps have been proven to be effective in improving the performance of algorithms. Therefore, comparative analysis is required when choosing a chaotic map. Three effective chaotic maps are selected in this paper, including Hybrid map, Piecewise linear map [40] and Chebyshev map [41]. The formula for piecewise linear map can be expressed as follows.

$$x(n+1) = \begin{cases} \frac{x_n}{1-\lambda} & 0 < x_n < 1-\lambda \\ \frac{x_n-(1-\lambda)}{\lambda} & 1-\lambda < x_n < 1 \end{cases} \tag{19}$$

where, the value of λ is 0.6. The formula for Chebyshev map can be expressed as follows.

$$x(n+1) = cos(\frac{i}{cos(X_n)}) \tag{20}$$

where, i is the population number. It should be noted that the initial value of both chaotic maps is 0.7.

The algorithms corresponding to the piecewise linear map and the Chebyshev map are named AHJAYA-P and AHJAYA-C, respectively. Similarly, the three algorithms are still experimented in the CEC2020 competition, the experiment is carried out 30 times, and the maximum number of evaluations is 15,000 times. The experimental results are shown in Table 2.

From the experimental results in the table, the AHJAYA algorithm with Hybrid map performs the best, accounting for 10 of the 20 best values. It should be noted that AHJAYA-P also showed strong competitiveness. Therefore, Hybrid map is chosen in this paper.

Table 2. Results of different chaotic maps at the CEC2020 competition.

F	Item	AHJAYA	AHJAYA-P	AHJAYA-C
CEC2020F1	Mean	7.482×10^7	$\mathbf{9.973 \times 10^5}$	8.764×10^7
	Std	1.016×10^8	$\mathbf{2.484 \times 10^6}$	1.349×10^8
CEC2020F2	Mean	$\mathbf{1.830 \times 10^3}$	1.899×10^3	1.894×10^3
	Std	2.867×10^2	3.352×10^2	$\mathbf{2.600 \times 10^2}$
CEC2020F3	Mean	7.391×10^2	$\mathbf{7.358 \times 10^2}$	7.433×10^2
	Std	9.474	**8.745**	14.17
CEC2020F4	Mean	$\mathbf{1.901 \times 10^3}$	1.903×10^3	1.912×10^3
	Std	16.16	**0.9553**	31.71
CEC2020F5	Mean	$\mathbf{6.713 \times 10^3}$	1.023×10^4	1.193×10^4
	Std	$\mathbf{4.880 \times 10^3}$	8.561×10^3	1.691×10^4
CEC2020F6	Mean	$\mathbf{1.601 \times 10^3}$	$\mathbf{1.601 \times 10^3}$	$\mathbf{1.601 \times 10^3}$
	Std	**0.1688**	0.2774	0.2487
CEC2020F7	Mean	$\mathbf{3.288 \times 10^3}$	3.338×10^3	3.295×10^3
	Std	$\mathbf{7.755 \times 10^2}$	9.050×10^2	9.513×10^2
CEC2020F8	Mean	$\mathbf{2.310 \times 10^2}$	2.318×10^3	2.312×10^3
	Std	21.07	**6.422**	34.50
CEC2020F9	Mean	$\mathbf{2.710 \times 10^3}$	2.728×10^3	2.712×10^3
	Std	89.78	**65.79**	90.08
CEC2020F10	Mean	2.938×10^3	2.934×10^3	$\mathbf{2.932 \times 10^3}$
	Std	23.32	**20.03**	23.08

5.3. Parameter settings

In order to verify the performance of the proposed AHJAYA, JAYA, PGJAYA [42] and EJAYA [27] are selected as comparison algorithms. The setting of specific parameters is shown in Table 3. The maximum number of evaluations is set to 15,000 for single diodes, double diodes, STM6-40/36 and STP6-120/36. All experiments are run 30 times. All of the simulation experiments would be carried out with HP DL380 Gen 10 server with 32GB RAM and Intel Xeon Bronze 3106 × 2 cores, and MATLAB 2017b software.

Table 3. Parameter settings of algorithms.

Algorithm	Parameter
JAYA	$NP = 50$
PGJAYA	$NP = 50$
EJAYA	$NP = 50$
AHJAYA	$NP = 50$, $Q = 0.3$

5.4. PV model selection and parameter setting

Three different PV models were chosen to test the performance of the AHJAYA on four PV datasets. For the single-diode and double-diode model, R.T.C. France solar cell of the 57 mm diameter commercial is selected. For the PV module model, monocrystalline STM6-40/36 and polycrystalline STP6-120/36 is selected. The setting of specific relevant parameters is shown in Tables 4 and 5.

Table 4. Correlation data of three PV models.

Parameter	The single-diode/ double-diode model	STM6-40/36	STP6-120/36
NP	1	1	1
NS	1	36	36
Data Volume	26	20	24
temperature	25 ℃	51 ℃	55 ℃
Radiance	1000 W/m²	1000 W/m²	1000 W/m²

Table 5. Parameter settings of three PV models.

Parameter	R.T.C. France solar cell		STM6-40/36		STP6-120/36	
	LB	UB	LB	UB	LB	UB
$\boldsymbol{I_{pv}}$(A)	0	1	0	2	0	8
$\boldsymbol{I_{sd_1}}$, $\boldsymbol{I_{sd_2}}$, $\boldsymbol{I_{sd}}$($\boldsymbol{\mu}$A)	0	1	0	50	0	50
$\boldsymbol{R_P}$(Ω)	0	100	0	1000	0	1500
$\boldsymbol{R_S}$(Ω)	0	0.5	0	0.36	0	0.36
$\boldsymbol{n_1}$, $\boldsymbol{n_2}$, $\boldsymbol{n}$	1	2	1	60	1	50

5.5. Experimental results of the single-diode model

For the single-diode model, the RMSE results obtained by the four algorithms are shown in Table 6, including the best value, the worst value, the mean, and the standard deviation. It can be seen from the results that the three algorithms (including AHJAYA, PGJAYA, JAYA) can obtain the best RMSE value, but only the AHJAYA has the best performance in the worst value, average value and standard deviation. On the other hand, the standard deviation of the AHJAYA is smaller than other algorithms, which means that the AHJAYA is more stable. In addition, the Wilcoxon Signed Ranks test visually shows the direct difference of the algorithm. In the data in Table 6, the proposed AHJAYA algorithm has obvious advantages among the four algorithms and ranks first.

The best RMSE values and the corresponding extracted five parameters are shown in Table 7. The accuracy of these parameters cannot be determined from a numerical point of view alone. Therefore, these values are reintroduced into the function and used to calculate the simulated current values. Figure 2(a),(b) show the fitting curves of measured current and simulated current, and measured power and simulated power, respectively. From Figure 2, it can be clearly seen that the five parameters extracted by the AHJAYA are very accurate, because the simulated data can match the real data well.

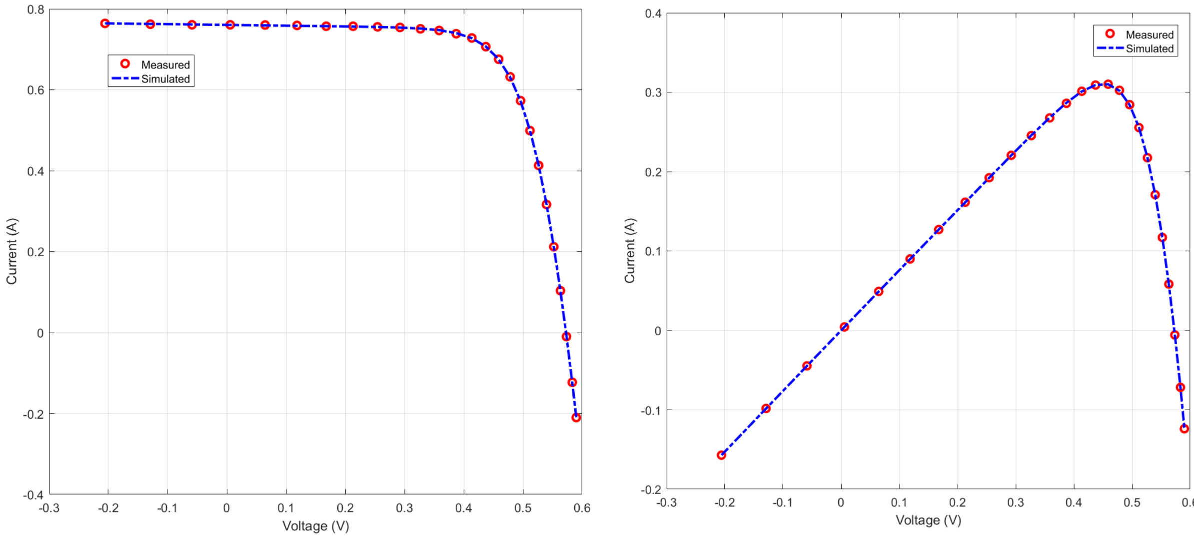
The accuracy of these algorithms is analyzed by the above experiments, and the convergence also needs to be analyzed. Therefore, the convergence curves of these algorithms on the single-diode model are shown in Figure 3. It can be seen from Figure 3 that the proposed AHJAYA converges faster than the other three algorithms.

Table 6. Results of the single-diode model.

Algorithm	RMSE				Wilcoxon Signed Ranks test				
	Best	Worst	Mean	Std	*R*+	*R*-	*P*-value	Ranking	Sig
AHJAYA	**9.86021878 × 10^{-4}**	**9.86021938 × 10^{-4}**	**9.86021880 × 10^{-4}**	**1.10061104 × 10^{-11}**	–	–	–	**1.65**	–
PGJAYA	**9.86021878 × 10^{-4}**	2.14258455 × 10^{-3}	1.06065486 × 10^{-3}	2.17305257 × 10^{-4}	396.0	39.0	2.88 × 10^{-6}	2.6667	+
JAYA	**9.86021878 × 10^{-4}**	9.86912488 × 10^{-4}	9.86059764 × 10^{-4}	1.65198791 × 10^{-7}	289.5	175.5	0.017518	1.8167	+
EJAYA	9.91219258 × 10^{-4}	1.29548522 × 10^{-3}	1.10739632 × 10^{-3}	8.21122196 × 10^{-5}	465.0	0.0	1.73 × 10^{-6}	3.8667	+

Table 7. Extracted parametric results on the single-diode model.

Algorithm	$\boldsymbol{I_{pv}}$(A)	$\boldsymbol{I_{sd}}$($\boldsymbol{\mu}$A)	$\boldsymbol{R_S}$(Ω)	$\boldsymbol{R_P}$(Ω)	$\boldsymbol{n}$	RMSE
AHJAYA	0.760775530	0.32302081	0.036377093	53.71852177	1.481183591	**9.86021878 × 10^{-4}**
PGJAYA	0.760775519	0.32302066	0.036377098	53.71864070	1.481183543	**9.86021878 × 10^{-4}**
JAYA	0.760775530	0.32302083	0.036377092	53.71852852	1.481183598	**9.86021878 × 10^{-4}**
EJAYA	0.7603639579	0.36597422	0.035795327	58.53225747	1.493883074	9.91219258 × 10^{-4}



(a) Fitting curve of measured current and simulated current obtained by AHJAYA

(b) Fitting curve of measured power and simulated power obtained by AHJAYA

Figure 2. The Fitting curve between the measured data and the simulated data is obtained by the AHJAYA on the single-diode model.

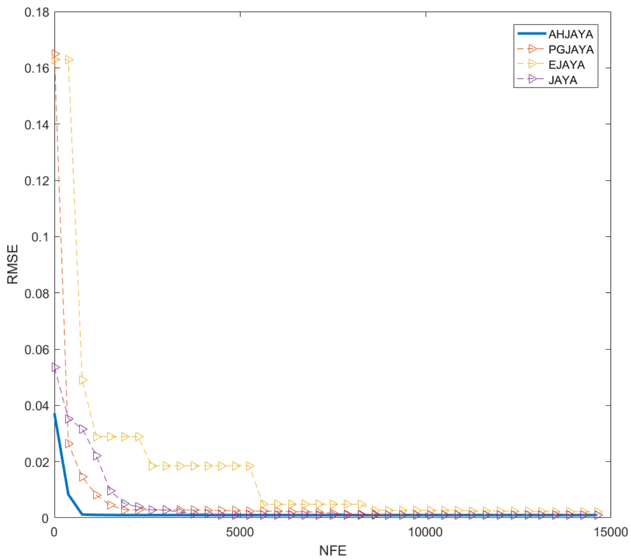


Figure 3. Comparison of the four algorithms on the convergence curve on the single-diode model.

5.6. Experimental results of the double-diode model

Different from the single-diode model, the double-diode model has two more unknown parameters that need to be extracted, which undoubtedly increases the complexity of the problem. The RMSE obtained by these algorithms on the double-diode model are shown in Table 8. From the data in this table, the proposed AHJAYA achieves optimal values in all four aspects. On the other hand, the experimental results of the Wilcoxon Signed Ranks test also prove that the AHJAYA algorithm is superior to other algorithms, ranking first overall. In addition, the best RMSE and the corresponding extracted parameters obtained by the four algorithms are shown in Table 9. To verify the accuracy of the proposed AHJAYA, the extracted 7 parameters are substituted into the function to calculate the simulated current and power. On the double-diode model, the fitting curve between the measured data and the simulated data calculated by the AHJAYA is shown in Figure 4. It can be seen from Figure 4 that the simulated calculated data fit the measured data. Therefore, the AHJAYA exhibits superior performance on the double-diode model.

On the other hand, Figure 5 shows the convergence curves of the four algorithms on the two-diode model. There is no doubt that the proposed AHJAYA converges faster after less evaluation times, and its performance is very superior.

Table 8. Results of the double-diode model.

Algorithm	RMSE				Wilcoxon Signed Ranks test				
	Best	Worst	Mean	Std	R+	R-	P-value	Ranking	Sig
AHJAYA	**9.82487154 × 10⁻⁴**	**9.90382279 × 10⁻⁴**	**9.85493674 × 10⁻⁴**	**1.86113881 × 10⁻⁶**	–	–	–	**1.6**	–
PGJAYA	9.83949066 × 10^{-4}	2.78220880 × 10^{-3}	1.23245866 × 10^{-3}	4.61314505 × 10^{-4}	446.0	19.0	1.13 × 10^{-5}	2.7	+
JAYA	9.82612337 × 10^{-4}	1.45994353 × 10^{-3}	1.04945516 × 10^{-3}	1.22731480 × 10^{-4}	331.0	134.0	0.042767	1.9667	+
EJAYA	9.91361757 × 10^{-4}	2.44975182 × 10^{-3}	1.56518485 × 10^{-3}	4.02468734 × 10^{-4}	465.0	0.0	1.73 × 10^{-6}	3.7333	+

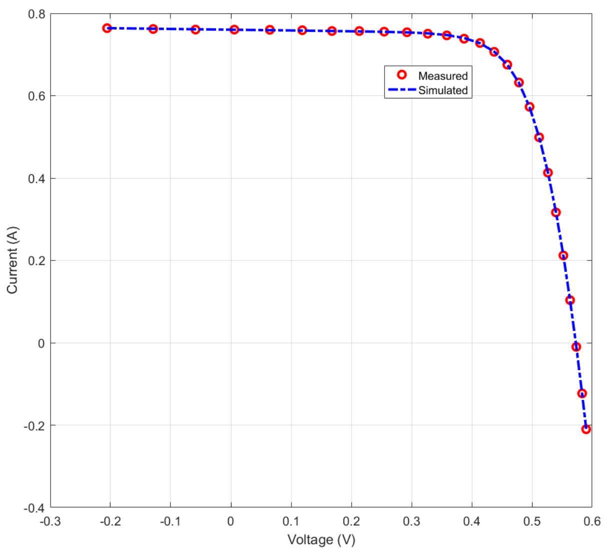
Table 9. Extracted parametric results on the double-diode model.

Algorithm	I_{pv}(A)	I_{sd_1}(μA)	R_S(Ω)	R_P(Ω)	n_1	I_{sd_2}(μA)	n_2	RMSE
AHJAYA	0.760769937	0.24275673	0.036659994	55.27879521	1.457039771	0.61290764	1.999999894	**9.82487154 × 10^{-4}**
PGJAYA	0.760780015	0.21712876	0.036700105	55.07646837	1.449218387	0.48025473	1.867436800	9.83949066 × 10^{-4}
JAYA	0.760780001	0.61575991	0.036670449	55.14992386	1.999995543	0.24182516	1.456685393	9.82612337 × 10^{-4}
EJAYA	0.760753139	0.27549484	0.035885453	57.02338365	1.498412038	9.11113904	1.481853469	9.91361757 × 10^{-4}

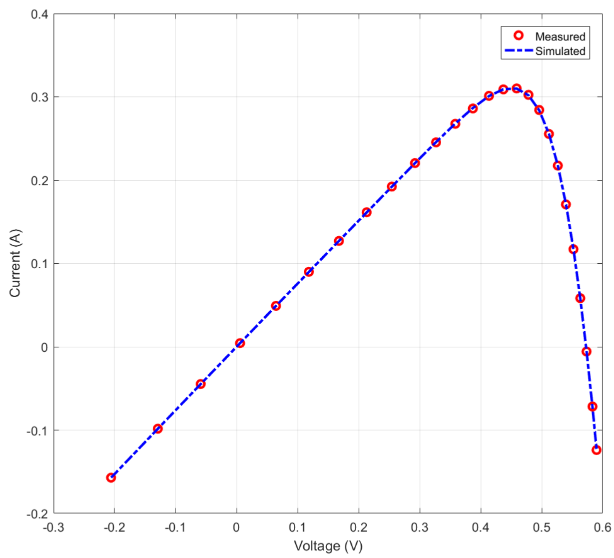
5.7. Experimental results of the STM6-40/36

The statistical results obtained by the four algorithms are shown in Table 10. From the data in the table, whether it is the optimal value, the worst value, the average value, and the standard deviation, the proposed AHJAYA shows excellent performance. The results of the Wilcoxon Signed Ranks test also prove the superiority of the AHJAYA algorithm compared to other algorithms, and it still ranks first. The optimal RMSE and the corresponding extracted parameters obtained by the four algorithms are shown in Table 11. The parameters extracted by the AHJAYA in Table 11 are brought into the function to recalculate the simulated current and power. Figure 6 shows the fitting curve of the measured data and the simulated data. It can be seen from the Figure 6 that the simulated data agrees with the measured data. This also proves the superior performance of the AHJAYA on the PV module model.

The convergence curves of the four algorithms on STM6-40/36 are drawn in Figure 7. It can be seen intuitively from the Figure 7 that the proposed AHJAYA has faster convergence speed and better performance.



(a) Fitting curve of measured current and simulated current obtained by AHJAYA.



(b) Fitting curve of measured power and simulated power obtained by AHJAYA.

Figure 4. The Fitting curve between the measured data and the simulated data is obtained by the AHJAYA on the double-diode model.

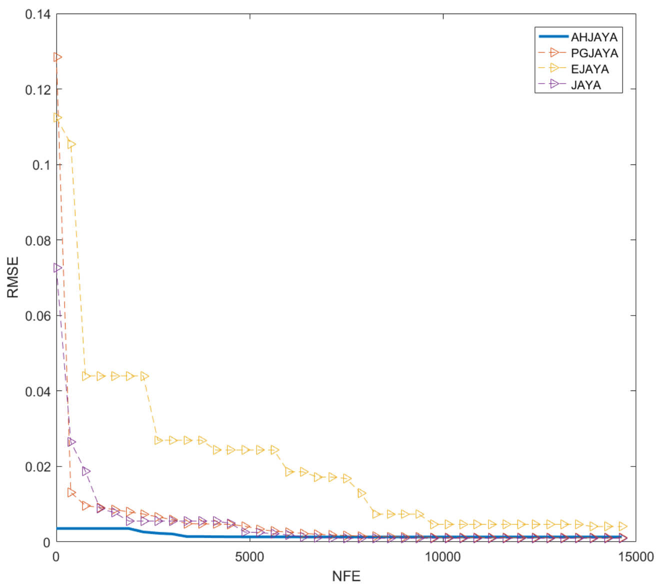
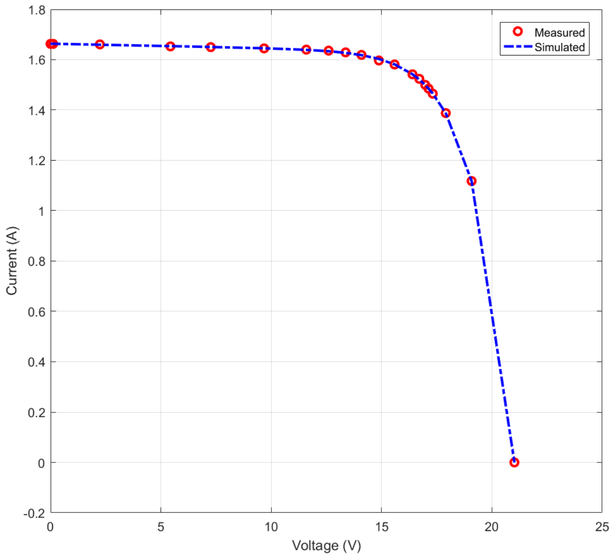
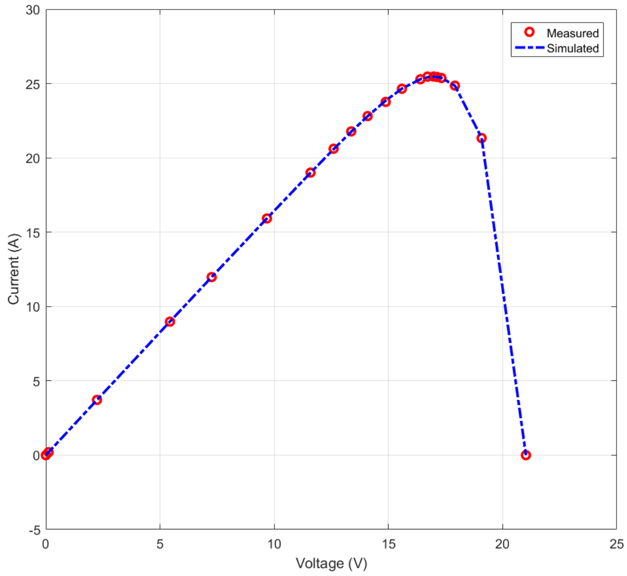


Figure 5. Comparison of the four algorithms on the convergence curve on the double-diode model.



(a) Fitting curve of measured current and simulated current obtained by AHJAYA.



(b) Fitting curve of measured power and simulated power obtained by AHJAYA.

Figure 6. The Fitting curve between the measured data and the simulated data is obtained by the AHJAYA on the STM6-40/36.

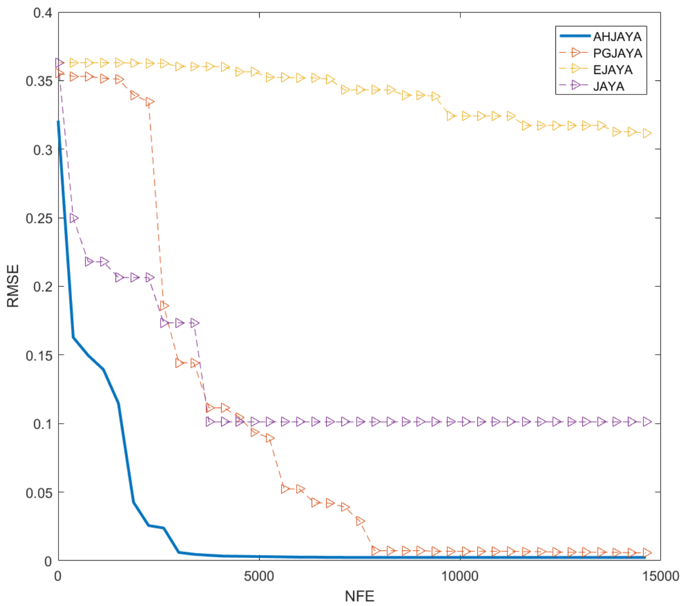


Figure 7. Comparison of the four algorithms on the convergence curve on the STM6-40/36.

5.8. Experimental results of the STP6-120/36

For the polycrystalline STP6-120/36 PV module model, Table 10 presents the statistical results obtained by the four algorithms. The best RMSE is obtained by AHJAYA and JAYA algorithm within 30 times. On the other hand, only the proposed AHJAYA exhibits superior performance in the worst value, the mean value, and the standard deviation. This also proves that the AHJAYA is more accurate and has less error than other algorithms. Similarly, from the results of the Wilcoxon Signed Ranks test, the superiority of AHJAYA has been proved again, and the overall ranking is still the first. The fitting curve of the measured data and the simulated data obtained by the AHJAYA is shown in Figure 8. It can be seen from the Figure 8 that the degree of curve fitting is very ideal. This further proves the accuracy of the AHJAYA.

Figure 9 shows the convergence curves of the four algorithms on the polycrystalline STP6-120/36 model. Compared with other algorithms, the AHJAYA converges faster and has less error. It is worth noting that the PGJAYA can also converge quickly, but the performance is slightly weaker than the AHJAYA.

Table 10. Results of the STM6-40/36.

Algorithm	RMSE				Wilcoxon Signed Ranks test				
	Best	Worst	Mean	Std	$R+$	R-	P-value	Ranking	Sig
AHJAYA	**1.72981371 × 10^{-3}**	**1.73632117 × 10^{-3}**	**1.73016570 × 10^{-3}**	**1.27864305 × 10^{-6}**	–	–	–	**1.0333**	–
PGJAYA	1.73260649 × 10^{-3}	8.80770029 × 10^{-2}	7.65418704 × 10^{-3}	1.73520638 × 10^{-2}	458.0	7.0	3.52 × 10^{-6}	2.2667	+
JAYA	1.77421268 × 10^{-3}	0.310756394	6.09604410 × 10^{-2}	7.00785933 × 10^{-2}	465.0	0.0	1.73 × 10^{-6}	3.3	+
EJAYA	2.91991914 × 10^{-3}	1.26039985	2.70636929 × 10^{-2}	2.73980326 × 10^{-2}	465.0	0.0	1.73 × 10^{-6}	3.4	+

Table 11. Extracted parametric results on the STM6-40/36.

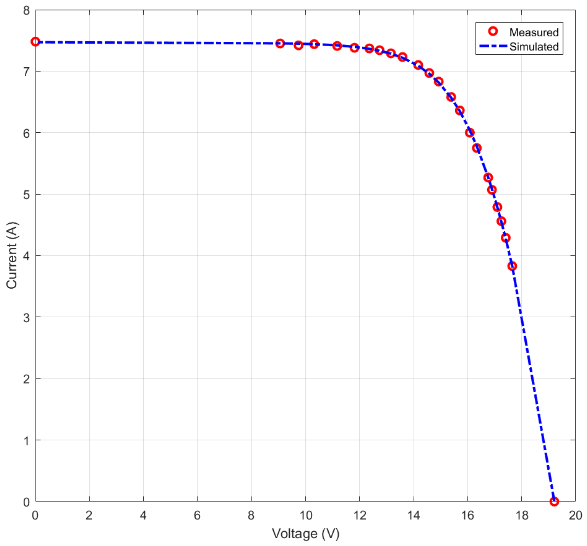
Algorithm	$\boldsymbol{I_{pv}}$(A)	$\boldsymbol{I_{sd}}$($\boldsymbol{\mu}$A)	$\boldsymbol{R_S}$(Ω)	$\boldsymbol{R_P}$(Ω)	$\boldsymbol{n}$	RMSE
AHJAYA	1.663904777	1.73865693	0.004273771	15.92829412	1.520302923	**1.72981371 × 10^{-3}**
PGJAYA	1.664361048	1.52929620	0.004695732	15.26160020	1.506340859	1.73260649 × 10^{-3}
JAYA	1.663533208	2.05368089	0.003749887	16.82518075	1.538826938	1.77421268 × 10^{-3}
EJAYA	1.682022065	2.55648595	0.001631409	9.240291210	1.564878675	2.91991914 × 10^{-3}

Table 12. Results of the STP6-120/36.

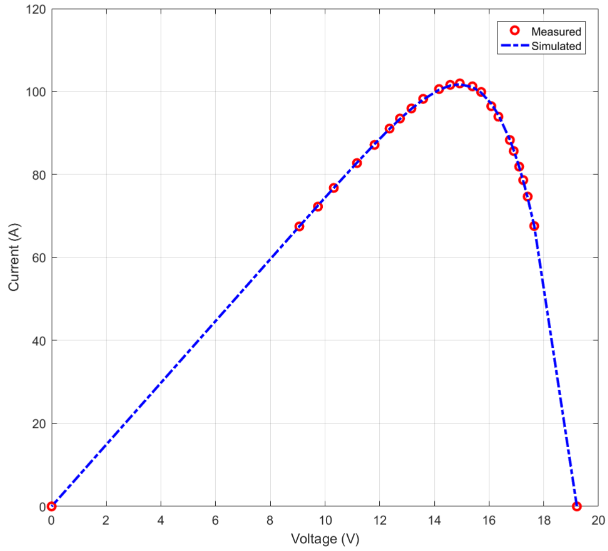
Algorithm	RMSE				Wilcoxon Signed Ranks test				
	Best	Worst	Mean	Std	$R+$	R-	P-value	Ranking	Sig
AHJAYA	**1.66006031 × 10^{-2}**	**1.66579770 × 10^{-2}**	**1.66043640 × 10^{-2}**	**1.13952391 × 10^{-5}**	–	–	–	**1.05**	–
PGJAYA	1.66007020 × 10^{-2}	0.955660164	9.13108271 × 10^{-2}	0.216928478	462.0	3.0	2.35 × 10^{-6}	2.4667	+
JAYA	**1.66006031 × 10^{-2}**	0.948928454	0.292906570	0.351482388	465.0	0.0	1.92 × 10^{-6}	3.05	+
EJAYA	3.22271816 × 10^{-2}	0.883508238	0.175116862	0.216340554	465.0	0.0	1.73 × 10^{-6}	3.4333	+

Table 13. Extracted parametric results on the STP6-120/36.

Algorithm	I_{pv}(A)	$I_{sd}(\mu$A)	$R_S(\Omega)$	$R_P(\Omega)$	n	RMSE
AHJAYA	7.472529926	2.33499493	0.004594634	22.21989296	1.260103473	**$1.66006031 \times 10^{-2}$**
PGJAYA	7.461290096	2.47641968	0.004578694	142.7790881	1.264984632	$1.66007020 \times 10^{-2}$
JAYA	7.472529833	2.33499581	0.004594634	22.21998258	1.260103505	**$1.66006031 \times 10^{-2}$**
EJAYA	7.390327287	0.01786976	0.003455219	1354.451365	1.457573218	$3.22271816 \times 10^{-2}$



(a) Fitting curve of measured current and simulated current obtained by AHJAYA.



(b) Fitting curve of measured power and simulated power obtained by AHJAYA.

Figure 8. The Fitting curve between the measured data and the simulated data is obtained by the AHJAYA on the STP6-120/36.

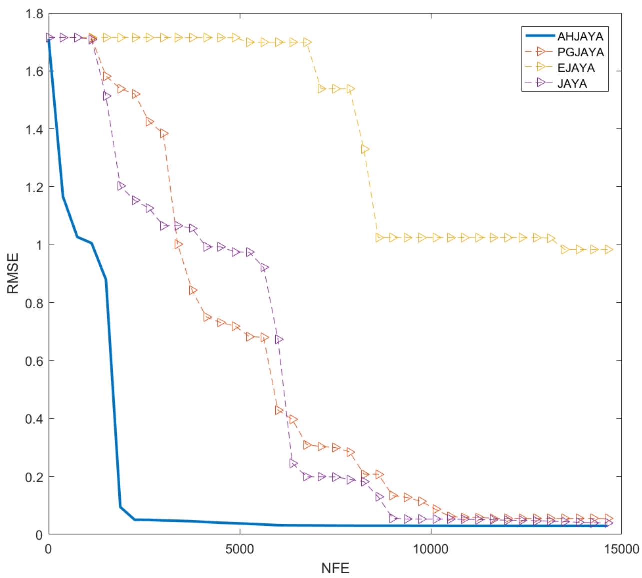


Figure 9. Comparison of the four algorithms on the convergence curve on the STP6-120/36.

5.9. Discussions of different components

Since three different improvement components are introduced, including a self-adaptive coefficient strategy, linear population reduction strategy, and chaotic opposition-based learning strategy. Therefore, it is necessary to analyze the effectiveness of different strategies by ablation study.

Because of there are three improvement components, six variants are developed, including AHJAYA-2 with only self-adaptive coefficient strategy, AHJAYA-3 with only linear population reduction strategy, AHJAYA-4 with only chaotic opposition-based learning strategy, AHJAYA-5 with self-adaptive coefficient strategy and linear population reduction strategy, AHJAYA-6 with self-adaptive coefficient strategy and chaotic opposition-based learning strategy and AHJAYA-7 with chaotic opposition-based learning strategy and linear population reduction strategy. The statistical experimental results and convergence curves are shown in Table 14 and Figure 10, respectively.

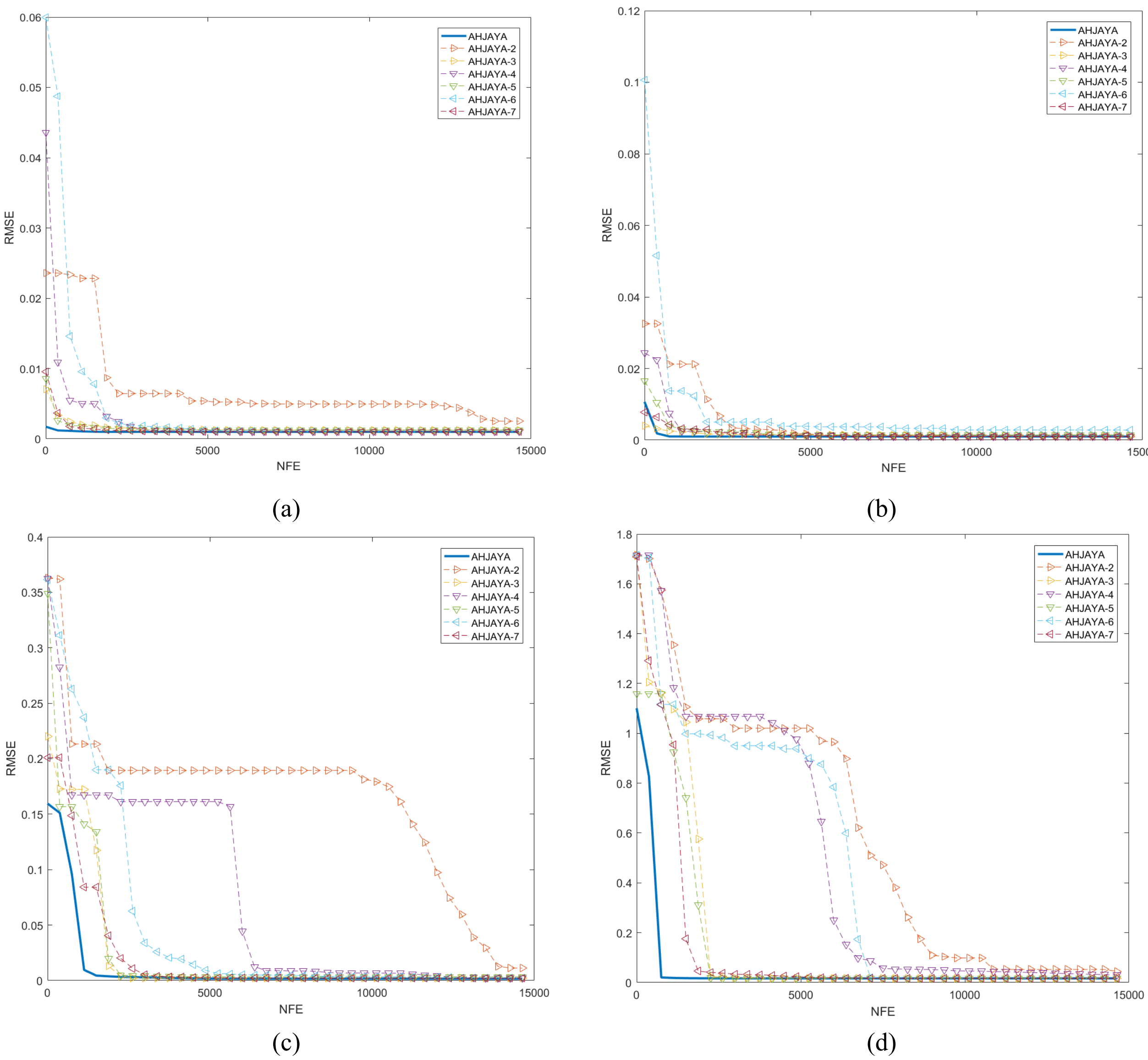


Figure 10. Convergence curves of different components in AHAJYA on PV models: (a) SDM, (b) DDM, (c) STM6-40/36, (d) STP6-120/36.

Table 14. Analysis of different components in AHJAYA for different PV models.

Model	Algorithm	RMSE			
		Best	Worst	Mean	Std
SDM	AHJAYA-2	**$9.86021878 \times 10^{-4}$**	$8.32514744 \times 10^{-3}$	$1.26527138 \times 10^{-3}$	$1.33943233 \times 10^{-3}$
	AHJAYA-3	**$9.86021878 \times 10^{-4}$**	1.22051981	$1.00262840 \times 10^{-3}$	$4.69406450 \times 10^{-5}$
	AHJAYA-4	**$9.86021878 \times 10^{-4}$**	$1.36934635 \times 10^{-3}$	$9.98805720 \times 10^{-4}$	$6.99839566 \times 10^{-5}$
	AHJAYA-5	**$9.86021878 \times 10^{-4}$**	$2.43770117 \times 10^{-3}$	$1.11861029 \times 10^{-3}$	$3.09270611 \times 10^{-4}$
	AHJAYA-6	**$9.86021878 \times 10^{-4}$**	$5.02575424 \times 10^{-3}$	$1.15966446 \times 10^{-3}$	$7.40695595 \times 10^{-4}$
	AHJAYA-7	**$9.86021878 \times 10^{-4}$**	$9.86266559 \times 10^{-4}$	$9.86042789 \times 10^{-4}$	$6.31049833 \times 10^{-8}$
	AHJAYA	**$9.86021878 \times 10^{-4}$**	**$9.86021938 \times 10^{-4}$**	**$9.86021880 \times 10^{-4}$**	**$1.10061104 \times 10^{-11}$**
DDM	AHJAYA-2	$9.84892255 \times 10^{-4}$	$2.07867164 \times 10^{-3}$	$1.16744223 \times 10^{-3}$	$2.95964194 \times 10^{-4}$
	AHJAYA-3	$9.82505888 \times 10^{-4}$	$1.97806747 \times 10^{-3}$	$1.11322952 \times 10^{-3}$	$2.12030912 \times 10^{-4}$
	AHJAYA-4	$9.83248764 \times 10^{-4}$	$9.91318178 \times 10^{-4}$	$9.85473135 \times 10^{-4}$	$1.59661583 \times 10^{-6}$
	AHJAYA-5	$9.84788755 \times 10^{-4}$	$2.25301378 \times 10^{-3}$	$1.23099472 \times 10^{-3}$	$3.24447231 \times 10^{-4}$
	AHJAYA-6	$9.82650184 \times 10^{-4}$	$1.27173338 \times 10^{-3}$	$1.01036772 \times 10^{-3}$	$7.38763464 \times 10^{-5}$
	AHJAYA-7	$9.82538855 \times 10^{-4}$	$9.91093795 \times 10^{-4}$	$9.85382667 \times 10^{-4}$	$1.54241939 \times 10^{-6}$
	AHJAYA	**$9.82488468 \times 10^{-4}$**	**$9.86203608 \times 10^{-4}$**	**$9.84766563 \times 10^{-4}$**	**$1.38004612 \times 10^{-6}$**
STM6-40/36	AHJAYA-2	$1.93984355 \times 10^{-3}$	0.186748526	$4.79984858 \times 10^{-2}$	$4.84300989 \times 10^{-2}$
	AHJAYA-3	$1.73273321 \times 10^{-3}$	$6.95388282 \times 10^{-3}$	$3.33409243 \times 10^{-3}$	$1.12312499 \times 10^{-3}$
	AHJAYA-4	$1.73457370 \times 10^{-3}$	$3.99357393 \times 10^{-3}$	$2.51059137 \times 10^{-3}$	$5.90157813 \times 10^{-4}$
	AHJAYA-5	$2.35383504 \times 10^{-3}$	$5.79530516 \times 10^{-2}$	$9.81840973 \times 10^{-3}$	$1.45183654 \times 10^{-2}$
	AHJAYA-6	**$1.72981371 \times 10^{-3}$**	$2.52479525 \times 10^{-2}$	$4.38971519 \times 10^{-3}$	$5.35198180 \times 10^{-3}$
	AHJAYA-7	$1.72981391 \times 10^{-3}$	$2.12677114 \times 10^{-3}$	$1.86241971 \times 10^{-3}$	$1.19387377 \times 10^{-4}$
	AHJAYA	**$1.72981371 \times 10^{-3}$**	**$2.12613108 \times 10^{-3}$**	**$1.77413665 \times 10^{-3}$**	**$1.18880229 \times 10^{-4}$**
STP6-120/36	AHJAYA-2	$1.66122817 \times 10^{-2}$	0.872035345	0.215236723	0.276920517
	AHJAYA-3	$1.66128591 \times 10^{-2}$	$4.82478355 \times 10^{-2}$	$2.61314099 \times 10^{-2}$	$9.23757503 \times 10^{-3}$
	AHJAYA-4	$1.66006056 \times 10^{-2}$	$3.23255844 \times 10^{-2}$	$1.98005470 \times 10^{-2}$	$4.41034383 \times 10^{-3}$
	AHJAYA-5	$1.66342237 \times 10^{-2}$	0.504667897	$6.73033137 \times 10^{-2}$	0.107342292
	AHJAYA-6	$1.66006041 \times 10^{-2}$	0.334609995	$2.88070799 \times 10^{-2}$	$5.77937642 \times 10^{-2}$
	AHJAYA-7	**$1.66006031 \times 10^{-2}$**	$1.83025327 \times 10^{-2}$	$1.68013974 \times 10^{-2}$	$3.45831872 \times 10^{-4}$
	AHJAYA	**$1.66006031 \times 10^{-2}$**	**$1.68253440 \times 10^{-2}$**	**$1.66213999 \times 10^{-2}$**	**$4.94692736 \times 10^{-5}$**

For the single-diode model, it can be seen from Figure 10(a) that AHJAYA can achieve higher accuracy faster than the other six variant algorithms. From the data in Table 14, it can be seen that the best value can be taken by seven algorithms, but only the AHJAYA algorithm performs the best in the other three aspects (including the worst, mean, and standard deviation). Therefore, in this single diode model, the AHJAYA has the best overall performance.

For the double-diode model, it can be seen from Figure 10(b) that AHJAYA-3, AHJAYA-4, AHJAYA-5, AHJAYA-7 and AHJAYA can all show very strong competitiveness. It can be seen from the data in Table 14 that only the AHJAYA algorithm performs the best in four aspects. Therefore, on the double -diode model, the AHJAYA performs best.

For STM6-40/36, it can be intuitively seen from Figure 10(c) that AHJAYA performs better than the other six algorithms. From the data in Table 14, it can be seen that the best values are obtained by

AHJAYA-6 and AHJAYA, in addition, the AHJAYA also performs the best in the other three aspects (including the worst, mean, and standard deviation). Overall, AHJAYA performs better on this PV model module.

For STP6-120/36, similarly, from the comprehensive performance in Figure 10(d) and Table 14, AHJAYA not only performs the best in convergence, but also in the four aspects of RMSE. Therefore, it is still the proposed AHJAYA that performs the best in this PV model module.

5.10. Comparison of AHJAYA with other mature algorithms

To further verify the superiority of the AHJAYA, in this section, the proposed AHJAYA is compared with other published algorithms. Most of the data in this section are obtained from [43]. The specific details are shown in Table 15 to Table 22. Where NA represents that the paper does not give data.

Table 15. Comparison of the results of the AHJAYA with other mature algorithms on the single-diode model.

Algorithm	RMSE				NFES
	Best	Worst	Mean	Std	
GOTLBO (2016) [56]	9.8744×10^{-4}	1.9824×10^{-3}	1.3349×10^{-3}	2.09×10^{-4}	10,000
SATLBO (2017)[57]	9.8602×10^{-4}	9.9494×10^{-3}	9.8780×10^{-4}	2.03×10^{-6}	50,000
IJAYA (2017) [58]	9.8603×10^{-4}	1.0622×10^{-3}	9.9204×10^{-4}	1.40×10^{-5}	50,000
TLABC (2018) [59]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	1.86×10^{-5}	50,000
MLBSA (2018) [52]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	9.15×10^{-12}	50,000
DE/WOA (2018) [44]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	3.55×10^{-17}	50,000
OBWOA (2018) [7]	9.8602×10^{-4}	NA	9.8603×10^{-3}	1.02×10^{-8}	1,500,000
ITLBO (2019) [45]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	2.19×10^{-17}	50,000
PGJAVA (2019) [42]	9.8602×10^{-4}	9.8603×10^{-4}	9.8602×10^{-4}	1.45×10^{-9}	50,000
BHCS (2019) [46]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	2.61×10^{-17}	50,000
FPSO (2019) [60]	9.8602×10^{-4}	NA	NA	NA	NA
ILCOA (2019) [53]	9.8602×10^{-4}	NA	NA	1.01×10^{-8}	10,000 × NP
BSARDVs (2020) [61]	9.8602×10^{-4}	NA	NA	NA	25,000
ELBA (2020) [47]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	1.97×10^{-17}	15,000
EOTLBO (2020) [48]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	4.13×10^{-17}	20,000
CLJAYA (2020) [62]	9.8602×10^{-4}	NA	NA	NA	20,000
CBSA (2020) [63]	9.8602×10^{-4}	NA	NA	NA	25,000
ATLDE (2020) [14]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	2.44×10^{-17}	30,000
EJAYA (2021) [49]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	6.80×10^{-17}	30,000
IGSK (2021) [50]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	3.58×10^{-17}	10,000
EABOA (2021) [54]	9.8602×10^{-4}	9.8784×10^{-4}	9.8678×10^{-4}	9.30×10^{-7}	50,000
SFLBS (2021) [55]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	1.43×10^{-14}	60,000
RLDE (2021) [51]	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	3.48×10^{-17}	30,000
AHJAYA	9.8602×10^{-4}	9.8602×10^{-4}	9.8602×10^{-4}	2.92×10^{-17}	25,000

For the single-diode model, from the four values of RMSE, only DE/WOA [44], ITLBO [45], BHCS [46], ELBA [47], EOTLBO [48], ATLDE [14], EJAYA [49], IGSK [50], RLDE [51] and

AHJAYA algorithm perform better. Because their values are smaller and more precise. On the other hand, only the IGSK, EOTLBO, and ELBA use fewer evaluation times than the AHJAYA. Fewer evaluation times are used, which means less resources are consumed, therefore, the proposed AHJAYA is highly competitive in these algorithms.

For the double-diode model, the algorithms that are superior in the four aspects of RMSE are MLBSA [52], DE/WOA, OBWOA [7], PGJAYA [42], BHCS, ILCOA [53], ITBLO, ELBA, ATLDE, EJAYA, IGSK, EABOA [54], SFLBS [55], RLDE and AHJAYA algorithm. However, among these algorithms, only the IGSK uses less evaluation times than the AHJAYA. Therefore, in this model, the AHJAYA is very competitive and has an absolute advantage.

Table 16. Comparison of the results of the AHJAYA with other mature algorithms on the double-diode model.

Algorithm	RMSE				NFES
	Best	Worst	Mean	Std	
GOTLBO (2016) [56]	9.8318×10^{-4}	1.7877×10^{-3}	1.2436×10^{-3}	2.09×10^{-4}	20,000
SATLBO (2017) [57]	9.8280×10^{-4}	1.0470×10^{-3}	9.9811×10^{-4}	1.95×10^{-5}	50,000
IJAYA (2017) [58]	9.8293×10^{-4}	1.4055×10^{-3}	1.0269×10^{-3}	9.83×10^{-5}	50,000
TLABC (2018) [59]	9.8415×10^{-4}	1.5048×10^{-3}	1.0555×10^{-3}	1.55×10^{-5}	50,000
MLBSA (2018) [52]	9.8249×10^{-4}	9.8798×10^{-4}	9.8518×10^{-4}	1.35×10^{-6}	50,000
DE/WOA (2018) [44]	9.8248×10^{-4}	9.8603×10^{-4}	9.8297×10^{-4}	9.15×10^{-7}	50,000
OBWOA (2018) [7]	9.8251×10^{-4}	NA	9.8294×10^{-4}	1.13×10^{-7}	1,500,000
PGJAYA (2019) [42]	9.8263×10^{-4}	9.9499×10^{-4}	9.8582×10^{-4}	2.54×10^{-6}	50,000
BHCS (2019) [46]	9.8249×10^{-4}	9.8687×10^{-4}	9.8380×10^{-4}	1.54×10^{-6}	50,000
FPSO (2019) [60]	9.8253×10^{-4}	NA	NA	NA	NA
ILCOA (2019) [53]	8.8257×10^{-4}	NA	NA	6.25×10^{-7}	$10{,}000 \times NP$
ITLBO (2019) [45]	9.8248×10^{-4}	9.8812×10^{-4}	9.8497×10^{-4}	1.54×10^{-6}	50,000
BSARDVs (2020) [61]	9.8248×10^{-4}	NA	NA	NA	45,000
ELBA (2020) [47]	9.8248×10^{-4}	9.8615×10^{-4}	9.8349×10^{-4}	6.25×10^{-7}	50,000
EOTLBO (2020) [48]	9.8248×10^{-4}	9.8942×10^{-4}	9.8473×10^{-4}	1.54×10^{-5}	20,000
CLJAYA (2020) [62]	9.8249×10^{-4}	NA	NA	NA	48,000
CBSA (2020) [63]	9.8248×10^{-4}	NA	NA	NA	50,000
ATLDE (2020) [14]	9.8248×10^{-4}	9.8603×10^{-4}	9.8372×10^{-4}	1.37×10^{-6}	30,000
EJAYA (2021) [49]	9.8248×10^{-4}	9.8602×10^{-4}	9.8448×10^{-4}	1.51×10^{-6}	30,000
IGSK (2021) [50]	9.8248×10^{-4}	9.8602×10^{-4}	9.8273×10^{-4}	8.96×10^{-7}	20,000
EABOA (2021) [54]	9.8607×10^{-4}	1.0012×10^{-3}	9.9190×10^{-4}	6.62×10^{-6}	50,000
SFLBS (2021) [55]	9.8249×10^{-4}	9.8787×10^{-4}	9.8541×10^{-4}	1.79×10^{-6}	60,000
RLDE (2021) [51]	9.8248×10^{-4}	9.8695×10^{-4}	9.8457×10^{-4}	1.75×10^{-6}	30,000
AHJAYA	9.8248×10^{-4}	9.8919×10^{-4}	9.8475×10^{-4}	1.64×10^{-6}	25,000

For STM6-40/36, all algorithms except the BHCS can perform well on the four values of RMSE. However, only the IGSK has fewer evaluation times than the AHJAYA, which proves that the AHJAYA still has advantages over other mature algorithms.

For STP6-120/36, similarly, all algorithms except the BHCS can perform very well, and only the IGSK has less evaluation times than the AHJAYA. Therefore, the AHJAYA is in a leading position among other mature algorithms.

Table 17. Comparison of the results of the AHJAYA with other mature algorithms on the STM6-40/36.

Algorithm	RMSE				NFES
	Best	Worst	Mean	Std	
BHCS (2019) [46]	1.7298×10^{-3}	3.3299×10^{-3}	1.8365×10^{-3}	4.06×10^{-4}	50,000
ITLBO (2019) [45]	1.7298×10^{-3}	1.7298×10^{-3}	1.7298×10^{-3}	4.75×10^{-18}	50,000
ELBA (2020) [47]	1.7298×10^{-3}	1.7298×10^{-3}	1.7298×10^{-3}	6.16×10^{-18}	50,000
ATLDE (2020) [14]	1.7298×10^{-3}	1.7298×10^{-3}	1.7298×10^{-3}	8.22×10^{-18}	30,000
EJAYA (2021) [49]	1.7298×10^{-3}	1.7298×10^{-3}	1.7298×10^{-3}	1.47×10^{-17}	30,000
IGSK (2021) [50]	1.7298×10^{-3}	1.7298×10^{-3}	1.7298×10^{-3}	7.02×10^{-18}	15,000
RLDE (2021) [51]	1.7298×10^{-3}	1.7298×10^{-3}	1.7298×10^{-3}	1.58×10^{-17}	30,000
AHJAYA	1.7298×10^{-3}	1.7298×10^{-3}	1.7298×10^{-3}	2.58×10^{-17}	25,000

Table 18. Comparison of the results of the AHJAYA with other mature algorithms on the STP6-120/36.

Algorithm	RMSE				NFES
	Best	Worst	Mean	Std	
BHCS (2019) [46]	1.6601×10^{-2}	0.13482	2.4360×10^{-2}	2.61×10^{-2}	50,000
ITLBO (2019) [45]	1.6601×10^{-2}	1.6601×10^{-2}	1.6601×10^{-2}	7.22×10^{-17}	50,000
ATLDE (2020) [14]	1.6601×10^{-2}	1.6601×10^{-2}	1.6601×10^{-2}	1.02×10^{-16}	30,000
EJAYA (2021) [49]	1.6601×10^{-2}	1.6601×10^{-2}	1.6601×10^{-2}	2.68×10^{-16}	30,000
IGSK (2021) [50]	1.6601×10^{-2}	1.6601×10^{-2}	1.6601×10^{-2}	1.71×10^{-16}	15,000
RLDE (2021) [51]	1.6601×10^{-2}	1.6601×10^{-2}	1.6601×10^{-2}	1.98×10^{-16}	30,000
AHJAYA	1.6601×10^{-2}	1.6601×10^{-2}	1.6601×10^{-2}	1.24×10^{-16}	25,000

Table 19. Comparison of extracted parameters between the AHJAYA and other mature algorithms on the single-diode model.

Algorithm	$\boldsymbol{I_{pv}}$(A)	$\boldsymbol{I_{sd}}$($\boldsymbol{\mu}$A)	$\boldsymbol{R_S}$(Ω)	$\boldsymbol{R_P}$(Ω)	$\boldsymbol{n}$	RMSE
GOTLBO (2016) [56]	0.7608	0.3316	0.0363	54.1154	1.4838	9.8744×10^{-4}
IJAYA (2017) [58]	0.7608	0.3228	0.0364	53.7595	1.4811	9.8603×10^{-4}
SATLBO (2017) [57]	0.7608	0.3232	0.0363	53.7295	1.4812	9.8602×10^{-4}
CWOA (2017) [64]	0.76077	0.3239	0.03636	53.742465	1.4812	9.8602×10^{-4}
MSSO (2017) [65]	0.760777	0.323564	0.036370	53.742465	1.481244	9.8607×10^{-4}
IWOA (2018) [66]	0.7608	0.3232	0.0364	53.7317	1.4812	9.8602×10^{-4}
HFAPS (2018) [67]	0.760777	0.322622	0.0363819	53.6784	1.48106	9.8602×10^{-4}
TLABC (2018) [59]	0.76078	0.32302	0.03638	53.71636	1.48118	9.8602×10^{-4}
MLBSA (2018) [52]	0.7608	0.32302	0.0364	53.7185	1.4812	9.8602×10^{-4}

Continued on next page

Algorithm	I_{pv}(A)	I_{sd}(μA)	R_S(Ω)	R_P(Ω)	n	RMSE
DE/WOA (2018) [44]	0.760776	0.323021	0.036377	53.718524	1.481184	9.8602×10^{-4}
OBWOA (2018) [7]	0.76077	0.3232	0.0363	53.6836	1.5208	9.8602×10^{-4}
PGJAYA (2018) [42]	0.7608	0.3230	0.0364	53.7185	1.4812	9.8602×10^{-4}
BHCS (2019) [46]	0.76078	0.32302	0.03638	53.71852	1.48118	9.8602×10^{-4}
FPSO (2019) [60]	0.76077552	0.323020	0.036370	53.718520	1.48110817	9.8602×10^{-4}
ILCOA (2019) [53]	0.760775	0.323021	0.036377	53.718679	1.481108	9.8602×10^{-4}
ITLBO (2019) [45]	0.7608	0.3230	0.0364	53.7185	1.4812	9.8602×10^{-4}
BSARDVs (2020) [61]	0.760776	0.323021	0.036377	53.718520	1.481184	9.8602×10^{-4}
ELBA (2020) [47]	0.760776	0.323021	0.036377	53.718523	1.481185	9.8602×10^{-4}
EOTLBO (2020) [48]	0.76077553	0.32302083	0.03637709	53.71852514	1.48118359	9.8602×10^{-4}
SGDE (2020) [8]	0.76078	0.32302	0.036377	53.71853	1.481184	9.8602×10^{-4}
CLJAYA (2020) [62]	0.76078	0.3230208	0.0363771	53.718521	1.481184	9.8602×10^{-4}
CPMPSO (2020) [68]	0.760776	0.323021	0.036377	53.71852	1.481184	9.8602×10^{-4}
NPSOPC (2020) [69]	0.7608	0.3325	0.03639	53.7583	1.4814	9.8856×10^{-4}
CBSA (2020) [63]	0.760776	0.323021	0.036377	53.71852	1.481184	9.8602×10^{-4}
ATLDE (2020) [14]	0.76077553	0.32302082	0.03637712	53.71852699	1.48118359	9.8602×10^{-4}
EJAYA (2021) [49]	0.76078	0.32302	0.03638	53.71852	1.48118	9.8602×10^{-4}
IGSK (2021) [50]	0.76077553	0.323	0.036377093	53.71852532	1.481183592	9.8602×10^{-4}
EABOA (2021) [54]	0.760771077	0.322929	0.036379593	53.76600144	1.481153457	9.8602×10^{-4}
SFLBS (2021) [55]	0.76078	0.323021	0.03638	53.7185	1.481184	9.8602×10^{-4}
RLDE (2021) [51]	0.7608	0.3231	0.0364	53.7185	1.4812	9.8602×10^{-4}
AHJAYA	0.760775530	0.32302081	0.036377093	53.71852177	1.481183591	9.8602×10^{-4}

Table 20. Comparison of extracted parameters between the AHJAYA and other mature algorithms on the double-diode model.

Algorithm	I_{pv}(A)	I_{sd_1}(μA)	R_S(Ω)	R_P(Ω)	n_1	I_{sd_2}(μA)	n_2	RMSE
GOTLBO (2016) [56]	0.7608	0.8002	0.0368	56.0753	2	0.2205	1.4490	9.8318×10^{-4}
IJAYA (2017) [58]	0.7601	0.0050445	0.0376	77.8519	1.2186	0.75904	1.6247	9.8293×10^{-4}
SATLBO (2017) [57]	0.7608	0.2509	0.0366	55.1170	1.4598	0.5454	1.9994	9.8280×10^{-4}
CWOA (2017) [64]	0.76077	0.24150	0.03666	55.20160	1.45651	0.60000	1.98990	9.8272×10^{-4}

Continued on next page

Algorithm	I_{pv}(A)	I_{sd_1}(μA)	R_S(Ω)	R_P(Ω)	n_1	I_{sd_2}(μA)	n_2	RMSE
MSSO (2017) [65]	0.760748	0.234925	0.036688	55.714662	1.454255	0.671593	1.995305	9.8281×10^{-4}
IWOA (2018) [66]	0.7608	0.6771	0.0367	55.4082	2	0.2355	1.4545	9.8255×10^{-4}
HFAPS (2018) [67]	0.760781	0.225974	0.0367404	55.4855	1.45101	0.7493580	2.000000	9.8248×10^{-4}
TLABC (2018) [59]	0.76081	0.42394	0.03667	54.66797	1.90750	0.24011	1.45671	9.8415×10^{-4}
MLBSA (2018) [52]	0.7608	0.22728	0.0670	55.4612	1.4515	0.73835	2.0000	9.8249×10^{-4}
DE/WOA (2018) [44]	0.760781	0.225974	0.0367400	55.485437	1.451017	0.749346	2.000000	9.8248×10^{-4}
OBWOA (2018) [7]	0.76076	0.22990	0.03671	55.3990	1.49154	0.61956	2.000000	9.8251×10^{-4}
PGJAYA (2018) [42]	0.7608	0.21031	0.0368	55.8135	1.4450	0.88534	2.0000	9.8263×10^{-4}
BHCS (2019) [46]	0.76078	0.74935	0.03674	55.48544	2.00000	0.22597	1.45102	9.8249×10^{-4}
FPSO (2019) [60]	0.76078	0.22731	0.036737	55.39230	1.45160	0.72786	1.99969	9.8253×10^{-4}
ILCOA (2019) [53]	0.76078	0.22601	0.036739	55.5320	1.45101	0.74921	2.00000	9.8257×10^{-4}
ITLBO (2019) [45]	0.7608	0.2260	0.0367	55.4854	1.4510	0.7493	2.0000	9.8248×10^{-4}
SGDE (2020) [8]	0.76079	0.28070	0.036480	55.3667	1.46966	0.24996	1.93228	9.8441×10^{-4}
BSARDVs (2020) [61]	0.760781	0.225808	0.0367411	55.4874	1.45096	0.750861	2	9.8248×10^{-4}
ELBA (2020) [47]	0.760781	0.749338	0.03674	55.48544	2	0.225975	1.451018	9.8248×10^{-4}
EOTLBO (2020) [48]	0.76078108	0.22597468	0.03674043	55.48543568	1.45101692	0.74934431	2	9.8248×10^{-4}
CLJAYA (2020) [62]	0.76078	0.226051	0.03674	55.48599	1.45105	0.74876	1.99999	9.8249×10^{-4}
CPMPSO (2020) [68]	0.76078	0.74935	0.3674	55.48544	2	0.22597	1.45102	9.8248×10^{-4}
NPSOPC (2020) [69]	0.76078	0.25093	0.3663	55.117	1.45982	0.545418	1.99941	9.8208×10^{-4}
CBSA (2020) [63]	0.76078	0.2259739	0.3674	55.48544	1.451017	0.74935	2	9.8248×10^{-4}
ATLDE (2020) [14]	0.76078108	0.22597412	0.03674043	55.48544744	1.45101671	0.74934885	2.00000000	9.8248×10^{-4}

Continued on next page

Algorithm	I_{pv}(A)	I_{sd_1}(μA)	R_S(Ω)	R_P(Ω)	n_1	I_{sd_2}(μA)	n_2	RMSE
EJAYA (2021) [49]	0.76078	0.22597	0.03674	55.48509	1.45102	0.74934	2	9.8248×10^{-4}
IGSK (2021) [50]	0.760781079	0.7493	0.036740429	55.48543425	2	0.226	1.451016893	9.8248×10^{-4}
EABOA (2021) [54]	0.76082865	0.25072	0.0366266	55.3660129	1.45988481	0.72069	1.99997318	9.8607×10^{-4}
RLDE (2021) [51]	0.7608	0.226	0.0367	55.4847	2	0.7492	1.451	9.8248×10^{-4}
AHJAYA	0.760764957	0.37522564	0.036618133	54.83299515	1.872336230	0.24007472	1.457199516	9.8248×10^{-4}

Table 21. Comparison of extracted parameters between the AHJAYA and other mature algorithms on the STM6-40/36.

Algorithm	I_{pv}(A)	I_{sd}(μA)	R_S(Ω)	R_P(Ω)	n	RMSE
CWOA (2017)[64]	1.7	1.6338	0.0050	15.4	1.5	1.8000×10^{-3}
HFAPS (2018)[67]	1.6663	1.0703	0.24849	490.03	53.016	1.9700×10^{-3}
OBWOA (2018) [7]	1.6642	1.65025	0.0044	15.5299	1.51424	1.7530×10^{-3}
BHCS (2019) [46]	1.66390	1.73866	0.00427	15.92829	1.52030	1.7298×10^{-3}
FPSO (2019) [60]	1.2323	7.4732	0.0049	9.6889	1.2086	1.3000×10^{-3}
ILCOA (2019) [53]	1.2001	7.4812	0.0049	9.6991	1.2067	1.6932×10^{-2}
ITLBO (2019) [45]	1.6639	1.7387	0.0043	15.9283	1.5203	1.7298×10^{-3}
ELBA (2020) [47]	1.663905	1.738657	0.004274	15.928294	1.520305	1.7298×10^{-3}
ATLDE (2020) [14]	1.66390478	1.73865697	0.00427377	15.92829439	1.52030293	1.7298×10^{-3}
EJAYA (2021) [49]	1.6639	1.73866	0.00427	15.92829	1.5203	1.7298×10^{-3}
IGSK (2021) [50]	1.663904777	1.7387	0.004273771	15.92829435	1.520302921	1.7298×10^{-3}
RLDE (2021) [51]	1.6639	1.7387	0.00427	15.9283	1.5203	1.7298×10^{-3}
AHJAYA	1.663904777	1.73865693	0.004273771	15.92829412	1.520302923	1.7298×10^{-3}

Table 22. Comparison of extracted parameters between the AHJAYA and other mature algorithms on the STP6-120/36.

Algorithm	I_{pv}(A)	I_{sd}(μA)	R_S(Ω)	R_P(Ω)	n	RMSE
CWOA (2017)[64]	7.4760	1.2	0.00000490	9.7942	1.2069	1.7601×10^{-2}
ITLBO (2019) [45]	7.4725	2.335	0.0046	22.2199	1.2601	1.6601×10^{-2}
BHCS (2019) [46]	7.47253	2.33499	0.00459	22.21990	1.26010	1.6601×10^{-2}
ATLDE (2020) [14]	7.47252992	2.33499485	0.00459463	22.21989607	1.26010347	1.6601×10^{-2}
EJAYA (2021) [49]	7.47253	2.33499	0.00459	22.21989	1.2601	1.6601×10^{-2}
IGSK (2021) [50]	7.47252992	2.335	0.004594635	22.21989406	1.260103467	1.6601×10^{-2}
RLDE (2021) [51]	7.4725	2.335	0.0046	22.2199	1.2601	1.6601×10^{-2}
AHJAYA	7.472529926	2.33499493	0.004594634	22.21989296	1.260103473	1.6601×10^{-2}

Through the above comparisons, the superior performance of the proposed AHJAYA is further proved in terms of photovoltaic model parameters. The AHJAYA is in the leading position among the mature algorithms in terms of PV model parameter extraction. It is important to note that there are still algorithms that perform well, such as the IGSK. In addition, the best RMSE of many algorithms and the corresponding extracted parameters are summarized in Table 19 to Table 22.

6. Conclusions and future work

In order to extract the parameters in the photovoltaic model more accurately and efficiently, a chaotic self-adaptive JAYA algorithm, called AHJAYA, is proposed in this paper. In the proposed AHJAYA, the self-adaptive coefficient strategy is introduced, which changes the priority of the optimal search agent and the worst search agent in the evolution strategy, and improves the exploration ability of the algorithm. Then combined with the linear population reduction strategy and chaotic opposition-based learning, the convergence speed of the algorithm is improved. On the other hand, the algorithm is prevented from falling into local optimum. Firstly, the AHJAYA is compared with several well-known algorithms, and the performance of the AHJAYA is preliminarily verified. Then, the results are further compared with the results of well-established algorithms for PV model parameter extraction. The final results show that the AHJAYA has better performance than most of the algorithms and is in the leading position.

In future work, the proposed AHJAYA will likely be used to solve higher-dimensional complex problems, and even multi-objective versions will be developed to solve practical problems.

Acknowledgments

The authors would also like to thank the supports of the following projects: The scientific research team project of Jing Chu University of technology with grant number TD202001. National Training Program of Innovation and Entrepreneurship for Undergraduates with grant number 202111336006. The key research and development project of Jing men with grant numbers 2019YFZD009. Provincial teaching reform research project of Hubei universities with grant number 2020683.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. S. Li, W. Gong, X. Yan, C. Hu, D. Bai, L. Wang, Parameter estimation of photovoltaic models with memetic adaptive differential evolution, *Sol. Energy*, **190** (2019), 465–474. https://doi.org/10.1016/j.solener.2019.08.022
2. Z. Liao, Q. Gu, S. Li, Z. Hu, B. Ning, An improved differential evolution to extract photovoltaic cell parameters, *IEEE Access*, **8** (2020), 177838–177850. http://doi.org/ 10.1109/ACCESS.2020.3024975
3. S. Li, Q. Gu, W. Gong, B. Ning, An enhanced adaptive differential evolution algorithm for parameter extraction of photovoltaic models, *Energy Convers. Manage.*, **205** (2020), 112443. https://doi.org/10.1016/j.enconman.2019.112443
4. Z. Liao, Z. Chen, S. Li, Parameters extraction of photovoltaic models using triple-phase teaching-learning-based optimization, *IEEE Access*, **8** (2020), 69937–69952. https://doi.org/10.1109/ACCESS.2020.2984728
5. H. M. Ridha, H. Hizam, C. Gomes, A. A. Heidari, H. Chen, M. Ahmadipour, et al., Parameters extraction of three diode photovoltaic models using boosted LSHADE algorithm and Newton Raphson method, *Energy*, **224** (2021), 120136. https://doi.org/10.1016/j.energy.2021.120136
6. S. Li, W. Gong, Q. Gu, A comprehensive survey on meta-heuristic algorithms for parameter extraction of photovoltaic models, *Renewable Sustainable Energy Rev.*, **141** (2021), 110828. https://doi.org/10.1016/j.rser.2021.110828
7. M. Abd Elaziz, D. Oliva, Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm, *Energy Convers. Manage.*, **171** (2018), 1843–1859. https://doi.org/10.1016/j.enconman.2018.05.062
8. J. Liang, K. Qiao, M. Yuan, K. Yu, B. Qu, S. Ge, et al., Evolutionary multi-task optimization for parameters extraction of photovoltaic models, *Energy Convers. Manage.*, **207** (2020), 112509. https://doi.org/10.1016/j.enconman.2020.112509
9. A. Askarzadeh, A. Rezazadeh, Parameter identification for solar cell models using harmony search-based algorithms, *Sol. Energy*, **86** (2012), 3241–3249. https://doi.org/10.1016/j.solener.2012.08.018
10. T. Kang, J. Yao, M. Jin, S. Yang, T. Duong, A novel improved cuckoo search algorithm for parameter estimation of photovoltaic (PV) models, *Energies*, **11** (2018), 1–31. https://doi.org/10.3390/en11051060

11. M. R. AlRashidi, M. F. AlHajri, K. M. El-Naggar, A. K. Al-Othman, A new estimation approach for determining the I–V characteristics of solar cells, *Sol. Energy*, **85** (2011), 1543–1550. https://doi.org/10.1016/j.solener.2011.04.013
12. A. Askarzadeh, A. Rezazadeh, Artificial bee swarm optimization algorithm for parameters identification of solar cell models, *Appl. Energy*, **102** (2013), 943–949. https://doi.org/10.1016/j.apenergy.2012.09.052
13. R. Ben Messaoud, Extraction of uncertain parameters of single-diode model of a photovoltaic panel using simulated annealing optimization, *Energy Rep.*, **6** (2020), 350–357. https://doi.org/10.1016/j.egyr.2020.01.016
14. S. Li, W. Gong, L. Wang, X. Yan, C. Hu, A hybrid adaptive teaching–learning-based optimization and differential evolution for parameter identification of photovoltaic models, *Energy Convers. Manage.*, **225** (2020), 113474. https://doi.org/10.1016/j.enconman.2020.113474
15. K. G. K. Harish, Modeling of solar cell under different conditions by Ant Lion Optimizer with LambertW function, *Appl. Soft Comput.*, **71** (2018), 141–151. https://doi.org/10.1016/j.asoc.2018.06.025
16. H. M. Ridha, H. Hizam, S. Mirjalili, M. L. Othman, M. E. Ya'acob, L. Abualigah, A novel theoretical and practical methodology for extracting the parameters of the single and double diode photovoltaic models, *IEEE Access*, **10** (2022), 11110–11137. https://doi.org/10.1109/ACCESS.2022.3142779
17. A. A. Al-Shamma'a, H. O. Omotoso, F. A. Alturki, H. M. H. Farh, A. Alkuhayli, K. Alsharabi, et al., Parameter estimation of photovoltaic cell/modules using bonobo optimizer, *Energies*, **15** (2022), 140. https://doi.org/10.3390/en15010140
18. W. Zhou, P. Wang, A. A. Heidari, X. Zhao, H. Turabieh, M. Mafarja, et al., Metaphor-free dynamic spherical evolution for parameter estimation of photovoltaic modules, *Energy Rep.*, **7** (2021), 5175–5202. https://doi.org/10.1016/j.egyr.2021.07.041
19. A. Farah, A. Belazi, F. Benabdallah, A. Almalaq, M. Chtourou, M. A. Abido, Parameter extraction of photovoltaic models using a comprehensive learning Rao-1 algorithm, *Energy Convers. Manage.*, **252** (2022), 115057. https://doi.org/10.1016/j.enconman.2021.115057
20. J. Luo, J. Zhou, X. Jiang, A modification of the imperialist competitive algorithm with hybrid methods for constrained optimization problems, *IEEE Access*, **9** (2021), 161745–161760. https://doi.org/10.1109/ACCESS.2021.3133579
21. M. A. E. Sattar, A. Al Sumaiti, H. Ali, A. A. Z. Diab, Marine predators algorithm for parameters estimation of photovoltaic modules considering various weather conditions, *Neural Comput. Appl.*, **33** (2021), 11799–11819. https://doi.org/10.1007/s00521-021-05822-0
22. S. Jiao, G. Chong, C. Huang, H. Hu, M. Wang, A. A. Heidari, et al., Orthogonally adapted Harris hawks optimization for parameter estimation of photovoltaic models, *Energy*, **203** (2020), 117804. https://doi.org/10.1016/j.energy.2020.117804
23. Y. Yu, K. Wang, T. Zhang, Y. Wang, C. Peng, S. Gao, A population diversity-controlled differential evolution for parameter estimation of solar photovoltaic models, *Sustainable Energy Technol. Assess.*, **51** (2022), 101938. https://doi.org/10.1016/j.seta.2021.101938
24. S. Gao, K. Wang, S. Tao, T. Jin, H. Dai, J. Cheng, A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models, *Energy Convers. Manage.*, **230** (2021), 113784. https://doi.org/10.1016/j.enconman.2020.113784

25. R. V. Rao, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, *Int. J. Ind. Eng. Comput.*, **7** (2016), 19–34. http://dx.doi.org/10.5267/j.ijiec.2015.8.004
26. Y. Zhang, Z. Jin, Comprehensive learning Jaya algorithm for engineering design optimization problems, *J. Intell. Manuf.*, **2021** (2021). https://doi.org/10.1007/s10845-020-01723-6
27. Y. Zhang, A. Chi, S. Mirjalili, Enhanced Jaya algorithm: A simple but efficient optimization method for constrained engineering design problems, *Knowl. Based Syst.,* **233** (2021), 107555. https://doi.org/10.1016/j.knosys.2021.107555
28. M. Afifi, H. Rezk, M. Ibrahim, M. El-Nemr, Multi-objective optimization of switched reluctance machine design using jaya algorithm (MO-Jaya), *Mathematics*, **9** (2021), 1107. https://doi.org/10.3390/math9101107
29. S. Basak, B. Bhattacharyya, B. Dey, Combined economic emission dispatch on dynamic systems using hybrid CSA-JAYA Algorithm, *Int. J. Syst. Assur. Eng. Manage.*, **2022** (2022). https://doi.org/10.1007/s13198-022-01635-z
30. D. Saadaoui, M. Elyaqouti, K. Assalaou, D. B. hmamou, S. Lidaighbi, Multiple learning JAYA algorithm for parameters identifying of photovoltaic models, *Mater. Today Proc.*, **52** (2022), 108–123. https://doi.org/10.1016/j.matpr.2021.11.106
31. M. F. Tefek, M. Arslan, Highway accident number estimation in Turkey with Jaya algorithm, *Neural Comput. Appl.*, **34** (2022), 5367–5381. https://doi.org/10.1007/s00521-022-06952-9
32. J. Gholami, M. R. Kamankesh, S. Mohammadi, E. Hosseinkhani, S. Abdi, Powerful enhanced Jaya algorithm for efficiently optimizing numerical and engineering problems, *Soft Comput.*, **2022** (2022). https://doi.org/10.1007/s00500-022-06909-z
33. X. Jian, Y. Cao, A chaotic second order oscillation JAYA Algorithm for parameter extraction of photovoltaic models, *Photonics*, **9** (2022). https://doi.org/10.3390/photonics9030131
34. S. Belagoune, N. Bali, K. Atif, H. Labdelaoui, A discrete chaotic Jaya algorithm for optimal preventive maintenance scheduling of power systems generators, *Appl. Soft Comput.*, **119** (2022), 108608. https://doi.org/10.1016/j.asoc.2022.108608
35. A. Aleti, I. Moser, A systematic literature review of adaptive parameter control methods for evolutionary algorithms, *Assoc. Comput. Mach.*, **49** (2017), 1–35. https://doi.org/10.1145/2996355
36. Z. Lei, S. Gao, S. Gupta, J. Cheng, G. Yang, An aggregative learning gravitational search algorithm with self-adaptive gravitational constants, *Exp. Syst. Appl.*, **152** (2020), 113396. https://doi.org/10.1016/j.eswa.2020.113396
37. R. Tanabe, A. S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in *2014 IEEE Congress on Evolutionary Computation (CEC)*, (2014), 1658–1665. https://doi.org/10.1109/CEC.2014.6900380
38. H. Yang, S. Gao, R. L. Wang, Y. Todo, A ladder spherical evolution search algorithm, *IEICE Trans. Inf. Syst.*, **104** (2021), 461–464. http://doi.org/10.1587/transinf.2020EDL8102
39. X. Yu, X. Wu, W. Luo, Parameter identification of photovoltaic models by hybrid adaptive JAYA Algorithm, *Mathematics*, **10** (2022), 183. https://doi.org/10.3390/math10020183
40. Y. J. Zhang, Y. X. Yan, J. Zhao, Z. M. Gao, AOAAO: The hybrid algorithm of arithmetic optimization algorithm with aquila optimizer, *IEEE Access*, **10** (2022), 10907–10933. https://doi.org/10.1109/ACCESS.2022.3144431

41. J. Zhao, Z.-M. Gao, The chaotic slime mould algorithm with chebyshev map, in *2nd International Conference on Artificial Intelligence and Computer Science*, **1631** (2020), 012071. https://doi.org/10.1088/1742-6596/1631/1/012071
42. K. Yu, B. Qu, C. Yue, S. Ge, X. Chen, J. Liang, A performance-guided JAYA algorithm for parameters identification of photovoltaic cell and module, *Appl. Energy*, **237** (2019), 241–257. https://doi.org/10.1016/j.apenergy.2019.01.008
43. Z. Yan, S. Li, W. Gong, An adaptive differential evolution with decomposition for photovoltaic parameter extraction, *Math. Biosci. Eng.*, **18** (2021), 7363–7388. https://doi.org/10.1016/j.apenergy.2019.01.008
44. G. Xiong, J. Zhang, X. Yuan, D. Shi, Y. He, G. Yao, Parameter extraction of solar photovoltaic models by means of a hybrid differential evolution with whale optimization algorithm, *Sol. Energy*, **176** (2018), 742–761. https://doi.org/10.1016/j.solener.2018.10.050
45. S. Li, W. Gong, X. Yan, C. Hu, D. Bai, L. Wang, et al., Parameter extraction of photovoltaic models using an improved teaching-learning-based optimization, *Energy Convers. Manage.*, **186** (2019), 293–305. https://doi.org/10.1016/j.enconman.2019.02.048
46. X. Chen, K. Yu, Hybridizing cuckoo search algorithm with biogeography-based optimization for estimating photovoltaic model parameters, *Sol. Energy*, **180** (2019), 192–206. https://doi.org/10.1016/j.solener.2019.01.025
47. L. M. P. Deotti, J. L. R. Pereira, I. C. Silva Júnior, Parameter extraction of photovoltaic models using an enhanced Lévy flight bat algorithm, *Energy Convers. Manage.*, **221** (2020), 113114. https://doi.org/10.1016/j.enconman.2020.113114
48. G. Xiong, J. Zhang, D. Shi, L. Zhu, X. Yuan, Parameter extraction of solar photovoltaic models with an either-or teaching learning based algorithm, *Energy Convers. Manage.*, **224** (2020), 113395. https://doi.org/10.1016/j.enconman.2020.113395
49. X. Yang, W. Gong, Opposition-based JAYA with population reduction for parameter estimation of photovoltaic solar cells and modules, *Appl. Soft Comput.*, **104** (2021), 107218. https://doi.org/10.1016/j.asoc.2021.107218
50. K. M. Sallam, M. A. Hossain, R. K. Chakrabortty, M. J. Ryan, An improved gaining-sharing knowledge algorithm for parameter extraction of photovoltaic models, *Energy Convers. Manage.*, **237** (2021), 114030. https://doi.org/10.1016/j.enconman.2021.114030
51. Z. Hu, W. Gong, S. Li, Reinforcement learning-based differential evolution for parameters extraction of photovoltaic models, *Energy Rep.*, **7** (2021), 916–928. https://doi.org/10.1016/j.egyr.2021.01.096
52. K. Yu, J. J. Liang, B. Y. Qu, Z. Cheng, H. Wang, Multiple learning backtracking search algorithm for estimating parameters of photovoltaic models, *Appl. Energy*, **226** (2018), 408–422. https://doi.org/10.1016/j.apenergy.2018.06.010
53. N. Pourmousa, S. M. Ebrahimi, M. Malekzadeh, M. Alizadeh, Parameter estimation of photovoltaic cells using improved Lozi map based chaotic optimization algorithm, *Sol. Energy*, **180** (2019), 180–191. https://doi.org/10.1016/j.solener.2019.01.026
54. W. Long, T. Wu, M. Xu, M. Tang, S. Cai, Parameters identification of photovoltaic models by using an enhanced adaptive butterfly optimization algorithm, *Energy*, **229** (2021), 120750. https://doi.org/10.1016/j.energy.2021.120750

55. Y. Liu, A. A. Heidari, X. Ye, C. Chi, X. Zhao, C. Ma, et al., Evolutionary shuffled frog leaping with memory pool for parameter optimization, *Energy Rep.*, **7** (2021), 584–606. https://doi.org/10.1016/j.egyr.2021.01.001
56. X. Chen, K. Yu, W. Du, W. Zhao, G. Liu, Parameters identification of solar cell models using generalized oppositional teaching learning based optimization, *Energy*, **99** (2016), 170–180. https://doi.org/10.1016/j.energy.2016.01.052
57. K. Yu, X. Chen, X. Wang, Z. Wang, Parameters identification of photovoltaic models using self-adaptive teaching-learning-based optimization, *Energy Convers. Manage.*, **145** (2017), 233–246. https://doi.org/10.1016/j.enconman.2017.04.054
58. K. Yu, J. J. Liang, B. Y. Qu, X. Chen, H. Wang, Parameters identification of photovoltaic models using an improved JAYA optimization algorithm, *Energy Convers. Manage.*, **150** (2017), 742–753. https://doi.org/10.1016/j.enconman.2017.08.063
59. X. Chen, B. Xu, C. Mei, Y. Ding, K. Li, Teaching–learning–based artificial bee colony for solar photovoltaic parameter estimation, *Appl. Energy*, **212** (2018), 1578–1588. https://doi.org/10.1016/j.apenergy.2017.12.115
60. S. M. Ebrahimi, E. Salahshour, M. Malekzadeh, F. Gordillo, Parameters identification of PV solar cells and modules using flexible particle swarm optimization algorithm, *Energy*, **179** (2019), 358–372. https://doi.org/10.1016/j.energy.2019.04.218
61. Y. Zhang, C. Huang, Z. Jin, Backtracking search algorithm with reusing differential vectors for parameter identification of photovoltaic models, *Energy Convers. Manage.*, **223** (2020), 113266. https://doi.org/10.1016/j.enconman.2020.113266
62. Y. Zhang, M. Ma, Z. Jin, Comprehensive learning Jaya algorithm for parameter extraction of photovoltaic models, *Energy*, **211** (2020), 118644. https://doi.org/10.1016/j.energy.2020.118644
63. Y. Zhang, M. Ma, Z. Jin, Backtracking search algorithm with competitive learning for identification of unknown parameters of photovoltaic systems, *Expert Syst. Appl.*, **160** (2020), 113750. https://doi.org/10.1016/j.eswa.2020.113750
64. D. Oliva, M. Abd El Aziz, A. Ella Hassanien, Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm, *Appl. Energy*, **200** (2017), 141–154. https://doi.org/10.1016/j.apenergy.2017.05.029
65. P. Lin, S. Cheng, W. Yeh, Z. Chen, L. Wu, Parameters extraction of solar cell models using a modified simplified swarm optimization algorithm, *Sol. Energy*, **144** (2017), 594–603. https://doi.org/10.1016/j.solener.2017.01.064
66. G. Xiong, J. Zhang, D. Shi, Y. He, Parameter extraction of solar photovoltaic models using an improved whale optimization algorithm, *Energy Convers. Manage.*, **174** (2018), 388–405. https://doi.org/10.1016/j.enconman.2018.08.053
67. A. M. Beigi, A. Maroosi, Parameter identification for solar cells and module using a Hybrid Firefly and Pattern Search Algorithms, *Sol. Energy*, **171** (2018), 435–446. https://doi.org/10.1016/j.solener.2018.06.092
68. J. Liang, S. Ge, B. Qu, K. Yu, F. Liu, H. Yang, et al., Classified perturbation mutation based particle swarm optimization algorithm for parameters extraction of photovoltaic models, *Energy Convers. Manage.*, **203** (2020), 112138. https://doi.org/10.1016/j.enconman.2019.112138

69. X. Lin, Y. Wu, Parameters identification of photovoltaic models using niche-based particle swarm optimization in parallel computing architecture, *Energy*, **196** (2020), 117054. https://doi.org/10.1016/j.energy.2020.117054