Energy 99 (2016) 170–180

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

Parameters identification of solar cell models using generalized oppositional teaching learning based optimization

Xu Chen [a,*], Kunjie Yu [b], Wenli Du [b], Wenxiang Zhao [a], Guohai Liu [a]

[a] School of Electrical and Information Engineering, Jiangsu University, Zhenjiang, 212013, China
[b] Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai, 200237, China

ARTICLE INFO

Article history:
Received 9 October 2015
Received in revised form
17 January 2016
Accepted 18 January 2016
Available online 12 February 2016

Keywords:
Solar cell models
Parameter identification
Teaching learning based optimization
Generalized opposition-based learning

ABSTRACT

This paper presents a new optimization method called GOTLBO (generalized oppositional teaching learning based optimization) to identify parameters of solar cell models. GOTLBO employs generalized opposition-based learning to basic teaching learning based optimization through the initialization step and generation jumping so that the convergence speed is enhanced. The performance of GOTLBO is comprehensively evaluated in thirteen benchmark functions and two parameter identification problems of solar cell models, i.e., single diode model and double diode model. Simulation results indicate the excellent performance of GOTLBO compared with four well-known evolutionary algorithms and other parameter extraction techniques proposed in the literature.

1. Introduction

The increase in the cost of fossil fuels and their probable depletion, air pollution, global warming phenomenon, and severe environmental laws have resulted in renewable energy sources gaining the attention of many nations to produce electricity [1]. The PV (photovoltaic) system such as solar cell has been obtained increasing interest recently, because of several promising features like renewability, less pollution, ease of installation, and noise-free [2]. For PV systems, it is very important to select a model to closely emulate the characteristics of PV cells [3]. Several models have been introduced and proved to be successful in representing the behavior of the solar cell systems by considering many physical variables. Among them, two equivalent solar cell models are widely used in practice: single and double diode models [4]. The single diode model has five unknown parameters, so it is much more common to use. The double diode model contains seven unknown parameters and offers higher accuracy. Both single and double diode models require the knowledge of all unknown parameters, which is usually not provided by manufactures. Precise parameters of mathematical model play a key role in the simulation, evaluation, and optimization of solar cell systems. As a result, it is necessary to take into consideration the parameters identification with a feasible optimization method [5].

Two main approaches have been used in the literature to solve the parameter identification problems of solar cell models: deterministic and heuristic. Deterministic methods, such as least squares [6], Lambert W-functions [7], and the iterative curve fitting [8], impose several model restrictions such as convexity and differentiability in order to be correctly applied. Therefore, they are very sensitive to the initial solution, and most often lead to local optima.

With the development of soft computing technologies, many heuristic methods, which are of conceptual and computational simplicity, being excellent real-world problem solvers and robust to dynamic environments, capable of solving problems with no known solutions and with no need for analytic expression of the problems, have been applied to the parameter estimation problems of solar cell [9]. GA (Genetic algorithms) [10–12], PSO (particle swarm optimization) [13,14], DE (differential evolution) [15–18], SA (simulated annealing) [19], PS (pattern search) [20], HS (harmony search) [21], TLBO (teaching learning based optimization) [22], ABSO (artificial bee swarm optimization) [2], BBO (biogeography-based optimization) [9], SFLA (shuffled frog leaping algorithm) [23], CS (cuckoo search) [24], and FPA (flower pollination algorithm) [25] have been proposed to improve the parameter accuracy for cell models.

* Corresponding author.
E-mail address: xuchen@ujs.edu.cn (X. Chen).

http://dx.doi.org/10.1016/j.energy.2016.01.052

Nomenclature

a	diode ideality factor
a_1	Diffusion diode ideality factor
a_2	recombination diode ideality factor
I_d	diode currents (A)
I_{d1}	first diode currents (A)
I_{d2}	second diode currents (A)
I_L	cell output current (A)
I_{ph}	photo generated current (A)
I_{sd}	reverse saturation current of diode (A)
I_{sd1}	diffusion currents (A)
I_{sd2}	saturation currents (A)
I_{sh}	shunt resistor current (A)
k	Boltzmann constant (1.3806503×10^{23} J/K)
q	electron charge ($1.60217646 \times 10^{-19}$C)
R_S	series resistance (Ω)
R_{sh}	shunt resistance (Ω)
T	temperature of the junction (K)
V_L	cell output voltage (V)
V_t	junction thermal voltage (V)
ABSO	artificial bee swarm optimization
ANFES	average number of function evaluations
CLPSO	comprehensive learning particle swarm optimizer
CPSO	chaos particle swarm algorithm
DE	differential evolution
EA	evolutionary algorithm
FES	function evaluations
GA	genetic algorithm
GOBL	generalized opposition-based Learning
GOTLBO	generalized oppositional teaching learning based optimization
HS	harmony search
IGHS	innovative global harmony search
jDE	differential evolution with self-adaptive control parameter
NFES	number of function evaluations
OBL	opposition-based learning
OTLBO	opposition-based teaching learning based optimization
PS	pattern search
PSO	particle swarm optimization
R_{cr}-IJADE	repaired adaptive differential evolution
RMSE	root mean square error
SA	simulated annealing
SR	successful rate
STLBO	simplified teaching-learning based optimization
TLBO	teaching learning based optimization
VTR	value to reach

TLBO [26] is a recently proposed population-based algorithm, which simulates the teaching and learning process in the classroom. In this algorithm, students gather knowledge from the lecture delivered by teacher and also through the mutual interaction with other students. TLBO has emerged as one of the simplest and most efficient techniques. It requires few parameters and performs well on many optimization problems. The effectiveness of TLBO and its modified versions have been reported in continuous optimization problems [27,28], combinatorial problems [29], as well as real-world engineering problems [30–31].

In this paper, we develop a new optimization method called GOTLBO (oppositional teaching learning based optimization) for the parameters extraction of both single and double diode models. In GOTLBO, GOBL (generalized opposition-based learning) is integrated with original TLBO through the initialization step and generation jumping so that the convergence speed can be enhanced. The performance of GOTLBO is firstly evaluated on 13 well-known benchmark functions, and compared with those of four EAs (evolutional algorithms), including jDE, CLPSO, TLBO, and OTLBO. GOTLBO performs better than these EAs on the majority of the test functions. Then, GOTLBO is employed to identify the parameters for two solar cell models, i.e., single diode model and double diode model. The simulation results demonstrate that the performance of GOTLBO is very competitive compared with other parameter identification techniques proposed in the literature.

The rest of the paper is organized as follows. Section Problem statement states the problem formulations of solar cell models. Section TLBO briefly introduces basic TLBO. Our proposed approach is presented in Section Proposed approach: GOTLBO. In Section Evaluation GOTLBO on benchmark functions, GOTLBO is evaluated on 13 benchmark functions. Followed by Section Application to parameter identification of solar cell models, GOTLBO is used to solve parameter extraction problems of solar cell models. Finally, the last section concludes this paper.

2. Problem statement

2.1. Single diode model

In the single diode model, as shown in Fig. 1, the output current of solar cell can be formulated as follows [12,17]:

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

where I_L is the cell output current, I_{ph} is the photo generated current, I_d is the diode currents, and I_{sh} is the shunt resistor current. According the Shockley equation, the diode currents I_d can be calculated as:

$$I_d = I_{sd}\left(exp\left(\frac{V_L + I_L R_S}{aV_t}\right) - 1\right) \tag{2}$$

where I_{sd} is reverse saturation current of diode, V_L is the cell output voltage, a is the diode ideality factor, R_S is the series resistance, and V_t is the junction thermal voltage as:

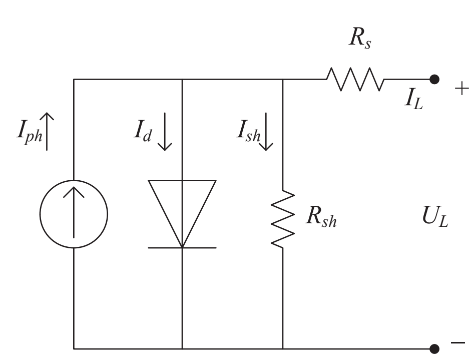

Fig. 1. Equivalent circuit of a single diode model.

$$V_t = \frac{kT}{q} \tag{3}$$

where k is the Boltzmann constant (1.3806503×10^{23} J/K), T is the temperature of the junction in Kelvin, and q is the electron charge ($1.60217646 \times 10^{-19}C$).

The shunt resistor current I_{sh} is formulated as

$$I_{sh} = \frac{V_L + I_L R_s}{R_{sh}} \tag{4}$$

where R_{sh} denotes the shunt resistance.

In the above single diode model, there are five parameters (i.e., I_{ph}, I_{sd}, R_S, R_{sh} and a) that need to be extracted from the I–V data of the solar cell.

2.2. Double diode model

The single diode models inherently neglect the effect of recombination current loss in the depletion region. Consideration of this loss, particularly at low voltage, leads to a more precise solution, known as the two-diode model, with reference to Fig. 2. The current–voltage relationship can be formulated as [12,17]:

$$\begin{aligned} I_L &= I_{ph} - I_{d1} - I_{d2} - I_{sh} \\ &= I_{ph} - I_{sd1}\left(exp\left(\frac{V_L + I_L R_S}{a_1 V_t}\right) - 1\right) - I_{sd2}\left(exp\left(\frac{V_L + I_L R_S}{a_2 V_t}\right) - 1\right) \\ &\quad - \frac{V_L + I_L R_s}{R_{sh}} \end{aligned} \tag{5}$$

where I_{d1} and I_{d2} are the first and second diode currents, I_{sd1} and I_{sd2} are respectively the diffusion and saturation currents, a_1 and a_2 denote the diffusion and recombination diode ideality factors, respectively. In the double diode model, there are seven parameters (i.e., I_{ph}, I_{sd1}, I_{sd2}, R_S, R_{sh} a_1 and a_2) that need to be extracted from the I–V data of the solar cell.

2.3. Objective function

It is noted that current equation is nonlinear transcendental function. In order to extract the parameters of different solar cell models from the I–V data using the optimization techniques, we first need to define the objective function to be optimized. In this work, the RMSE (root mean square error) is used as the objective function [2,17], which is described as:

$$F(x) = \sqrt{\frac{1}{N}\sum_{k=1}^{N} f_k(V_L, I_L, x)^2} \tag{6}$$

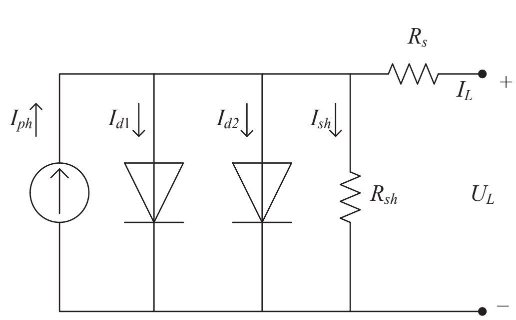

Fig. 2. Equivalent circuit of a double diode model.

where N is the number of experimental data.

For the single diode model

$$f_k(V_L, I_L, x) = I_{ph} - I_{sd}\left(exp\left(\frac{V_L + I_L R_S}{a V_t}\right) - 1\right) - \frac{V_L + I_L R_s}{R_{sh}} - I_L \tag{7}$$

$$x = \left\{I_{ph}, I_{sd}, R_S, R_{sh}, a\right\} \tag{8}$$

For the double diode model

$$\begin{aligned} f_k(V_L, I_L, x) &= I_{ph} - I_{sd1}\left(exp\left(\frac{V_L + I_L R_S}{a_1 V_t}\right) - 1\right) \\ &\quad - I_{sd2}\left(exp\left(\frac{V_L + I_L R_S}{a_2 V_t}\right) - 1\right) - \frac{V_L + I_L R_s}{R_{sh}} - I_L \end{aligned} \tag{9}$$

$$x = \left\{I_{ph}, I_{sd1}, I_{sd2}, R_S, R_{sh}, a_1, a_2\right\} \tag{10}$$

In Eqs. (7)–(10), x is decision vector which consists of the parameters to be extracted. Obviously, in order to make the simulated data better fit the experimental data, the objective function $F(x)$ in Eq. (6) needs to be minimized. The smaller the objective function, the better the solutions obtained.

3. TLBO

TLBO is a recently proposed population-based evolutionary algorithm, and the main idea behind it is the simulation of a classical learning process consisting of a teacher phase and a learner phase [27]. The best solution in the entire population obtained so far is considered as the teacher, and the teacher shares his or her knowledge with the students to improve their outputs (i.e., grades or marks) in the teacher phase. The learners also learn knowledge from other students to improve their outputs in the learner phase. TLBO process is conducted through two basic operations of the teacher and learner phases.

3.1. Teacher phase

In teacher phase, the best-so-far learner is chosen as the teacher according to the fitness value, and the mean position of all learners should be calculated for updating the positions of all learners. Assume $X_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ is the position of the i-th learner for an n-dimensional optimization problem, the teaching process is formulated as follows:

$$X_{new,i} = X_{old,i} + rand()*(X_{Teacher} - TF*X_{Mean}) \tag{11}$$

where $X_{new,i}$ and $X_{old,i}$ are the new and old positions of i-th learner, $X_{Teacher}$ is the position of the current teacher, $X_{Mean} = \frac{1}{NP}\sum_{i=1}^{NP} X_i$ is the mean position of the current class, NP is the number of students, $rand()$ is the random number within the range [0,1], TF is the teaching factor, and its value is heuristically set to either 1 or 2. All learners should be re-evaluated after each iteration of Teacher Phase. If $X_{new,i}$ is better than $X_{old,i}$, $X_{new,i}$ will be accepted and flowed to learner phase, otherwise $X_{old,i}$ is not changed.

3.2. Learner phase

In learner phase, for the i-th learner X_i, another learner X_j which is different from X_i will be randomly selected from the class. The learning process is described as follows:

$$X_{new,i} = \begin{cases} X_{old,i} + rand()*\left(X_{old,i} - X_{old,j}\right) & if\ f(X_{old,i}) < f\left(X_{old,j}\right) \\ X_{old,i} + rand()*\left(X_{old,j} - X_{old,i}\right) & otherwise \end{cases} \tag{12}$$

where $X_{new,i}$ is the new position of i-th learner, $X_{old,i}$ and $X_{old,j}$ are the old positions of i-th and j-th learners, $rand()$ is the random number in the range [0,1]. $X_{new,i}$ is accepted if it gives a better function value.

4. Proposed approach: GOTLBO

4.1. Opposition-based learning

OBL (Opposition-based learning), developed by Tizhoosh [32], is a new concept in computational intelligence, which considers current estimate and its opposite estimate simultaneously in order to achieve a better approximation for a current candidate solution. It has been proved that an opposite candidate solution has a better chance to be closer to the global optimum solution than a random candidate solution [33]. The idea of OBL has been used by many researchers to enhance the convergence speed of various optimization approaches [34–36].

In OBL, the opposite number is defined as the mirror point of the solution from the center of the search space. In a one-dimensional search space, it is defined as follows:

$$x^O = a + b - x \tag{13}$$

where a, b are the extreme points of the search space.

For n-dimensional search space, the opposite point is defined as follows:

$$x_{ij}^O = a_j + b_j - x_{ij}; \quad x_{ij} \in \left[a_j, b_j\right] \quad j = 1, 2, \cdots, n \tag{14}$$

Wang et al. proposed generalized OBL (GOBL) by transforming candidates in current search space to a new search space [37]. The GOBL model is defined as follows:

$$x_{ij}^{GO} = k\left(a_j + b_j\right) - x_{ij}; \quad x_{ij} \in \left[a_j, b_j\right] \quad j = 1, 2, \cdots, n \tag{15}$$

where k is a randomly generated number in the interval [0, 1]. When the opposite candidate x_{ij}^{GO} jumps out of the definition domain $[a_j,b_j]$, the opposite candidate is assigned to a random value within $[a_j,b_j]$.

The generalized opposition-based optimization can be described as followed: let $X_i = \{x_{i1},x_{i2},\ldots,x_{in}\}$ be a point in n-dimensional space (i.e., a candidate solution). Assume $f(\cdot)$ is a fitness function which is used to measure the candidate's fitness. According to the definition of the generalized opposite point, $X_i^{GO} = \{x_{i1}^{GO}, x_{i2}^{GO}, \cdots, x_{i,n}^{GO}\}$ is the generalized opposite of X_i. Now, if $f(X_i^{GO})$ is better than $f(X_i)$, then point X_i can be replaced with X_i^{GO}; otherwise, we continue with X_i. Hence, the point X_i and its opposite point X_i^{GO} are evaluated simultaneously in order to continue with the fitter one.

4.2. The GOTLBO algorithm

The core ideal of GOTLBO is simultaneous consideration of the original population and the corresponding generalized opposite population to accelerate the convergence speed of the basic TLBO. GOTLBO utilizes GOBL scheme in two steps: generalized opposition-based initialization and generalized opposition-based generation jumping. The steps of GOTLBO are presented in Fig. 3.

It is also important to note that, by staying within variables' interval static boundaries, we would jump outside of the already shrunken search space and the knowledge of the current reduced space (converged population) would be lost [34]. Hence, in generalized opposition-based generation jumping, we calculate opposite points by using variables' current interval in the population which is, as the search does progress, increasingly smaller than the corresponding initial range.

Fig. 4 depicts the flowchart of the GOTLBO which is used to identify the optimal parameters of the solar cell models.

5. Evaluation of GOTLBO on benchmark functions

5.1. Experimental settings

GOTLBO is evaluated on 13 widely used benchmark functions with different characteristics selected from Ref. [38]. The detail descriptions of these functions are presented in Table 1. These functions can be classified into two groups according to their properties: (1) unimodal functions f_1~f_7; (2) multimodal functions f_8~f_{13}.

GOTLBO is compared with four commonly used EAs in the literature, namely jDE [39], CLPSO [40], TLBO [26], and OTLBO [41]. jDE [39], proposed by Brest et al., is an improved DE with self-adaptive control parameter approach. CLPSO [40] proposed by Liang et al., obtains promising results in the PSO literature, which updates a particle's velocity using all other particles' historical best information. TLBO [26], proposed by Rao et al., is the basic TLBO algorithm. OTLBO [41], proposed by Roy et al., incorporates basic OBL into TLBO to improve the performance. The parameter configurations for all involved algorithms are based on the suggestions in the corresponding literature and summarized in Table 2.

The following four performance criteria [42] are adopted to measure the final performance of each algorithm.

- **Error**: The error of a solution x is defined as $f(x)$–$f(x^*)$, where x^* is the global minimum provided in Table 1. The minimum error is recorded when the maximum number of functional evaluations $MaxFES$ is reached in 30 runs. The mean and standard deviation of the error values are calculated as well. The error value measures the solution accuracy of an algorithm.
- **SR (successful rate)**: A successful run indicates that the algorithm can reach VTR (value to reach) before the $MaxFES$ condition terminates the trial. SR is calculated as the number of successful runs divided by the total number of runs. The SR value measures the reliability of an algorithm.
- **ANFES:** The average number of function evaluations (ANFES) is also recorded when the required VTR is obtained. The ANFES value measures the convergence speed of an algorithm.
- **Convergence graphs**: The convergence graphs show the mean error performance of the best solution over the total number of runs, in the respective experiments.

All algorithms are executed over 30 independent runs to make a fair comparison. For functions $f_1(x)$~$f_{13}(x)$, the maximum number of functional evaluations $MaxFES = 10000D$ [42], and $VTR = 10^{-8}$ [42].

5.2. Analysis of jumping rate parameter

GOTLBO contains one parameter jumping rate Jr. In order to investigate the sensitivity of this parameter, we tested GOTLBO under different Jr values: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0. The error values of GOTLBO with different Jr is detailed in Table 3.

Step 1: Generalized opposition-based initialization:

1.1 Randomly initialize the entire individuals of population $P = \{X_1, X_2, \ldots, X_{NP}\}$ within the upper and lower limits;

1.2 Generate generalized opposite population $GOP = \{X_1^{GO}, X_2^{GO}, \ldots, X_{NP}^{GO}\}$;

1.3 Evaluate fitness of the population P and generalized opposite population GOP;

1.4 Select NP fittest individuals from the set $\{P \cup GOP\}$ as initial population.

Step 2: Teacher phase

2.1 For each student X_i, modify the student by gathering knowledge from the teacher using Eq.(11);

2.2 If $X_{new,i}$ is better than $X_{old,i}$, use $X_{new,i}$ to replace $X_{old,i}$.

Step 3: Learner phase

3.1 For each student X_i, update the student through the mutual interaction with other students using Eq. (12);

3.2 If $X_{new,i}$ is better than $X_{old,i}$, use $X_{new,i}$ to replace $X_{old,i}$.

Step 4: Generalized opposition-based generation jumping

4.1 If $rand < Jr$ (Jr is jumping rate), generate the generalized opposite population GOP;

4.2 Select NP fittest individuals from current population P and the generalized opposite population GOP as the new current population.

Step 5: If the termination criteria is satisfied, stop; otherwise go to Step 2.

Fig. 3. GOTLBO.

The best results obtained by GOTLBO with specific jumping rate Jr are highlighted with shade.

From Table 3, a large Jr is beneficial on unimodal functions, as GOTLBO with $Jr = 1.0$ exhibits the best results on 5 functions f_1~f_4, f_6. On multimodal functions, GOTLBO with $Jr = 0.4,0.5,0.9,1.0$ achieves best results on 3 functions, and GOTLBO with $Jr = 0.1,0.2,0.3,0.6,0.8,0.9$ achieves best results on 2 functions. GOTLBO with $Jr = 0$ performs poor on the most of functions, which indicates the importance of generalized opposition-based generation jumping.

Overall, GOTLBO with $Jr = 1.0$ can provide the best results on the test functions. Hence, it is used in the following comparisons with other EAs on benchmark functions.

5.3. Results on benchmark functions

Table 4 shows the error values of five EAs. It can be seen from Table 4 that GOTLBO obtains the best performance on all the unimodal functions except f_5. The Wilcoxon's rank-sum test results indicate that GOTLBO gets significantly better results than all other EAs on 4 functions f_2~f_4,f_7. CLPSO, TLBO, and OTLBO show the best performance on 2 functions. jDE only gets the best performance on only 1 functions f_6.

On multimodal functions f_8~f_{13}, jDE generally performs best among five EAs, as it gets best results on 5 functions f_8~f_9, f_{11}~f_{13}. GOTLBO is also very competitive, with the best performance on 3 functions f_9~f_{11}. Especially, GOTLBO achieves significantly better results than all other EAs on f_{10}. CLPSO, TLBO, and OTLBO get the best performance on only 1 function, respectively.

According to the last three rows of Table 4, GOTLBO is significantly better than jDE, CLPSO, TLBO and OTLBO on 6, 8, 7, and 7 functions, respectively. It is significantly worse than jDE, CLPSO, TLBO and OTLBO on 3, 3, 1, and 2 functions, and similar to them on 4, 2, 5, and 4 functions, respectively. Overall, GOTLBO generates the best results among five EAs on the test functions.

Table 5 shows the SR and ANFES values among five EAs. Three TLBO methods (i.e., TLBO, OTLBO, and GOTLBO) obtain the best SR values on 5 unimodal functions f_1~f_4,f_6. jDE and CLPSO obtains the best SR values on 3 unimodal functions f_1~f_2,f_6. Considering the ANFES values, GOTLBO gets the better results than other four EAs on 5 unimodal functions f_1~f_4,f_6.

jDE and CLPSO both obtain the best SR values on 5 multimodal functions. GOTLBO achieves the best SR values on 4 multimodal functions. Considering the ANFES values, GOTLBO gets the best results among five EAs on 3 multimodal functions. jDE and OTLBO both get the best results on 1 multimodal functions.

The convergence graphs of five EAs on some typical functions are plotted in Fig. 5. It is clear that GOTLBO convergences faster than all other EAs in the entire search process on 2 unimodal functions f_1,f_4. On multimodal functions f_9,f_{10}, the convergence speed of GOTLBO is the best at the earlier stage. However, this fast convergence speed may deteriorates the diversity of GOTLBO, as GOTLBO is surpassed by jDE and CLPSO at the later search stage on function f_9.

5.4. Discussion

GOTLBO has been evaluated on 13 widely used functions and compared with four representative EAs. Based on the above experiments, we can summarize that:

- A larger jumping rate Jr is beneficial for GOTLBO on unimodal functions. GOTLBO is not sensitive to Jr between 0.1 and 1 on the most of multimodal functions we have tested on.
- GOTLBO converge faster than jDE, CLPSO, TLBO, and OTLBO on unimodal functions. Meanwhile, GOTLBO can achieve the highest precise solutions among the five EAs on the most of unimodal functions.
- GOTLBO also converge faster than jDE, CLPSO, TLBO, and OTLBO on many multimodal functions. But this fast convergence speed sometimes may deteriorate the diversity of GOTLBO and lead to premature convergence.

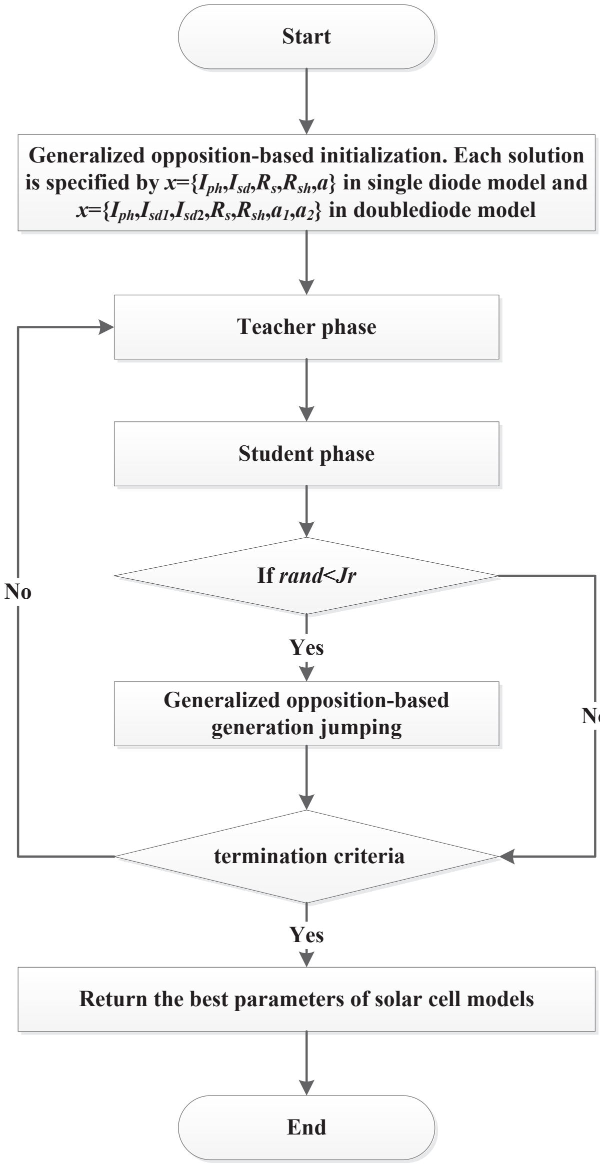

Fig. 4. Flowchart of the parameters extraction process by GOTLBO.

- GOTLBO is particularly suitable for unimodal functions. For some multimodal functions whose landscape is not very complex, GOTLBO is also very useful.

6. Application to parameter identification of solar cell models

In this section, the I–V characteristic of a 57 mm diameter commercial (R.T.C. France) silicon solar cell is used to evaluate the efficiency of the GOTLBO-based parameter identification method. The experimental data has been adopted from the system under 1 sun (1000 W/m^2) at 33C [6]. In the single and double diode models, the lower and upper boundaries of each parameter are shown in Table 6, which is the same as used in Refs. [2,17].

6.1. Comparison among different EAs

To record the RMSE values, *MaxFES* is set to be 10,000 and 20,000 for the single diode model and double diode model, respectively. The required accuracy VTR is set to be 0.002 for both single and double diode models to record the SR and NFES values.

Table 1
Benchmark functions.

Benchmark function	Dimension	Domain	Optimum
$f_1=\sum_{i=1}^{D}x_i^2$	30	$[-100,100]^D$	0
$f_2=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	30	$[-10,10]^D$	0
$f_3=\sum_{i=1}^{D}(\sum_{j=1}^{i}x_j)^2$	30	$[-100,100]^D$	0
$f_4=max_i\{\lvert x_i\rvert,1\le i\le D\}$	30	$[-100,100]^D$	0
$f_5(x)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(1-x_i)^2]$	30	$[-30,30]^D$	0
$f_6=\sum_{i=1}^{D}[\lvert x_i+0.5\rvert]^2$	30	$[-100,100]^D$	0
$f_7=\sum_{i=1}^{D}ix_i^4+random(0,1)$	30	$[-1.28,1.28]^D$	0
$f_8=-\sum_{i=1}^{D}(x_i sin(\sqrt{\lvert x_i\rvert}))$	30	$[-500,500]^D$	−12 569.487
$f_9=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10)$	30	$[-5.12,5.12]^D$	0
$f_{10}=-20\exp\left(-0.2\times\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e$	30	$[-32,32]^D$	0
$f_{11}=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	$[-600,600]^D$	0
$f_{12}=\frac{\pi}{D}\{100\sin^2(\pi y_1)+\sum_{i=1}^{D-1}(y_i-1)^2[1+10sin^2(\pi y_{i+1})]+(y_n-1)\}+\sum_{i=1}^{D}u(x_i,10,100,4)y_i=1+(x_i+1)/4,\ u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, & x_i>a\\ 0, & -a\le x_i\le a\\ k(-x_i-a)^m, & x_i<-a\end{cases}$	30	$[-50,50]^D$	0
$f_{13}=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{D-1}(x_i-1)^2[1+sin^2(3\pi x_{i+1})]+(x_n-1)\}+\sum_{i=1}^{D}u(x_i,5,100,4)$	30	$[-50,50]^D$	0

Table 2
Parameter settings of five EAs.

Algorithm	Parameters
jDE	$NP = 100$, $\tau_1 = \tau_2 = 0.1$, $F_l = 0.1$,$F_u = 0.9$;
CLPSO	$NP = 40$, w linearly decreasing from 0.9 to 0.2, $c = 1.494$, $m = 5$;
TLBO	$NP = 50$
OTLBO	$NP = 50$, $Jr = 0.3$
GOTLBO	$NP = 50$, $Jr = 1.0$

The population size is set as $NP = 20$ for all five EAs, and each algorithm is run 30 times independently.

Table 7 shows the results of GOTLBO with different and four other EAs for the single diode model. From Table 7, GOTLBO is not sensitive to Jr between 0.1 and 0.6. The minimal RMSE value of GOTLBO with $Jr = 0.1$ is 9.87442E−04, which is very close to the best value 9.86026E−04, obtained by OTLBO. Considering the SR, GOTLBO with $Jr = 0.1$~0.6 achieve a value of 100%. GOTLBO with $Jr = 0.4$ gets a value of 2739 in terms of ANFE. Overall, OTLBO exhibits the best performance among all EAs for the single diode model. The performance of GOTLBO is also very competitive.

Table 8 shows the results of GOTLBO with different Jr and four other EAs for the double diode model. GOTLBO can achieve satisfactory results with most of Jr values. The best RMSE and ANFES values of GOTLBO are better than those of other four EAs. GOTLBO with $Jr = 0$~0.1, 0.4~0.8 get a value of 100% in terms of SR.

6.2. Comparison with results in the literature

For the single diode model, the extracted parameters and best RMSE value of GOTLBO are compared with those of GA [12], CPSO [14], PS [20], SA [19], IGHS [21], ABSO [2], R_{cr}-IJADE [17], and STLBO [22]. These methods are chosen for comparison due to their good performance in the single diode model. The experimental results are reported in Table 9. From Table 9, R_{cr}-IJADE and STLBO provide the best RMSE value (9.8602E−04), and GOTLBO gets the second best RMSE value (9.87442E−04), followed by ABSO (9.9124E−04), IGHS (9.9306E−04), CPSO (1.3900E−03), PS (1.4940E−02), SA (1.9000E−02), and GA (1.9080E−02). NFES in the last row of Table 9 is the number of function evaluations to terminate the algorithms, which reflects the computational overhead. The NFES of GOTLBO and R_{cr}-IJADE are only 10,000, which is much less than those of STLBO (50,000), ABSO (150,000), IGHS (150,000), and CPSO (45,000). The NFES for GA, PS, and SA are not available in the literature.

For the double diode model, seven parameters need to be extracted. The extracted parameters and best RMSE value of GOTLBO are compared with those of PS [20], SA [19], IGHS [21], ABSO [2], R_{cr}-IJADE [17], and STLBO [22]. Table 10 shows the results of different methods. From Table 10, R_{cr}-IJADE and STLBO provide the best RMSE values (9.8248E−04). The RMSE of GOTLBO (9.83177E−04) is the second best, which is very close to that of R_{cr}-IJADE and STLBO. The other methods are ranked as ABSO (9.8344E−04), IGHS (9.8635E−04), PS (1.5180E−02), and SA (1.6640E−02). The NFES of GOTLBO and R_{cr}-IJADE are 20,000, smaller than STLBO (50,000), ABSO (150,000), and IGHS (150,000).

The I−V characteristic obtained by GOTLBO between the experimental data and simulated data are shown in Fig. 6. The results clearly indicate that the simulated data generated by GOTLBO are highly coincide with the experimental data both in the single and double diode model, which means that the extracted parameters of GOTLBO are very accurate.

7. Conclusions

In this paper, we have developed a new optimization method called GOTLBO, which can be efficiently used to extract the parameters of solar cell models. GOTLBO employs the concept of GOBL to accelerate the convergence speed of original TLBO through the initialization step and generation jumping. It is firstly

Table 3
Comparison on the Error values of GOTLBO with different jumping rate Jr.

Jr		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
f1	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f2	Mean	3.41E−272	2.19E−266	1.90E−264	1.07E−262	1.20E−262	9.95E−264	3.56E−265	7.60E−268	7.52E−274	1.08E−276	4.23E−282
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f3	Mean	6.85E−122	2.02E−127	8.62E−143	6.46E−165	3.71E−186	2.70E−211	9.06E−232	3.39E−251	9.01E−272	1.81E−289	6.47E−302
	SD	2.05E−121	1.07E−126	3.33E−142	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f4	Mean	3.38E−222	1.28E−224	3.47E−227	2.05E−231	1.79E−236	3.25E−244	5.72E−253	6.02E−261	4.90E−269	5.60E−279	3.50E−287
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f5	Mean	6.80E+00	5.47E+00	4.99E+00	5.46E+00	5.65E+00	6.54E+00	7.03E+00	7.88E+00	8.02E+00	8.54E+00	9.25E+00
	SD	2.58E+00	1.70E+00	1.89E+00	2.13E+00	1.64E+00	2.22E+00	1.93E+00	1.77E+00	2.12E+00	2.00E+00	2.32E+00
f6	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f7	Mean	1.63E−04	1.17E−04	1.11E−04	1.07E−04	1.14E−04	1.01E−04	8.82E−05	1.01E−04	9.68E−05	8.18E−05	9.94E−05
	SD	6.55E−05	5.74E−05	4.26E−05	4.16E−05	5.28E−05	4.78E−05	3.66E−05	4.69E−05	4.42E−05	3.90E−05	5.36E−05
f8	Mean	3.26E+03	3.12E+03	3.22E+03	3.11E+03	3.34E+03	3.12E+03	3.24E+03	3.18E+03	3.28E+03	3.27E+03	2.98E+03
	SD	7.34E+02	5.68E+02	7.02E+02	6.56E+02	6.87E+02	5.37E+02	5.77E+02	6.90E+02	6.88E+02	6.48E+02	6.46E+02
f9	Mean	8.75E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	5.01E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f10	Mean	3.55E−15	3.43E−15	3.43E−15	3.55E−15	3.32E−15	3.43E−15	3.20E−15	2.96E−15	3.20E−15	2.49E−15	2.49E−15
	SD	0.00E+00	6.49E−16	6.49E−16	0.00E+00	9.01E−16	6.49E−16	1.08E−15	1.35E−15	1.08E−15	1.66E−15	1.66E−15
f11	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f12	Mean	3.46E−03	9.23E−31	9.81E−31	6.53E−31	4.64E−31	8.55E−31	3.46E−03	7.92E−31	6.63E−31	7.06E−31	9.46E−29
	SD	1.89E−02	1.39E−30	3.44E−30	7.59E−31	7.25E−31	1.19E−30	1.89E−02	8.54E−31	7.82E−31	7.89E−31	5.15E−28
f13	Mean	2.80E−02	2.50E−02	2.88E−02	2.50E−02	2.77E−02	3.98E−02	3.83E−02	1.43E−02	2.49E−02	2.10E−02	2.75E−02
	SD	4.20E−02	3.71E−02	5.57E−02	3.74E−02	4.27E−02	5.68E−02	6.77E−02	1.91E−02	5.66E−02	3.41E−02	4.59E−02

The best results obtained by GOTLBO with specific jumping rate Jr are highlighted with shade.

Table 4
Comparison on the **Error** values among five EAs.

		jDE		CLPSO		TLBO		OTLBO		GOTLBO
f1	Mean	2.23E−61	+	3.48E−25	+	0.00E+00	=	0.00E+00	=	0.00E+00
	SD	3.34E−61		2.51E−25		0.00E+00		0.00E+00		0.00E+00
f2	Mean	2.81E−36	+	1.18E−15	+	5.83E−272	+	1.08E−228	+	4.23E−282
	SD	3.17E−36		4.59E−16		0.00E+00		0.00E+00		0.00E+00
f3	Mean	4.33E−07	+	3.91E+02	+	7.99E−121	+	4.66E−103	+	6.47E−302
	SD	5.58E−07		9.89E+01		4.32E−120		1.25E−102		0.00E+00
f4	Mean	4.87E−01	+	2.63E+00	+	7.16E−223	+	8.45E−211	+	3.50E−287
	SD	1.04E+00		4.04E−01		0.00E+00		0.00E+00		0.00E+00
f5	Mean	9.38E+00	=	9.18E−01	−	6.93E+00	−	1.59E+01	+	9.25E+00
	SD	1.18E+00		1.07E+00		2.51E+00		8.20E−01		2.32E+00
f6	Mean	0.00E+00	=	0.00E+00	=	0.00E+00	=	0.00E+00	=	0.00E+00
	SD	0.00E+00		0.00E+00		0.00E+00		0.00E+00		0.00E+00
f7	Mean	3.53E−03	+	3.74E−03	+	1.66E−04	+	1.65E−04	+	9.94E−05
	SD	8.50E−04		9.35E−04		5.12E−05		7.25E−05		5.36E−05
f8	Mean	1.34E−02	−	1.34E−02	−	3.71E+03	+	3.29E+03	=	2.98E+03
	SD	0.00E+00		3.32E−13		7.27E+02		5.73E+02		6.46E+02
f9	Mean	0.00E+00	=	0.00E+00	=	8.24E+00	+	7.66E+00	+	0.00E+00
	SD	0.00E+00		0.00E+00		6.13E+00		6.27E+00		0.00E+00
f10	Mean	3.79E−15	+	3.04E−13	+	3.55E−15	+	3.55E−15	+	2.49E−15
	SD	9.01E−16		7.38E−14		0.00E+00		0.00E+00		1.66E−15
f11	Mean	0.00E+00	=	9.07E−16	+	0.00E+00	=	0.00E+00	=	0.00E+00
	SD	0.00E+00		2.28E−15		0.00E+00		0.00E+00		0.00E+00
f12	Mean	1.57E−32	−	1.69E−26	+	3.46E−03	=	1.24E−31	−	9.46E−29
	SD	5.57E−48		8.88E−27		1.89E−02		4.00E−31		5.15E−28
f13	Mean	1.35E−32	−	4.15E−25	−	2.66E−02	=	6.59E−03	−	2.75E−02
	SD	5.57E−48		3.08E−25		3.34E−02		1.34E−02		4.59E−02
Win			6		8		7		7	
Tie			4		2		5		4	
Lose			3		3		1		2	

"+", "−", and "=" symbolize that the performance of the GOTLBO is better or worse than, and similar to that of the EA, respectively, according to the Wilcoxon rank-sum test at the 5% significance level.
The best results obtained by the five EAs are highlighted with shade.

evaluated on 13 benchmark functions and compared with four representative EAs, namely jDE, CLPSO, TLBO, and OTLBO. Simulation results show the excellent performance of GOTLBO on most of test functions. The simulation results and statistical analyses show that GOTLBO performs better than these algorithms on most of the test functions.

GOTLBO is simple and straightforward to implement, which makes it be ease of use for real-world problems. GOTLBO has been

Table 5
Comparison on the SR and ANFES values among five EAs.

		jDE	CLPSO	TLBO	OTLBO	GOTLBO
f1	SR	100%	100%	100%	100%	100%
	ANFES	59484	153306	6921	7908	6261
f2	SR	100%	100%	100%	100%	100%
	ANFES	81578	196098	11118	13024	10386
f3	SR	3%	0%	100%	100%	100%
	ANFES	NA	NA	28838	34539	11736
f4	SR	0%	0%	100%	100%	100%
	ANFES	NA	NA	13311	14004	10311
f5	SR	0%	0%	0%	0%	0%
	ANFES	NA	NA	NA	NA	NA
f6	SR	100%	100%	100%	100%	100%
	ANFES	22504	66950	2484	2878	2246
f7	SR	0%	0%	0%	0%	0%
	ANFES	NA	NA	NA	NA	NA
f8	SR	0%	0%	0%	0%	0%
	ANFES	NA	NA	NA	NA	NA
f9	SR	100%	100%	27%	33%	100%
	ANFES	117884	227858	NA	NA	14876
f10	SR	100%	100%	100%	100%	100%
	ANFES	90394	219395	10644	12063	9626
f11	SR	100%	100%	100%	100%	100%
	ANFES	62068	192608	7361	8243	6411
f12	SR	100%	100%	97%	100%	100%
	ANFES	54118	140870	NA	29636	42546
f13	SR	100%	100%	30%	70%	27%
	ANFES	58301	152567	NA	NA	NA

* NA indicates the required accuracy level is not obtained after MaxFES.
The best results obtained by the five EAs are highlighted with shade.

Table 6
Ranges of the parameters of the single and double diode models.

Parameter	Lower bound	Upper bound
$I_{ph}(A)$	0	1
$I_{sd}(\mu A)$	0	1
$R_s(\Omega)$	0	0.5
$R_{sh}(\Omega)$	0	100
a	1	2

applied to solve parameter identification problems of two solar cell models, i.e., single diode model and double diode model. Compared with other techniques proposed in the literature such as PS, SA, IGHS, ABSO, and Rcr-IJADE, our GOTLBO performs better, or at least comparably, in terms of solution accuracy and computational overhead. Therefore, GOTLBO can be used as an efficient and reliable alternative for parameter identification problems of solar cell models.

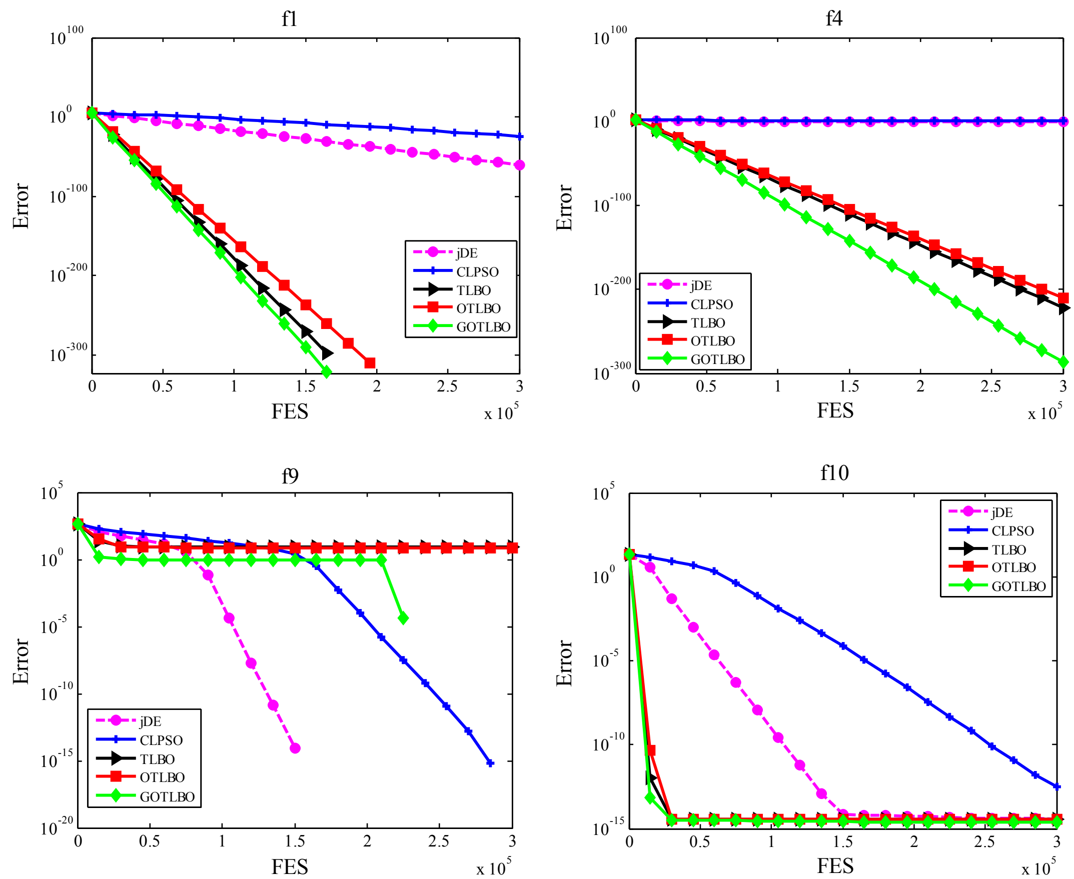

Fig. 5. Convergence graphs.

Table 7
Comparison of GOTLBO with four EAs for the single diode model.

Algorithm	RMSE				SR	ANFES	
	Min	Mean	Max	SD		Mean	SD
jDE	9.86022E−04	1.14054E−03	1.72897E−03	1.76520E−04	100%	1738	609
CLPSO	1.05672E−03	1.68858E−03	3.03829E−03	5.07229E−04	73%	NA	NA
TLBO	9.87010E−04	1.29346E−03	1.78338E−03	2.23610E−04	100%	2294	1527
OTLBO	9.86026E−04	1.14239E−03	1.86989E−03	2.08322E−04	100%	1881	895
GOTLBO$_{Jr=0}$	9.92035E−04	1.27228E−03	2.05132E−03	2.69790E−04	97%	NA	NA
GOTLBO$_{Jr=0.1}$	**9.87442E−04**	1.33488E−03	1.98244E−03	2.99407E−04	**100%**	3204	2421
GOTLBO$_{Jr=0.2}$	9.92927E−04	1.36512E−03	1.93535E−03	2.61387E−04	**100%**	3230	2334
GOTLBO$_{Jr=0.3}$	9.87692E−04	1.38062E−03	2.09366E−03	3.28572E−04	**100%**	3230	2334
GOTLBO$_{Jr=0.4}$	9.91682E−04	**1.31202E−03**	**1.77688E−03**	**2.38120E−04**	**100%**	**2739**	**1621**
GOTLBO$_{Jr=0.5}$	9.91682E−04	**1.31202E−03**	**1.77688E−03**	**2.38120E−04**	**100%**	3318	2075
GOTLBO$_{Jr=0.6}$	9.89083E−04	1.32761E−03	1.96823E−03	2.80337E−04	**100%**	3318	2075
GOTLBO$_{Jr=0.7}$	1.01079E−03	1.33742E−03	2.01852E−03	3.14779E−04	97%	NA	NA
GOTLBO$_{Jr=0.8}$	1.01079E−03	1.33742E−03	2.01852E−03	3.14779E−04	93%	NA	NA
GOTLBO$_{Jr=0.9}$	9.95436E−04	1.47957E−03	2.40575E−03	3.50812E−04	90%	NA	NA
GOTLBO$_{Jr=1.0}$	1.00684E−03	1.50696E−03	2.14982E−03	3.44799E−04	90%	NA	NA

The best results obtained by the five EAs are highlighted with shade. The best results obtained by GOTLBO with specific jumping rate Jr are highlighted with bold.

Table 8
Comparison of GOTLBO with four EAs for the double diode model.

Algorithm	RMSE				SR	ANFES	
	Min	Mean	Max	SD		Mean	SD
jDE	9.84154E−04	1.01293E−03	1.35460E−03	7.54583E−05	100%	3967	2352
CLPSO	1.00720E−03	1.37731E−03	2.07598E−03	2.89687E−04	97%	NA	NA
TLBO	9.96598E−04	1.35292E−03	2.31028E−03	2.57136E−04	97%	NA	NA
OTLBO	9.84070E−04	1.23906E−03	1.78645E−03	2.53451E−04	100%	5228	4138
$GOTLBO_{Jr=0}$	**9.83177E−04**	1.24360E−03	1.78774E−03	2.09115E−04	**100%**	6198	4576
$GOTLBO_{Jr=0.1}$	9.86210E−04	1.25547E−03	1.91105E−03	2.61927E−04	**100%**	6150	5132
$GOTLBO_{Jr=0.2}$	9.87365E−04	1.38591E−03	3.38840E−03	5.09549E−04	90%	NA	NA
$GOTLBO_{Jr=0.3}$	9.86645E−04	1.25344E−03	2.08921E−03	2.63692E−04	97%	NA	NA
$GOTLBO_{Jr=0.4}$	9.84483E−04	**1.21075E−03**	**1.62545E−03**	**1.73575E−04**	**100%**	**5226**	**2984**
$GOTLBO_{Jr=0.5}$	9.84483E−04	**1.21075E−03**	**1.62545E−03**	**1.73575E−04**	**100%**	7168	4926
$GOTLBO_{Jr=0.6}$	9.89291E−04	1.27595E−03	1.84934E−03	2.27464E−04	**100%**	6257	3974
$GOTLBO_{Jr=0.7}$	9.91644E−04	1.25798E−03	1.84738E−03	2.25080E−04	**100%**	6221	3760
$GOTLBO_{Jr=0.8}$	9.93008E−04	1.24347E−03	1.94132E−03	2.27739E−04	**100%**	6028	4100
$GOTLBO_{Jr=0.9}$	9.89896E−04	1.36039E−03	2.51450E−03	2.96539E−04	97%	NA	NA
$GOTLBO_{Jr=1.0}$	9.94041E−04	1.45316E−03	2.68617E−03	4.14327E−04	87%	NA	NA

The best results obtained by the five EAs are highlighted with shade. The best results obtained by GOTLBO with specific jumping rate *Jr* are highlighted with bold.

Table 9
Comparison of GOTLBO with parameter extraction techniques in literature for the single diode model.

Algorithm	GA	CPSO	PS	SA	IGHS	ABSO	Rcr-IJADE	STLBO	GOTLBO
Iph(A)	0.7619	0.7607	0.7617	0.762	0.7608	0.7608	0.760776	0.76078	0.760780
Isd(μA)	0.8087	0.4	0.998	0.4798	0.3435	0.30623	0.323021	0.32302	0.331552
Rs(Ω)	0.0299	0.0354	0.0313	0.0345	0.0361	0.03659	0.036377	0.03638	0.036265
Rsh(Ω)	42.3729	59.012	64.1026	43.1034	53.2845	52.2903	53.718526	53.7187	54.115426
a	1.5751	1.5033	1.6	1.5172	1.4874	1.47878	1.481184	1.48114	1.483820
RMSE	1.9080E−02	1.3900E−03	1.4940E−02	1.9000E−02	9.9306E−04	9.9124E−04	9.8602E−04	9.8602E−04	9.87442E−04
NFES	NA	45000	NA	NA	150000	150000	10000	50000	10000

Table 10
Comparison of GOTLBO with parameter extraction techniques in literature for the double diode model.

Algorithm	PS	SA	IGHS	ABSO	Rcr-IJADE	STLBO	GOTLBO
Iph(A)	0.7602	0.7623	0.7608	0.76077	0.760781	0.760780	0.760752
Isd1(μA)	0.9889	0.4767	0.9731	0.26713	0.225974	0.225660	0.800195
Rs(Ω)	0.032	0.0345	0.0369	0.03657	0.03674	0.036740	0.036783
Rsh(Ω)	81.3008	43.1034	53.8368	54.6219	55.485443	55.492000	56.075304
a1	1.6	1.5172	1.9213	1.46512	1.451017	1.450850	1.999973
Isd2(μA)	0.0001	0.01	0.1679	0.38191	0.749347	0.752170	0.220462
a2	1.192	2	1.4281	1.98152	2	2.000000	1.448974
RMSE	1.5180E−02	1.6640E−02	9.8635E−04	9.8344E−04	9.8248E−04	9.8248E−04	9.83177E−04
NFES	NA	NA	150,000	150,000	20,000	50,000	20,000

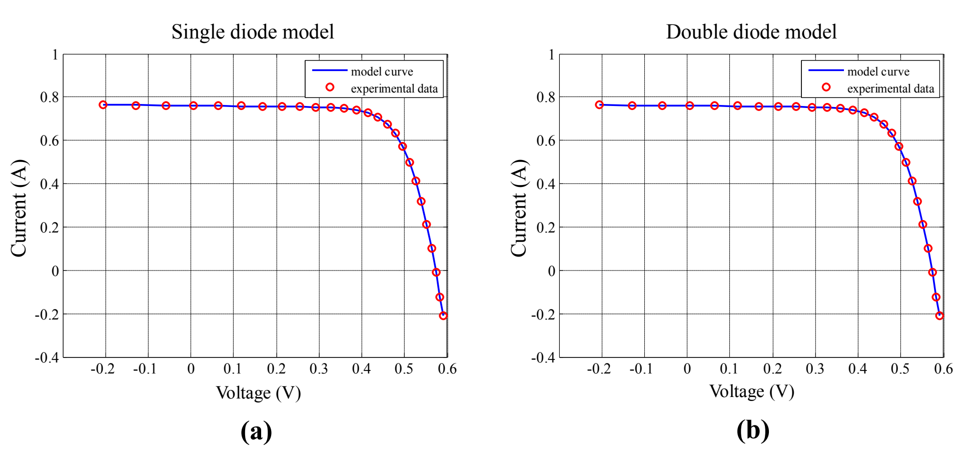

Fig. 6. Comparison on the I–V characteristics between the experimental data and simulated data obtained by GOTLBO for (a) the single diode model (b) the double diode model.

Acknowledgments

The authors would like to acknowledge the supports by the Research Talents Startup Foundation of Jiangsu University (Grant No. 15JDG139), the National Natural Science Foundation of China (Grant No. 51422702), and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

The authors would like to thank Dr. J. Brest, Dr. P. N. Suganthan, and Dr. W. Gong for their kindly providing the source codes of jDE, CLPSO and Rcr-IJADE.

References

[1] Oliva D, Cuevas E, Pajares G. Parameter identification of solar cells using artificial bee colony optimization. Energy 2014;72:93–102.
[2] Askarzadeh A, Rezazadeh A. Artificial bee swarm optimization algorithm for parameters identification of solar cell models. Appl Energy 2013;102:943–9.
[3] Ishaque K, Salam Z, Taheri H, Shamsudin A. A critical evaluation of EA computational methods for photovoltaic cell parameter extraction based on two diode model. Sol Energy 2011;85(9):1768–79.
[4] Brano VL, Orioli A, Ciulla G, Di Gangi A. An improved five-parameter model for photovoltaic modules. Sol Energy Mater Sol Cells 2010;94(8):1358–70.
[5] Han W, Wang H-H, Chen L. Parameters identification for photovoltaic module based on an improved artificial fish swarm algorithm. Sci World J 2014:2014.
[6] Easwarakhanthan T, Bottin J, Bouhouch I, Boutrit C. Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers. Int J Sol Energy 1986;4(1):1–12.
[7] Ortiz-Conde A, Sánchez FJG, Muci J. New method to extract the model parameters of solar cells from the explicit analytic solutions of their illuminated I–V characteristics. Sol Energy Mater Sol Cells 2006;90(3):352–61.
[8] Chan D, Phillips J, Phang J. A comparative study of extraction methods for solar cell model parameters. Solid-State Electron 1986;29(3):329–37.
[9] Niu Q, Zhang L, Li K. A biogeography-based optimization algorithm with mutation strategies for model parameter estimation of solar and fuel cells. Energy Convers Manag 2014;86:1173–85.
[10] Jervase JA, Bourdoucen H, Al-Lawati A. Solar cell parameter extraction using genetic algorithms. Meas Sci Technol 2001;12(11):1922.
[11] Zagrouba M, Sellami A, Bouaïcha M, Ksouri M. Identification of PV solar cells and modules parameters using the genetic algorithms: application to maximum power extraction. Sol Energy 2010;84(5):860–6.
[12] AlRashidi M, AlHajri M, El-Naggar K, Al-Othman A. A new estimation approach for determining the I–V characteristics of solar cells. Sol Energy 2011;85(7):1543–50.
[13] Ye M, Wang X, Xu Y. Parameter extraction of solar cells using particle swarm optimization. J Appl Phys 2009;105(9):094502.
[14] Wei H, Cong J, Lingyun X, Deyun S. Extracting solar cell model parameters based on chaos particle swarm algorithm. Conference Extracting solar cell model parameters based on chaos particle swarm algorithm. IEEE, p. 398–402.
[15] Ishaque K, Salam Z. An improved modeling method to determine the model parameters of photovoltaic (PV) modules using differential evolution (DE). Sol Energy 2011;85(9):2349–59.
[16] Jiang LL, Maskell DL, Patra JC. Parameter estimation of solar cells and modules using an improved adaptive differential evolution algorithm. Appl Energy 2013;112:185–93.
[17] Gong W, Cai Z. Parameter extraction of solar cell models using repaired adaptive differential evolution. Sol Energy 2013;94:209–20.
[18] Ishaque K, Salam Z, Mekhilef S, Shamsudin A. Parameter extraction of solar photovoltaic modules using penalty-based differential evolution. Appl Energy 2012;99:297–308.
[19] El-Naggar K, AlRashidi M, AlHajri M, Al-Othman A. Simulated Annealing algorithm for photovoltaic parameters identification. Sol Energy 2012;86(1):266–74.
[20] AlHajri M, El-Naggar K, AlRashidi M, Al-Othman A. Optimal extraction of solar cell parameters using pattern search. Renew Energy 2012;44:238–45.
[21] Askarzadeh A, Rezazadeh A. Parameter identification for solar cell models using harmony search-based algorithms. Sol Energy 2012;86(11):3241–9.
[22] Niu Q, Zhang H, Li K. An improved TLBO with elite strategy for parameters identification of PEM fuel cell and solar cell models. Int J Hydrogen Energy 2014;39(8):3837–54.
[23] Hasanien HM. Shuffled frog leaping algorithm for photovoltaic model identification. Sustain Energy, IEEE Trans 2015;6(2):509–15.
[24] Ma J, Ting T, Man KL, Zhang N, Guan S-U, Wong PW. Parameter estimation of photovoltaic models via cuckoo search. J Appl Math 2013:2013.
[25] Alam D, Yousri D, Eteiba M. Flower pollination algorithm based solar PV parameter estimation. Energy Convers Manag 2015;101:410–22.
[26] Rao RV, Savsani VJ, Vakharia D. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer Aided Des 2011;43(3):303–15.
[27] Rao R, Savsani V, Vakharia D. Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci 2012;183(1):1–15.
[28] Rao R, Savsani V, Balic J. Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems. Eng Optim 2012;44(12):1447–62.
[29] Baykasoğlu A, Hamzadayi A, Köse SY. Testing the performance of teaching–learning based optimization (TLBO) algorithm on combinatorial problems: flow shop and job shop scheduling cases. Inf Sci 2014;276:204–18.
[30] Toğan V. Design of planar steel frames using teaching–learning based optimization. Eng Struct 2012;34:225–32.
[31] Roy PK. Teaching learning based optimization for short-term hydrothermal scheduling problem considering valve point effect and prohibited discharge constraint. Int J Electr Power & Energy Syst 2013;53:10–9.
[32] Tizhoosh HR. Opposition-based learning: a new scheme for machine intelligence. Conference Opposition-based learning: a new scheme for machine intelligence. IEEE, p. 695–701.
[33] Rahnamayan S, Tizhoosh HR, Salama MM. Opposition versus randomness in soft computing techniques. Appl Soft Comput 2008;8(2):906–18.
[34] Rahnamayan S, Tizhoosh HR, Salama M. Opposition-based differential evolution. Evol Comput IEEE Trans 2008;12(1):64–79.
[35] Wu Z, Ni Z, Zhang C, Gu L. Opposition based comprehensive learning particle swarm optimization. Conference Opposition based comprehensive learning particle swarm optimization, vol. 1. IEEE, p. 1013–1019.
[36] El-Abd M. Generalized opposition-based artificial bee colony algorithm. Conference Generalized opposition-based artificial bee colony algorithm. IEEE, p. 1–4.
[37] Wang H, Wu Z, Rahnamayan S, Liu Y, Ventresca M. Enhancing particle swarm optimization using generalized opposition-based learning. Inf Sci 2011;181(20):4699–714.
[38] Kennedy J. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. Conference Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, vol. 3. IEEE.
[39] Brest J, Greiner S, Bošković B, Mernik M, Zumer V. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. Evol Comput IEEE Trans 2006;10(6):646–57.
[40] Liang JJ, Qin AK, Suganthan PN, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. Evol Comput IEEE Trans 2006;10(3):281–95.
[41] Roy PK, Paul C, Sultana S. Oppositional teaching learning based optimization approach for combined heat and power dispatch. Int J Electr Power & Energy Syst 2014;57:392–403.
[42] Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL report. 2005. p. 2005005.