

On Solar Photovoltaic Parameter Estimation: Global Optimality Analysis and a Simple Efficient Differential Evolution Method

Shuhua Gao, Cheng Xiang*, Ming Yu, Kuan Tak Tan, Tong Heng Lee

Abstract—The literature presents a vast array of advanced metaheuristic methods for photovoltaic parameter estimation. However, the focus of this study is not to introduce another new method into this already crowded field. Instead, we examine two important but often overlooked questions: (i) are existing results globally optimal, and (ii) can a simpler method achieve comparable performance. We conduct case studies using two widely used I-V curve datasets. To address the first issue, we develop a branch and bound algorithm that, despite its sluggishness, either certifies the global minimum or provides a tight upper bound. These values serve as useful benchmarks for fair metaheuristic evaluation and further development. To answer the second question, our extensive examination and comparison surprisingly reveal that a basic differential evolution (DE) algorithm can achieve the certified global minimum or obtain the best known result. Additionally, the DE algorithm's runtime is much shorter than that of current state-of-the-art metaheuristic methods, making it a great choice for time-sensitive applications. This remarkable finding suggests that using increasingly complex metaheuristics in ordinary PV parameter estimation problems might be unnecessary. Finally, we discuss the implications of these outcomes and propose the simple DE method as the premier choice for industrial applications.

I. INTRODUCTION

Accurate modeling of solar photovoltaic (PV) systems is necessary for their effective design, simulation, power forecasting, and optimal control [1]–[3]. The dominating method to describe solar PV systems uses an analogous electrical circuit model [4], which has been further specialized to the single-diode model (SDM), the double-diode model (DDM). Despite the intuitiveness of these circuit models, the main difficulty lies in the accurate determination of unknown parameters in the model [3]–[6].

PV parameter estimation is commonly formulated as a nonlinear optimization problem from the perspective of I-V (current-voltage) curve fitting. The problem has been widely attempted with various metaheuristic algorithms. Most metaheuristics are population-based by exploiting a swarm of

interacting agents to search the solution space efficiently [3]. Since metaheuristic algorithms are mostly not problem-specific, any metaheuristic optimizer may be applied to PV parameter estimation in principle. It is unsurprising that a large number of metaheuristic methods have been proposed for PV parameter estimation. Some recent examples include guaranteed convergence particle swarm optimization [7], improved JAYA optimization [8], performance-guided JAYA [9], self-adaptive ensemble-based differential evolution [10], teaching-learning-based optimization [2], [11], grey wolf optimizer and cuckoo search based hybrid method [12], among many others. We omit their technical details due to space limit. Interested readers may refer to [1], [3], [6].

Despite the increasing interest in such metaheuristics, none of them can guarantee or identify the discovery of the global optimum [5]. Moreover, the minimal root mean square error (RMSE) of curve fitting attained by different metaheuristics has suffered from stagnation with no further reduction in recent studies (see [2, Table 3], [9, Table 3], and Table VI). Thus, one may wonder naturally whether the best-known result is already the global minimum such that we can avoid futile efforts by building more advanced metaheuristics blindly. Also, since a variety of metaheuristics can get the same RMSE value, another natural query is how sophisticated a metaheuristic has to be to achieve effective PV parameter estimation. An industrial practitioner desires certainly an effective yet simple and fast algorithm. In particular, the algorithm's efficiency is critical for time-sensitive applications, for example, the real-time monitoring of solar cell degradation via photovoltaic curves telemetry using a microprocessor on a satellite [13].

In this study, we attempt to answer the above two questions through extensive investigations using the two most broadly studied benchmark datasets (mainly to facilitate comparison) [14]. The main contributions of this paper are listed below.

- The global minimum RMSE of the SDM has been certified on both datasets for the first time using an interval arithmetic based branch and bound method. Besides, a useful upper bound of the global minimum for the DDM is obtained. These values serve as valuable references for assessing and developing metaheuristics.
- We show that an *intentionally* simple differential evolution (DE) algorithm is adequate to attain the global minimum for the SDM and achieve equally high accuracy for the DDM compared with a variety of sophisticated metaheuristics. Moreover, the DE algorithm excels in performance stability and incomparable time efficiency owing to its simplicity and efficacy, making

Shuhua Gao is with School of Control Science and Engineering, Shandong University, Jinan 250100, China shuhua.gao@sdu.edu.cn

Cheng Xiang (*corresponding author*) and Tong Heng Lee are with Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 elexc@nus.edu.sg

Ming Yu is with Power Automation Pte Ltd, 438B Alexandra Road, Alexandra TechnoPark, 119968 Singapore

Kuan Tak Tan is with the Engineering Cluster, Singapore Institute of Technology, 10 Dover Drive, 138683 Singapore

This work was supported partially by National Natural Science Foundation of China under Grant 62303277, Young Experts of Taishan Scholar Project (NO. tsqn202306067) and the Natural Science Fund of Shandong Province under Grant 2023HWYQ-003 and ZR2023QF048 and partially by the Energy Market Authority of Singapore through the EDGE Programme under LA/Contract No: EDGE-GC2018-002.

it particularly suitable for real-time applications.

- Based on our discoveries and comprehensive comparisons with state-of-the-art methods, we recommend the simple DE algorithm to solar industry engineers as the first choice in practical applications, particularly time-sensitive ones. Furthermore, we provide useful suggestions for researchers to reconsider perspectives on PV parameter estimation and to avoid designing overly complex metaheuristic methods.

It should be noted that the purpose of this paper, as previously stated, is not to develop another new metaheuristic method for PV parameter estimation. The simple differential evolution method focused on in this study is an established technique and certainly not new, but we choose it purposefully to highlight the danger of over-engineering that is often present in current metaheuristic approaches. Rather than introducing another such method, this study aims to rigorously establish the limit of estimation accuracy achievable through metaheuristic techniques. Additionally, we aim to revive the use of the classic differential evolution algorithm for PV parameter estimation, which has been unfortunately overlooked in current literature.

II. SYSTEM MODELING AND PROBLEM FORMULATION

A. Modeling of PV systems

The electrical circuit corresponding to the SDM is shown in Fig. 1(a). Specifically, the circuit contains a current source I_{ph} , which refers to the photocurrent generated by the PV cell, a diode flowing current I_d , and two resistors with resistance R_p and R_s , respectively. We can calculate the diode current I_d using the Shockley equation as follows,

$$I_d = I_0 \left[\exp \left(\frac{q(V + IR_s)}{nkT} \right) - 1 \right], \quad (1)$$

where I_0 is the reverse saturation current of the diode, n is the diode ideal factor, T is the temperature in Kelvin, and V is the output voltage of the cell. The others are just physical constants: the electron charge $q = 1.60217646 \times 10^{-19}$ C and the Boltzmann constant $k = 1.3806503 \times 10^{-23}$ J/K.

The output current I is computed using first principles by

$$I = I_{ph} - I_0 \left[\exp \left(\frac{q(V + IR_s)}{nkT} \right) - 1 \right] - \frac{V + IR_s}{R_p}. \quad (2)$$

There are five unknown parameters in (2), which are collected into a parameter vector $\theta_S = [I_{ph}, I_0, n, R_s, R_p]$.

Despite the simplicity and usefulness of the above SDM, it does not consider the effect of recombination current loss in the depletion region [1], [6]. An additional diode can be introduced into the circuit to compensate for this specific loss to attain higher accuracy. The equivalent circuit of the DDM is illustrated in Fig. 1(b). In analogy to the SDM (2),

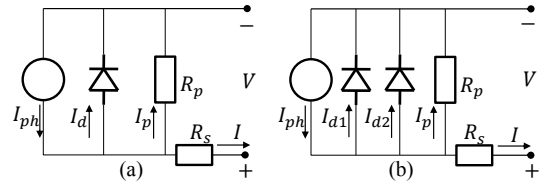


Fig. 1: Equivalent circuit of a PV cell. (a) SDM; (b) DDM.

the DDM is derived straightforwardly as follows:

$$\begin{aligned} I &= I_{ph} - I_{d1} - I_{d2} - I_p \\ &= I_{ph} - I_{01} \left[\exp \left(\frac{q(V + IR_s)}{n_1 kT} \right) - 1 \right] - \\ &\quad I_{02} \left[\exp \left(\frac{q(V + IR_s)}{n_2 kT} \right) - 1 \right] - \frac{V + IR_s}{R_p}, \quad (3) \end{aligned}$$

where I_{01} and I_{02} are the reverse saturation current of the two diodes, and n_1 and n_2 denote the ideality factor of the two diodes, respectively. The DDM has seven parameters in total, denoted by $\theta_D = [I_{ph}, I_{01}, I_{02}, n_1, n_2, R_s, R_p]$.

A PV module contains multiple PV cells connected in series or parallel. It is standard to assume the same parameter values for all cells for computational tractability purposes. Then, all cells are lumped into a single, functionally equivalent cell [9], [14], [15]. Thus, the same procedure is used to fit either the SDM (2) or the DDM (3) to the I-V curve of a PV module.

B. Optimization problem formulation

The fundamental principle of parameter estimation via I-V curve fitting is to find appropriate parameter values such that the current values calculated with either the SDM (2) or the DDM (3) match the measurement values for a set of data points [1]. Without loss of generality, we discuss below the problem formulation using the SDM (2), termed f_S hereafter, with parameters θ_S , whose principle can be transplanted to the DDM case seamlessly.

Note that we cannot write down a simple closed-form solution $I = f_S^{-1}(V)$ for the model f_S (2) to compute I given V . As a workaround, given a measurement (V^m, I^m) and a tentative parameter vector θ_S , the majority of metaheuristic-based studies compute the *predicted* current with the model approximately but computationally economically as

$$I \approx f_S(V^m, I^m; \theta_S), \quad (4)$$

and try to reduce the deviation between I and I^m by adjusting θ_S (see [2], [9], and [15] among others).

The root mean square error (RMSE) is widely used to quantify the difference between computed current values and the measurement values [10], [13]. Supposing there are N data points in the I-V curve, we get the following constrained optimization problem, which is known widely as *nonlinear least-squares regression* in the literature.

$$\text{minimize} \quad J(\theta) = \sum_{i=1}^N (f(V_i^m, I_i^m; \theta) - I_i^m)^2, \quad (5a)$$

$$\text{subject to} \quad \theta \in \Theta. \quad (5b)$$

Algorithm 1 Interval branch and bound optimization

Input: objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and bound constraints $X \subset \mathbb{R}^n$, precision parameters ϵ_f and ϵ_x

Output: lower and upper bounds of the global minimum value $[\underline{f}, \bar{f}]$, a list of boxes L_S that contain all possible global minimizers

```
1: initialize a list  $L \leftarrow \{[\mathbf{x}]\}$  with  $[\mathbf{x}]$  corresponding to  $X$ 
2: initialize an empty candidate solution list  $L_S$ 
3:  $\bar{f} \leftarrow \infty$  ▷ Upper bound of  $f^*$ 
4: while  $L \neq \emptyset$  do
5:   choose  $[\mathbf{x}] \in L$  and remove  $[\mathbf{x}]$  from  $L$ 
6:   contract  $[\mathbf{x}]$ 
7:   evaluate  $f$  at the center of  $[\mathbf{x}]$  and get value  $f_c$ 
8:   Update  $\bar{f}$  by  $\bar{f} \leftarrow \min\{\bar{f}, f_c\}$ 
9:   if  $[\mathbf{x}]$  satisfies criteria (6) then
10:    append  $[\mathbf{x}]$  to  $L_S$ 
11:   else
12:     split  $[\mathbf{x}]$  into subboxes and add them to  $L$ 
13:   end if
14: end while
15: remove any box  $[\mathbf{x}] \in L_S$  from  $L_S$  with  $f([\mathbf{x}]) > \bar{f}$ 
16:  $\underline{f} \leftarrow \min_{[\mathbf{x}] \in L_S} f([\mathbf{x}])$  ▷ Lower bound of  $f^*$ 
```

where f refers to either the SDM f_S (2) or the DDM f_D (3), and θ is the corresponding parameter vector θ_S or θ_D . (V_i^m, I_i^m) is the i -th data point in measurement. Θ denotes the specified bound constraints of θ that take physical reality into consideration (see Table I for examples).

III. OPTIMIZATION METHODS

In this section, we apply a branch and bound (B&B) based *deterministic* global optimization technique for rigorous certification of the solution optimality. Then, we choose deliberately a simple *stochastic* optimization algorithm, *differential evolution* (DE), to compete the increasingly sophisticated metaheuristic methods prevalent in the literature.

A. Deterministic global optimization with an interval arithmetic based branch and bound algorithm

The fundamental task of deterministic global optimization (DGO) is to determine *rigorously* (i.e., with theoretical guarantees) the global minimum of an objective function f subject to a set of constraints [16]. However, finding the global minimum for a general nonconvex optimization problem like (5) is NP-hard [16]. We are practically more interested in identifying a solution sufficiently close to the true global minimum, called the ϵ -global minimum [16].

The most popular algorithmic framework of DGO is the branch and bound (B&B) method [16]. The search space is divided recursively into smaller subspaces and forms accordingly a tree structure of subproblems. The consequential pruning of search space is performed by eliminating subproblems whose lower bound is no better than the best upper bound found so far. Interval analysis is a handy tool to estimate the lower and upper bounds of regions/branches of the search space [17].

The general B&B framework with interval arithmetic is depicted in Algorithm 1, where $[\mathbf{x}]$ denotes an n -dimensional

interval vector (also known as an *interval box*, whose components are intervals). Applying a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ to $[\mathbf{x}]$ yields another interval termed $f([\mathbf{x}]) = [\underline{f}([\mathbf{x}]), \bar{f}([\mathbf{x}])]$, where $\underline{f}([\mathbf{x}])$ and $\bar{f}([\mathbf{x}])$ denote the lower and upper bounds respectively and are calculated rigorously with interval analysis [17]. Each iteration is composed of three main components: box selection (Line 5), box contracting (Line 6), and box splitting (Line 12). In particular, the purpose of contracting is to delete subboxes inside $[\mathbf{x}]$ that cannot contain a globally optimal solution to reduce search space [17, Chapter 12]. In order to be included in L_S , a box $[\mathbf{x}]$ must satisfy two conditions that are checked in Line 9:

$$\text{width}([\mathbf{x}]) \leq \epsilon_x, \quad \text{width}(f([\mathbf{x}])) \leq \epsilon_f, \quad (6)$$

where $\text{width}(\cdot)$ denotes the *width* of a box defined by its largest diameter [16]. ϵ_x and ϵ_f are two tolerance parameters provided by the user, often known as the *precision*. After the main loop finishes, we post-process the solution list L_S in Line 15 to discard boxes which cannot contain the global minimum \mathbf{x}^* according to the latest knowledge of \bar{f} .

At the end of Algorithm 1, we get the (usually very tight) bounds of the global minimum $f^* \in [\underline{f}, \bar{f}]$. It is guaranteed that $\bar{f}([\mathbf{x}]) - f^* \leq 2\epsilon_f, \forall [\mathbf{x}] \in L_S$ [17]. Any \mathbf{x} inside the remaining boxes L_S becomes an ϵ -global minimum with $\epsilon = 2\epsilon_f$ in this case [16]. Though the *exact* global minimum \mathbf{x}^* and f^* are still unknown and remain computationally intractable, a reasonably tight bound by setting small ϵ_f and ϵ_x in Algorithm 1 is usually enough for practical purposes. Note that Algorithm 1 only sketches out the basic skeleton of interval B&B algorithms. We resort to a dedicated interval analysis library *ibexopt* (<http://www.ibex-lib.org> (v2.8)) in actual implementation.

B. Stochastic global optimization with a simple DE

Though an interval B&B algorithm can ascertain the global optimum rigorously in theory, it is generally much more computationally expensive than metaheuristic algorithms, rendering itself impractical in industrial applications [5]. We will report its intolerably long running time in Section IV-B. In many engineering applications, it is either unnecessary or computationally intractable to obtain the *exact* global minimum [16], and metaheuristic methods are particularly useful in these scenarios. In view of the abundance of metaheuristics (recall Section I), we take an elementary differential evolution (DE) algorithm *on purpose* to investigate whether highly complicated metaheuristics are really necessary for PV parameter estimation. Again, we would like to emphasize that our objective is not to develop yet another new metaheuristic but to investigate whether a fundamental one is empirically sufficient for practical PV parameter estimation tasks.

Each iteration of DE comprises three steps: selection, crossover, and mutation. The distinguishing feature of DE is its mutation with *difference* vectors [18]. To minimize a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with bound constraints $X \subset \mathbb{R}^n$, we outline the simple DE in Algorithm 2, whose main body includes only five lines in agreement with its simplicity. The

Algorithm 2 Simple differential evolution

Input: objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and bound constraints $X \subset \mathbb{R}^n$, control parameters N_p, C_r, F, G

Output: the best vector \hat{x} and the function value \hat{f}

```

1: generate an initial population  $P^0 \leftarrow \{\mathbf{x}_i^0\}_{i=1}^{N_p}$  with (7)
2: for  $g$  from 0 to  $G - 1$  do
3:   for each vector  $\mathbf{x}_i^g \in P^g$  do
4:     generate a donor vector  $\mathbf{v}_i^g$  by (8)            $\triangleright$  mutation
5:      $\mathbf{v}_i^g \leftarrow$  bounce-back( $\mathbf{v}_i^g$ ) by (9)
6:      $\mathbf{u}_i^g \leftarrow$  crossover( $\mathbf{v}_i^g, \mathbf{x}_i^g$ ) by (10)
7:      $\mathbf{x}_i^{g+1} \leftarrow$  select( $\mathbf{u}_i^g, \mathbf{x}_i^g$ ) by (11)
8:     insert  $\mathbf{x}_i^{g+1}$  into the new population  $P^{g+1}$ 
9:   end for
10: end for
11:  $\hat{x} \leftarrow$  the best vector in  $P^G$  and  $\hat{f} \leftarrow f(\hat{x})$ 

```

initial population P comprises N_p vectors, and each initial vector $\mathbf{x}_i^0, i \in [1, N_p]$ is generated randomly by

$$x_{i,j}^0 = \underline{b}_j + \text{rand}(0, 1) \cdot (\bar{b}_j - \underline{b}_j), \quad (7)$$

where $x_{i,j}^0$ denotes the j -th component of \mathbf{x}_i^0 , \underline{b}_j and \bar{b}_j represent the lower and upper bound of the j -th variable respectively, $j \in [1, n]$. Besides, $\text{rand}(0, 1)$ generates a random number between 0 and 1.

Several mutation strategies have been developed for DE [18]. Here we adopt the most commonly used one called the “DE/rand/1” scheme. For each vector in the g -th iteration, a donor vector $\mathbf{v}_i^g, i \in [1, N_p]$, is produced by

$$\mathbf{v}_i^g = \mathbf{x}_a^g + F(\mathbf{x}_b^g - \mathbf{x}_c^g), \quad a \neq b \neq c \neq i, \quad (8)$$

where $a, b, c \in [1, N_p]$ are randomly chosen, and F is the scaling factor typically in the range $[0.4, 1]$ [18].

Note that the donor vector \mathbf{v}_i^g in (8) may lie outside the bounded region X . We adapt a simple *bounce-back* strategy [18] to handle bound constraints in Line 5, which relocates each infeasible component between the bound it violates and the corresponding value of the target vector \mathbf{x}_i^g :

$$v_{i,j}^g \leftarrow \begin{cases} \underline{b}_j + \text{rand}(0, 1) \cdot (x_{i,j}^g - \underline{b}_j) & \text{if } v_{i,j}^g < \underline{b}_j \\ \bar{b}_j - \text{rand}(0, 1) \cdot (\bar{b}_j - x_{i,j}^g) & \text{if } v_{i,j}^g > \bar{b}_j \end{cases}. \quad (9)$$

In DE, the donor vector \mathbf{v}_i^g and the target vector \mathbf{x}_i^g mate to produce a new vector \mathbf{u}_i^g named the *trial* vector. The binomial crossover scheme is widely used as follows:

$$u_{i,j}^g = \begin{cases} v_{i,j}^g & \text{if } \text{rand}(0, 1) \leq C_r \text{ or } j = \beta \\ x_{i,j}^g & \text{otherwise} \end{cases} \quad (10)$$

where $\beta \in [1, n]$ is a random integer that is generated anew for each i , and C_r is the user provided crossover rate [18]. Eq. (10) says each entry of \mathbf{u}_i^g comes from either \mathbf{v}_i^g or \mathbf{x}_i^g .

Finally, DE imposes *elitism* by selecting the better one between the target vector \mathbf{x}_i^g and the trial vector \mathbf{u}_i^g as the i -th vector into the next generation according to their fitness:

$$\mathbf{x}_i^{g+1} = \begin{cases} \mathbf{u}_i^g & \text{if } f(\mathbf{u}_i^g) \leq f(\mathbf{x}_i^g) \\ \mathbf{x}_i^g & \text{otherwise} \end{cases}. \quad (11)$$

TABLE I: Parameter search range in numerical experiments.

Parameter	RT		PW	
	Lower	Upper	Lower	Upper
I_{ph} (A)	0	1	0	2
I_0, I_{01}, I_{02} (μA)	0	1	0	50
n, n_1, n_2	1	2	1	50
R_s (Ω)	0	0.5	0	2
R_p (Ω)	0	100	0	2000

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Datasets and experimental settings

The two PV I-V datasets [14] shown in Fig. 2 serve as the *de facto* standard in evaluating algorithms’ performance (e.g., [3], [13]). The first dataset named “RT” contains 26 data points for an RTC France solar cell (1000 W/m², 33 °C). The second dataset “PW” with 25 data points refers to a Photowatt-PWP201 solar module (1000 W/m², 45 °C). Combining the two datasets and two models, we consider four cases in total. The naming rule is “*model+dataset*” for simplicity, e.g., case “SDM+RT” fits the SDM to the RT dataset. The parameter search range (i.e., Θ in (5)) in the literature, as listed in Table I, is adopted for fair comparisons.

All metaheuristic algorithms are implemented in MATLAB R2020a for a fair comparison of runtime. The results presented below were obtained on a laptop with a 1.8 GHz Core i7-8550U CPU, 8 GB RAM, and Windows 10.

B. Global optimality analysis via interval B&B

Despite the large number of metaheuristics for PV parameter estimation (see, e.g., [19, Table 1]), none of them can certify the finding of the global minimum. This section presents results regarding global optimality.

1) *SDM results*: Note that a B&B algorithm is generally computationally intensive. Following [5], we limited the runtime of `ibexopt` to 20000 seconds for “SDM+RT”. Unlike [5], we set the absolute and relative precision to smaller values (1E-13 and 1E-9) in order to obtain tighter bounds of the global minimum (that is, $[f, \bar{f}]$ in Algorithm 1). The optimization results for “SDM+RT” are reported in the first column of Table II. In particular, the bounds of the RMSE enjoy a negligible gap. Results reported in existing studies are mostly truncated to five significant digits. Hence, we can *certify* safely, for the first time, that 9.8602E-4 is indeed the global minimum RMSE value of “SDM+RT”. The parameter values in Table II yield the upper bound of RMSE, and those values match closely to results acquired with various metaheuristics (like [19, Table 1]).

We next examine the “SDM+PW” case. The major difference is the considerably widened parameter search range for PW in Table I, which may pose a big challenge to the interval B&B algorithm and require an even longer runtime. In contrast to [5], we increased the timeout to 40000 s. The best-known RMSE (see, e.g., [12, Table 13]) is 2.4250E-3, which shows exact agreement with the RMSE bounds reported in Table II. Besides, as expected, the parameter values reported therein are also extremely close to those in Table II. Again,

TABLE II: Results for SDM using interval B&B.

Variable	RT	PW
I_{ph} (A)	0.760779120136	1.03052020484
I_0 (μ A)	0.322873926858	3.48287904343
n	1.48113747635	48.6435574734
R_s (Ω)	0.0363792207867	1.20123680201
R_p (Ω)	53.7009537057	981.263690780
RMSE	[9.860250397955652E-4, 9.860250417458982E-4]	[2.425076598320144E-3, 2.425076599532477E-3]
Gap	1.950333050615427E-12	1.2123329007351913E-12
Time (s)	13547	38924

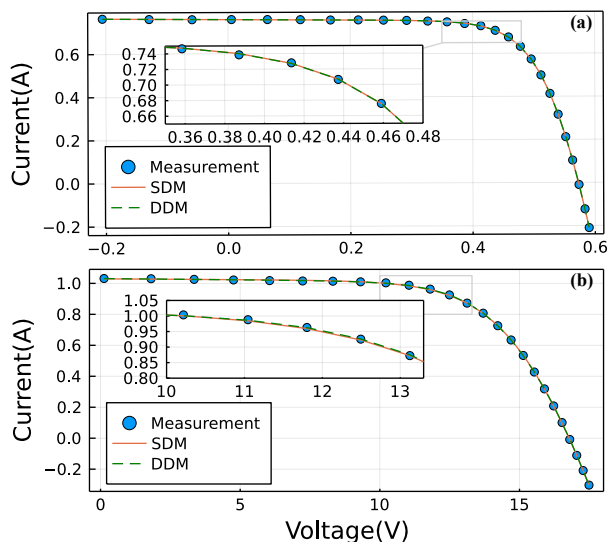


Fig. 2: Measured and estimated I-V curves using parameter values optimized by interval B&B. (a) RT; (b) PW.

such a consensus indicates the correctness of each other. The measured and the reconstructed I-V curves using SDM and parameters in Table II are shown in Fig. 2 with admirable fitting accuracy. Note that the negative current and voltage values therein simply imply a reverse direction [14].

2) *DDM results*: The DDM is more challenging due to its two extra parameters. We decided to allow `ibexopt` more time (24 hours) to get hopefully tighter optimality bounds. The overall workflow is identical to the SDM case above. The results are reported in Table III. Unfortunately, the lower bound remains zero, even if we run `ibexopt` for another 5 hours. This failure is probably caused by the notorious *cluster effect* in B&B methods [16].

Despite the zero lower bound, the revealed upper bound of the RMSE is still informative since it is very close to the best-known result, e.g., 9.8248E-4 for the “DDM+RT” case (see [9, Table 3]). The upper bound 9.8358E-4 in Table III implies that 9.8248E-4 is likely to be the global minimum though there is no theoretical guarantee. We did not find results reported for “DDM+PW” that are compatible with our analysis here. Nonetheless, the result of “DDM+PW” in Table III looks reasonable by comparing with the SDM counterpart in Table II: the RMSE upper bound of DDM is even smaller than the lower bound of SDM since, as expected, the DDM can better fit the data because of its two additional parameters [10]. Nonetheless, since the RMSE values of both

TABLE III: Results for DDM using interval B&B.

Variable	RT	PW
I_{ph} (A)	0.760815738919	1.0339286971
I_{01} (μ A)	0.217867184041	1.86575472010E-23
I_{02} (μ A)	0.781454995330	0.535399234849
n_1	1.44827388213	9.58860778809
n_2	1.98183166760	42.6724488388
R_s (Ω)	0.0367359827333	1.63619822583
R_p (Ω)	55.8931982861	607.690281231
RMSE	[0, 9.83581875679E-4]	[0, 1.61865668151E-3]
Gap	9.83581875679E-4	1.61865668151E-3
Time (s)	86400	43200

TABLE IV: Results of DE in a typical run.

	SDM		DDM	
	RT	PW	RT	PW
I_{ph} (A)	0.760775	1.03051	0.760781	1.03051
I_0/I_{01} (μ A)	0.323021	3.48226	0.225974	9.8113E-3
I_{02} (μ A)	—	—	0.749344	3.47245
n/n_1	1.481184	48.6428	1.45101	48.64282
n_2	—	—	1.99999	48.64283
R_s (Ω)	0.036377	1.20127	0.0367404	1.20127
R_p (Ω)	53.71852	981.982	55.4854	981.982
RMSE	9.8602E-4	2.4250E-3	9.8248E-4	2.4250E-3

models are pretty small, it is hard to differentiate visually the two estimated I-V curves, which are highlighted in Fig. 2.

C. Optimization via simple differential evolution (DE)

The interval B&B algorithm is not suitable for regular PV parameter estimation due to its excessively long execution time. By contrast, various metaheuristic methods can obtain a reasonably good solution in a far shorter time. However, an interesting, practically important, yet rarely studied problem is whether normal PV parameter estimation really demands the increasingly complicated metaheuristics prevailing in the recent literature. In this section, we try to get some empirical insights by examining whether the *intentionally* simple DE in Algorithm 2 can achieve comparable performance. Starting with the canonical values recommended in [18, Section III], we quickly determined appropriate control parameter values for the simple DE as $N_p = 50$, $C_r = 0.6$, $F = 0.9$, and $G = 800$ (for SDM) or 2000 (for DDM)

Since DE is a stochastic algorithm, we conventionally (e.g., [9], [15]) execute DE 30 times for each case and report the statistic characteristics. For illustration purposes, we list the DE results in a typical run in Table IV, whose estimated I-V curves are visually almost identical to Fig. 2 (since all RMSE values are likewise tiny) and thus omitted here. The convergence curves of this simple DE for RT using both models in a typical run are shown in Fig. 3. The DE usually took far fewer generations to converge than the *conservative* value G we specified. The convergence curves of DE with the PW dataset share a similar character and is omitted here but presented in the accompanying online materials.

The statistics of RMSE in 30 runs are reported in Table V. Overall, the performance of our DE algorithm is remarkably stable despite its inherent stochasticity. When applying to the SDM on both datasets and to “DDM+PW”, the DE algorithm always yields the same minimal RMSE in all 30 trials. Even

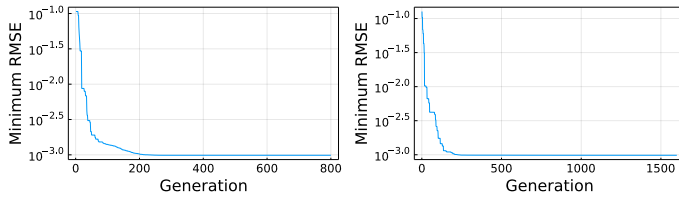


Fig. 3: Convergence curves of DE with RT: (a) SDM (b) DDM.

TABLE V: Statistics of RMSE values by DE in 30 runs.

	SDM		DDM	
	RT	PW	RT	PW
Min	9.8602E-4	2.4250E-3	9.8248E-4	2.4250E-3
Mean	9.8602E-4	2.4250E-3	9.8267E-4	2.4250E-3
Max	9.8602E-4	2.4250E-3	9.8602E-4	2.4250E-3
Std	4.3929E-17	2.9525E-17	7.1027E-7	2.3955E-17

in the worst “DDM+RT” case, the gap between the maximum and minimum RMSE values in 30 runs is still minor, as implied by the slight standard deviation in Table V.

Table V tells that the simple DE managed to find the global minimum RMSE, as identified by interval B&B, for both SDM cases. As for the more challenging DDM cases, DE achieved the best-known result for “DDM+RT”, i.e., 9.8248E-4. However, the RMSE attained by DE for “DDM+PW” is still above the identified upper bound (2.4250E-3 vs 1.6186E-3 in Table III). The main reason is presumably attributed to the extraordinarily small value of I_{01} in the potential optimal solution (around 1.86E-23 in Table III), which poses a overwhelming challenge to DE or any metaheuristic method in general (see Table VI below).

D. Comparison with existing algorithms

We compare the performance of the simple DE (Algorithm 2) with more sophisticated metaheuristics, inspecting both accuracy and efficiency. We select state-of-the-art algorithms of distinct methodology and pick particularly DE variants for a comprehensive comparison. The existing results have been listed as they appear in four latest articles: [2, Table 12], [9, Table 3], [15, Table 9], and [10, Table 9]. All results are listed in Table VI. Since the hardest case “DDM+PW” were not considered in these papers, we run the original source code of two recent studies [9], [10] (see <https://github.com/cilabzzu>) and report their results for fair comparisons. The statistics of the RMSE values in 30 runs are listed in Table VI.

To examine the statistical significance of performance difference between the simple DE and the other methods, we perform the Mann-Whitney U test [10] and report results in the “U test” column of Table VI. The null hypothesis H_0 indicates equally good performance, and the level of significance is 0.05. In the results reported in Table VI, the symbol “+” indicates a statistically significant performance difference, i.e., rejecting the null hypothesis, while “-” means

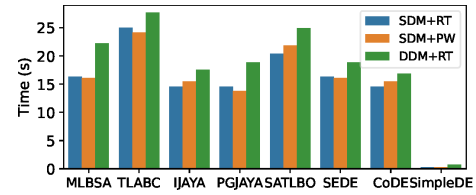


Fig. 4: Runtime comparison of different algorithms.

TABLE VI: Comparison of statistical results of various algorithms in four cases. From top to bottom: “SDM+RT”, “SDM+PW”, “DDM+RT”, and “DDM+PW”.

Method	RMSE				U test
	Min	Mean	Max	Std	
MLBSA [20]	9.8602E-4	9.8602E-4	9.8602E-4	7.0800E-11	+
TLABC [11]	9.8602E-4	9.9417E-4	1.0308E-3	1.1896E-5	+
IJAYA [8]	9.8602E-4	9.8605E-4	9.8684E-4	1.4931E-7	+
PGJAYA [9]	9.8602E-4	9.8602E-4	9.8603E-4	2.8029E-9	+
SATLBO [21]	9.8602E-4	9.8879E-4	1.0067E-3	4.8133E-6	+
SGDE [15]	9.8602E-4	9.8602E-4	9.8603E-4	2.4746E-9	-
SEDE [10]	9.8602E-4	9.8602E-4	9.8603E-4	4.2000E-17	-
CoDE [22]	9.8602E-4	9.8602E-4	9.8602E-4	2.3100E-17	-
Simple DE	9.8602E-4	9.8602E-4	9.8602E-4	2.9464E-17	-
MLBSA	2.4250E-3	2.4253E-3	2.4336E-3	1.5600E-6	+
TLABC	2.4250E-3	2.4254E-3	2.4287E-3	8.7464E-7	+
IJAYA	2.4250E-3	2.4251E-3	2.4253E-3	5.0766E-8	+
PGJAYA	2.4250E-3	2.4251E-3	2.4260E-3	1.7859E-7	+
SATLBO	2.4250E-3	2.4254E-3	2.4315E-3	1.1622E-6	+
SGDE	2.4250E-3	2.4250E-3	2.4250E-3	4.1697E-10	+
SEDE	2.4250E-3	2.4250E-3	2.4250E-3	3.1400E-17	-
CoDE	2.4250E-3	2.4250E-3	2.4250E-3	2.1700E-17	-
Simple DE	2.4250E-3	2.4250E-3	2.4250E-3	1.7547E-17	-
MLBSA	9.8248E-4	9.8506E-4	9.8613E-4	1.2400E-6	+
TLABC	1.0012E-3	1.2116E-3	1.9826E-3	2.1100E-4	+
IJAYA	9.8249E-4	9.8686E-4	9.9941E-4	3.2211E-6	+
PGJAYA	9.8260E-4	9.8603E-4	9.9599E-4	2.3666E-6	+
SATLBO	9.8282E-4	1.0054E-3	1.2306E-3	5.0271E-5	+
SGDE	9.8441E-4	9.8577E-4	9.8602E-4	4.0150E-7	+
SEDE	9.8248E-4	9.8289E-4	9.8602E-4	9.1700E-7	-
CoDE	9.8249E-4	1.0036E-3	1.5496E-3	1.0300E-4	+
Simple DE	9.8248E-4	9.8273E-4	9.8602E-4	8.9630E-7	-
PGJAYA	2.4250E-3	2.4272E-3	2.4485E-3	5.4346E-6	+
SEDE	2.4250E-3	2.4250E-3	2.4250E-3	6.6661E-17	-
Simple DE	2.4250E-3	2.4250E-3	2.4250E-3	2.7356E-17	-

there is no statistically significant evidence to conclude the performance difference.

Overall, our simple DE (Algorithm 2) and several strong competitors like SEDE can attain the best RMSE values. Statistically, the Mann-Whitney U test indicates that the accuracy of the simple DE is on par with the selected state-of-the-art approaches such as SEDE. The result is somewhat surprising given the extreme simplicity of the simple DE. Unfortunately, such simple metaheuristics have been largely overlooked in the current literature. Note that we compare the simple DE intentionally with three more complicated DE variants: SGDE, SEDE, and CoDE. In the two SDM cases in Table VI, the four DE algorithms exhibit almost identical accuracy in terms of RMSE values, while CoDE demonstrates the highest stability measured by the standard deviation and SGDE is the least stable one. By contrast, in the more challenging DDM cases, our simple DE outperforms both CoDE and SEDE with its enhanced performance stability.

The most apparent advantage of the simple DE is its substantially reduced running time (< 1 s). This impressive speedup is mainly brought by its extreme simplicity including only four computationally cheap equations in Algorithm 2. Besides, note that the evaluation of the objective function (5a) with a few dozens of data points is inexpensive, which implies consequently that it is usually the algorithm's internal computation burden that dominates the overall time consumption. This claim is supported particularly by the significantly longer runtime of the other three more complex DE variants in Fig. 4.

As for the "DDM+PW" case in Table VI, we notice that the best RMSE value attained by the three algorithms all turns out to be $2.4250E-3$, though this value is certainly not the global minimum (recall the upper bound ascertained in Table III). As mentioned in Section IV-C, this failure is possibly caused by the excessively small true value of I_{01} (see Table III) that can challenge all metaheuristic methods.

In Table VI, the RMSE values attained by various methods, on the level of $1E-4$ or $1E-3$, seem sufficiently small for practical applications. Such observations also justify the use of a simpler algorithm from another angle: the excessively high accuracy may be pragmatically unnecessary, and a practitioner can opt to trade off accuracy with algorithmic simplicity. Even better, extensive examinations above have validated the competitive accuracy of the simple DE method despite its extraordinary simplicity and efficiency.

V. CONCLUSION

In this paper, we tried to address two essential issues of PV parameter estimation that have been largely ignored in the current literature. With the two most widely used benchmark datasets, the globally minimum RMSE for the SDM and a reasonably tight upper bound for the DDM were certified rigorously by an interval analysis based B&B algorithm. However, the running time of this interval B&B algorithm is overly long for practical usages despite its theoretical guarantee. Next, we showed through extensive examinations that, for the first time and somewhat surprisingly, a simple DE algorithm (Algorithm 2) was capable of locating the global minimum or at least attaining the best-known result. Moreover, the simple and easy-to-tune DE algorithm is distinguished by its favorable performance stability and unmatched efficiency. Our findings imply that, unfortunately, many existing metaheuristics for PV parameter estimation might be overly complicated and risk over-engineering. We suggest that a practitioner start with the simple DE as the off-the-shelf tool, especially in real-time parameter estimation scenarios. Our code is available at <https://github.com/ShuhuaGao/rePVest> to ease reproduction.

REFERENCES

- [1] A. R. Jordehi, "Parameter estimation of solar photovoltaic (PV) cells: A review," *Renewable and Sustainable Energy Reviews*, vol. 61, pp. 354–371, Aug. 2016.
- [2] S. Li, W. Gong, X. Yan, C. Hu, D. Bai, L. Wang, and L. Gao, "Parameter extraction of photovoltaic models using an improved teaching-learning-based optimization," *Energy Conversion and Management*, vol. 186, pp. 293–305, Apr. 2019.
- [3] B. Yang, J. Wang, X. Zhang, T. Yu, W. Yao, H. Shu, F. Zeng, and L. Sun, "Comprehensive overview of meta-heuristic algorithm applications on PV cell parameter identification," *Energy Conversion and Management*, vol. 208, p. 112595, Mar. 2020.
- [4] M. G. Villalva, J. R. Gazoli, and E. R. Filho, "Comprehensive Approach to Modeling and Simulation of Photovoltaic Arrays," *IEEE Transactions on Power Electronics*, vol. 24, no. 5, pp. 1198–1208, May 2009.
- [5] R. Chenouard and R. A. El-Sehiemy, "An interval branch and bound global optimization algorithm for parameter estimation of three photovoltaic models," *Energy Conversion and Management*, vol. 205, p. 112400, Feb. 2020.
- [6] V. J. Chin, Z. Salam, and K. Ishaque, "Cell modelling and model parameters estimation techniques for photovoltaic simulator application: A review," *Applied Energy*, vol. 154, pp. 500–519, Sep. 2015.
- [7] H. G. G. Nunes, J. A. N. Pombo, S. J. P. S. Mariano, M. R. A. Calado, and J. A. M. Felipe de Souza, "A new high performance method for determining the parameters of PV cells and modules based on guaranteed convergence particle swarm optimization," *Applied Energy*, vol. 211, pp. 774–791, Feb. 2018.
- [8] K. Yu, J. J. Liang, B. Y. Qu, X. Chen, and H. Wang, "Parameters identification of photovoltaic models using an improved JAYA optimization algorithm," *Energy Conversion and Management*, vol. 150, pp. 742–753, Oct. 2017.
- [9] K. Yu, B. Qu, C. Yue, S. Ge, X. Chen, and J. Liang, "A performance-guided JAYA algorithm for parameters identification of photovoltaic cell and module," *Applied Energy*, vol. 237, pp. 241–257, Mar. 2019.
- [10] J. Liang, K. Qiao, K. Yu, S. Ge, B. Qu, R. Xu, and K. Li, "Parameters estimation of solar photovoltaic models via a self-adaptive ensemble-based differential evolution," *Solar Energy*, vol. 207, pp. 336–346, Sep. 2020.
- [11] X. Chen, B. Xu, C. Mei, Y. Ding, and K. Li, "Teaching-learning-based artificial bee colony for solar photovoltaic parameter estimation," *Applied Energy*, vol. 212, pp. 1578–1588, Feb. 2018.
- [12] W. Long, S. Cai, J. Jiao, M. Xu, and T. Wu, "A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models," *Energy Conversion and Management*, vol. 203, p. 112243, 2020.
- [13] R. Gutiérrez, J. M. Blanes, D. Marroquí, A. Garrigós, and F. J. Toledo, "System-on-Chip for Real-Time Satellite Photovoltaic Curves Telemetry," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 951–957, Mar. 2018.
- [14] T. Easwarakhanthan, J. Bottin, I. Bouhouch, and C. Boutrit, "Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers," *International journal of solar energy*, vol. 4, no. 1, pp. 1–12, 1986.
- [15] J. Liang, K. Qiao, M. Yuan, K. Yu, B. Qu, S. Ge, Y. Li, and G. Chen, "Evolutionary multi-task optimization for parameters extraction of photovoltaic models," *Energy Conversion and Management*, vol. 207, p. 112509, Mar. 2020.
- [16] C. A. Floudas, *Deterministic global optimization: theory, methods and applications*. Springer Science & Business Media, 2013, vol. 37.
- [17] E. Hansen and G. W. Walster, *Global optimization using interval analysis: revised and expanded*. CRC Press, 2003, vol. 264.
- [18] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
- [19] M. Čalasan, S. H. E. Abdel Aleem, and A. F. Zobaa, "On the root mean square error (RMSE) calculation for parameter estimation of photovoltaic models: A novel exact analytical solution based on Lambert W function," *Energy Conversion and Management*, vol. 210, p. 112716, Apr. 2020.
- [20] K. Yu, J. J. Liang, B. Y. Qu, Z. Cheng, and H. Wang, "Multiple learning backtracking search algorithm for estimating parameters of photovoltaic models," *Applied Energy*, vol. 226, pp. 408–422, Sep. 2018.
- [21] K. Yu, X. Chen, X. Wang, and Z. Wang, "Parameters identification of photovoltaic models using self-adaptive teaching-learning-based optimization," *Energy Conversion and Management*, vol. 145, pp. 233–246, Aug. 2017.
- [22] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 55–66, 2011.