Renewable Energy 134 (2019) 1129–1147

Contents lists available at ScienceDirect

Renewable Energy

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

An advanced onlooker-ranking-based adaptive differential evolution to extract the parameters of solar cell models

Nipotepat Muangkote [a], Khamron Sunat [b, *], Sirapat Chiewchanwattana [b], Sirilak Kaiwinit [a]

[a] Department of Business Computer, Mahasarakham Business School, Mahasarakham University, Mahasarakham 44150, Thailand
[b] Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

ARTICLE INFO

Article history:
Received 30 May 2018
Received in revised form
1 August 2018
Accepted 5 September 2018
Available online 6 September 2018

Keywords:
Evolutionary algorithm (EA)
Swarm intelligence (SI)
Nature-inspired (NI)
Solar cell models
Parameters extraction
Renewable energy

ABSTRACT

Solar cells are one of the renewable energy sources that have been widely used. The parameters extraction plays an important role in the speed and accuracy of models designed for photovoltaic (PV) solar cells and modules. In recent years, the evolutionary algorithm (EA), swarm intelligence (SI), and other nature-inspired (NI) algorithms have been widely used for the parameters extraction of PV modules. This paper presents a new method by improving the existing R_{cr}-IJADE with an onlooker-ranking-based mutation scheme. This mutation scheme is an effective and efficient vectors selection mechanism for encountering the objective function containing a flat basin. The improved algorithm referred to as OR_{cr}-IJADE, it is quickly and accurately extracted the parameters of solar cell models. 18 solar cell models and PV modules from several manufacturers were used to validate the algorithm. Comparative studies among the different algorithms were conducted using current-voltage (*I-V*) data. The results of OR_{cr}-IJADE were compared with 31 state-of-the-art EA, SI, and NI algorithms. The results confirm the superiority of the proposed method, as the accuracy, the success rate and the convergence speed are better than the competitors. The proposed algorithm is useful in developing highly accurate solar PV models with less computational effort used.

1. Introduction

In the near future, renewable energy sources, such as solar energy, will become one of the most important and widely used sources of energy. The qualities that the solar energy offers are clean, pollution-free, noise-free and maintenance-free characteristics, and the fact that it can be used any place in the world. Therefore, solar energy is able to intrigue most people's interests in today's world. Photovoltaic (PV) power systems directly convert solar energy (photons obtained from the sunlight) into electricity, and have been receiving significant attention in recent years because it has several promising advantages such as renewability, ease of installation, ease of maintenance and long-term economic prospects [1,2,3,4]. A modeling of the PV module involves the formulation of the non-linear, multi-variable, and multi-modal problem [1,3,5,6]. The enormous amount of works has been proposed to describe the current-voltage (*I-V*) relationship in solar cells over the past few years. However, the single and double diode models have been widely used for describing the non-linear *I-V* relationship [3,6,7]. The main parameters including the photo-generated current, saturation current, series resistance, shunt resistance, and diode ideality factor, are needed to be first determined to provide a precise modeling. It is important to determine these parameters due to the *I*–*V* curve and the information provided by the model parameters can be used for solar cell performance evaluation [8]. The problem is extracting the optimal parameters so that the model can produce are good fit with the experimental data.

Several parameters extraction techniques have been proposed to estimate the parameters of solar cells. There are two possible approaches to extract the parameters of solar cell models: the first one is the analytical approach, and the second one is the numerical extraction approach. As for the traditional extraction techniques, because of the difficulty in regard to using the parameters extraction analytical technique, the numerical extraction approach has

* Corresponding author.
E-mail addresses: mnipot@gmail.com (N. Muangkote), khamron_sunat@yahoo.com (K. Sunat), sunkra@kku.ac.th (S. Chiewchanwattana), sirilak.k@acc.msu.ac.th (S. Kaiwinit).

https://doi.org/10.1016/j.renene.2018.09.017
0960-1481/

received more interest from the researchers. The methods based on using swarm intelligence (SI), Evolutionary algorithm (EA), or nature-inspired (NI) algorithms have gained considerable attention [2,3]. The example applications of SI, EA, and NI approaches for extracting the solar cell parameter are those such as particle swarm optimization (PSO) [8–13], differential evolution (DE) [2,5,14–16], pattern search (PS) [1,17], genetic algorithm (GA) [18–20], an improved adaptive differential evolution (R_{cr}-IJADE) [3], teaching–learning-based optimization (TLBO) [21], and artificial bee colony optimization (ABC) [22].

Among several EA based algorithms, the differential evolution (DE) algorithm has been extensively used for the solar cell parameters extraction. The original DE was proposed in 1995 [23], and it is known for its ability to express several advantages, process rapid convergence (which is mostly found in the advanced version of DE), provides accurate results, performs well in exploration, requires few control parameters and be very simple in numerical computation. However, the original DE has some drawbacks such as its slowness in exploitation, sensitivity in the parameter setting, and difficulty in selecting a proper mutation strategy for a specific problem. Therefore, to relieve some of these defects, several advanced DE variants have been proposed in the literature. So far, the DE variants that have been proposed are: 1) a self-adaptation of the control parameters, such as jDE [24], JADE [25,26], and R_{cr}-IJADE [3]; and 2) an ensemble of different mutation operators was also introduced, such as SaDE [27], CoDE [28], EPSDE [29], and OXDE [30]. A rich review of some of state-of-the-art DE variants from 1995 to 2016 can be found in Das et al. [31,32].

Among the above references, the R_{cr}-IJADE is one of a very interesting DE variant. It has been outstandingly employed in determining the values of solar cell parameters [3]. Nevertheless, it was found that there are some rooms to improve the R_{cr}-IJADE algorithm by accelerating its convergence speed. A reason for choosing the R_{cr}-IJADE algorithm is because its performance in solving the solar cell parameters extraction problem has surpassed what has been proposed in the literature. However, to the best of our knowledge, the ranking-based mutation operator of the R_{cr}-IJADE method not only has too much exploitation ability, which consequently may lead to premature convergence but also lacks of an efficient mechanism to balance between the exploration and exploitation ability. To overcome these defects of the ranking-based operator, an approach for balancing the exploration and exploitation ability is being considered. The O(β)R operator was proposed by Charansiriphaisan et al. [33], it has been initially used to improve the performance of the original DE in solving the multilevel image thresholding problem. It is an effective and efficient vectors selection mechanism for enhancing the ordinary DE when the objective function contains a flat basin. Accordingly, it has a potential that can be further used to encourage the R_{cr}-IJADE algorithm. Concerning the aforementioned, the onlooker-ranking-based mutation operator (O(β)R) can encourage the R_{cr}-IJADE for an exploration and exploitation balancing, a convergence speed accelerating, and better final solutions of the solar cell parameters extraction achieving.

In this paper, a synergy of O(β)R into the R_{cr}-IJADE is presented and named as OR_{cr}-IJADE, and is then applied to extract the parameters of solar cell models. Since the given β value has an effect on the proposed algorithm in compromising between exploitation and exploration abilities. Therefore, a proper value of the β parameter in OR_{cr}-IJADE is firstly analyzed. According to the No-Free-Lunch (NFL) theorem for optimization [34,35], there is no optimization algorithm can outperform any other algorithm in general. In order to achieve a significant conclusion and biased avoidance, OR_{cr}-IJADE and 31 state-of-the-art algorithms were conducted to compare their abilities in solar cells parameters extraction.

The rest of the paper is organized as follows. In section 2, the survey SI, EA, and NI based parameters extraction of solar cell models are briefly presented. In section 3, the problem formulation and optimization process are briefly described. In section 4, the behavior of the solar cell parameters extraction problem is briefly analyzed. The basic idea of the JADE algorithm and crossover rate repair, the ranking-based operator, and the onlooker-ranking-based mutation operator are briefly described in section 5. Section 6, an advanced differential evolution (OR_{cr}-IJADE) is proposed. Section 7, the experiments, results, and the analysis are presented in this section. Finally, the conclusions and future work are pointed out in section 8.

2. Review of SI, EA, and NI based parameters extraction of solar cell models

There are many researchers concentrating on the parameters extraction of solar cells. A critical review of 34 methods developed to extract the parameters of solar cell problem over the past 35 years can be found in Cotfas et al. [36]. The accuracy of the extracted parameters highly depends on the choice of the fitting algorithm, especially on the initial values of the algorithm. The genetic algorithm (GA) and the particle swarm optimization (PSO) can improve the accuracy of the solar cell parameters without the very good choice of the values of the initial parameters as the final results were very good and the errors were very small in comparison to other methods. Accordingly, the several SI, EA, and other NI based algorithms that were developed to extract the parameters of a solar cell are mainly reviewed in this paper. In addition, the other SI, EA, and NI algorithms having a potential in the parameters determination of a solar cell are also reviewed.

The first group of algorithms is the SI, EA, and NI that have been applied to solving this problem. The SI, EA, and NI algorithms require an objective function to evaluate the difference between the actual values and the calculated values from the model. The objective function is to be minimized. There were several meta-heuristic techniques developed to extract the parameter of a solar cell. In 2011, the PS [17] was applied for estimating the parameters of the solar cell and the PV module. In 2012, AlHajri et al. [1] presented an application of the PS technique for extracting the parameters of different solar cell models, El-Naggar et al. [7] applied Simulated annealing (SA) for photovoltaic parameters identification, Askarzadeh and Rezazadeh [37] presented two variants of harmony search (HS) algorithm (the grouping-based global harmony search (GGHS) and innovative global harmony search (IGHS)) for specifying the parameters of single and double diode solar cell models. In the half past decades, DE variants had widely proposed to solve the particular parameters extraction of a solar cell model, i.e., Ishaque and Salam [15], and Ishaque et al. [5] in the year 2011, Ishaque et al. [2] in the year 2012. In 2013, Gong and Cai [3] proposed an improved version of JADE with a crossover rate repairing technique and ranking-based mutation operator (R_{cr}-IJADE) for a fast and accurate parameters extraction of solar cell models. The jDE, SaDE, CoDE, DEGL and JADE were applied to extract the parameters and were compared to the R_{cr}-IJADE. Other meta-heuristics, such as the family of ABC variants, bacterial foraging algorithm (BFA) that based on mimicry the foraging strategy of *Escherichia coli* bacteria present in the human intestine, bird mating optimization (BMO), and other meta-heuristic algorithms were also used by researchers. In 2013, artificial bee swarm optimization (ABSO) was proposed to identify the optimal parameters of solar cells by Askarzadeh and Rezazadeh [6], Rajasekar et al. [4] applied a BFA to model solar PV characteristics accurately, Hachana et al. [38] applied a hybrid ABC with DE (ABC-DE) for parameter identification of the photovoltaic cell/module, Askarzadeh and Rezazadeh [39,40] proposed and applied BMO to identify the parameters of the proton exchange membrane fuel cell model and parameters of solar cell models. In 2014, Oliva et al. [22] utilized the ordinary ABC for determining the parameters of a solar cell, Patel et al. [21] proposed the TLBO for the solar cell parameters extraction problem. In 2015, Khanna et al. [13] proposed a three-diode model and then applied the traditional PSO to estimate the parameters of the proposed solar cell model. In 2016, Muhsen et al. [41] proposed a new DE variant with a new formula for the mutation scaling factor and crossover rate adjustment to extract the parameters of a single diode model. In 2018, Gao et al. [42] proposed an improved shuffled complex evolution (ISCE) algorithm for fast and accurate parameters extraction of solar cell models.

Over the past two decades, the optimization algorithms are emerging to solve several other fields of optimization problems. Also, numerous SI, EA, and NI have been intensively developed and widely used in various applications. Therefore, the second group of algorithms is the other SI, EA, and NI algorithms that have not been yet developed for the solar cell parameters extraction problem, but they have been invented and have capably solved several other fields of optimization problems. The selected SI, EA, and NI algorithms are: CMA-ES [43], QPSO [44], CLPSO [45], GL-25 [46], WQPSO [47], GSA [48], SPSO2011 [49], ACiD [50], EPSDE [29], OXDE [30], WCA [51], DNLPSO [52], *Fl*ABCps [53], DMP-PSO [54], *Si*PABC_*Sf* [55], CoBiDE [56], GWO [57], LCA [58], DE-EIG [59], SFS [60], HeCoS [61], ALO [62], and SL-PSO [63]. Therefore, we will apply these algorithms for the solar cell parameters extraction and will compare with the proposed OR_{cr}-IJADE. Hereafter, a brief review of them is described.

In 2001, Hansen and Ostermeier [43] proposed the covariance matrix adaptation evolution strategy (CMA-ES), the algorithm based on the concepts of *derandomization* and *cumulation*. In 2004, Sun et al. [44] proposed a quantum-behaved PSO (QPSO) which based on the principles of quantum theory combined with the PSO algorithm. In 2006, Liang et al. [45] proposed a comprehensive learning PSO (CLPSO). During these same years, the GL-25 was proposed by Garcia-Martinez et al. [46], which is a GA variant that works on a hybrid real-coded genetic algorithm combining with global and local search. In 2008, the improved version of QPSO, named "weighted QPSO" (WQPSO), was proposed by Xi et al. [47], which works on a weighted principle meaning "best position" according to fitness values of the particle. In 2009, a gravitational search algorithm (GSA) was firstly proposed by Rashedi et al. [48], which works based on the principle of the law of gravity and mass interactions. In 2011, standard particle swarm optimization (SPSO2011) was proposed by Clerc [49]. Loshchilov et al. [50] proposed an adaptive coordinate descent (ACiD), which uses adaptive encoding (AE) coupled with adaptive coordinate descent (CD) that is used in the CMA-ES algorithm. The DE variant ensembling of mutation strategies and control parameters with the DE (EPSDE) was proposed by Mallipeddi et al. [29]. In 2012, OXDE was proposed by Wang et al. [30], which works based on using an orthogonal design into crossover operators for improving the performance of the ordinary DE. Eskandar et al. [51] proposed a new nature-inspired optimization technique called a water cycle algorithm (WCA), the algorithm based on the principle of the water cycle process, based on the observation of how rivers and streams flow to the sea. Nasir et al. [52] proposed a dynamic neighborhood learning based particle swarm optimizer (DNLPSO), it is an improved version of CLPSO aiming to enhance the diversity of the swarm. In 2013, Das et al. [53] proposed a novel variant of ABC family; fitness learning-based ABC with proximity stimuli (*Fl*ABCps), which based on a hybridization of fitness learning mechanism with a weighted selection scheme and proximity-based stimuli. In 2014, Kundu et al. [54] proposed an improved PSO with difference mean based perturbation (DMP-PSO), which based on the principle of dimensional mean based perturbation strategy, a simple aging guideline, and a set of nonlinearly time-varying acceleration coefficients. Das et al. [55] proposed a spatially informative perturbation-based ABC with saccadic flight (*Si*PABC_*Sf*), it is a novel in ABC family which inspired from the physical interpretation of the optic flow of information in honeybees. Wang et al. [56] proposed a novel DE variant called CoBiDE which works on the principle of incorporating the covariance matrix learning and the bimodal distribution parameter setting into DE. Mirjalili et al. [57] proposed a new meta-heuristic called grey wolf optimizer (GWO) which was inspired by the hunting behavior of grey wolves in nature. It yielded very competitive results compared to some well-known meta-heuristics. Kashan [58] mimicked a sport league competition and proposed a league championship algorithm (LCA). In 2015, another advanced DE variant DE-EIG was proposed by Guo and Yang [59] which is worked by utilizing eigenvectors of the covariance matrix of individual solutions into the crossover operator, which makes the crossover rotationally invariant. Salimi [60] proposed a stochastic fractal search (SFS), is based on the principle of the mathematics concept called the fractal. Ding et al. [61] proposed a novel method called a heterogeneous cuckoo search algorithm (HeCoS) and applied to an intelligent Takagi-Sugeno Modeling (iTaSuM) for identifying the structure and parameters of T-S fuzzy system. Mirjalili [62] proposed a novel nature-inspired algorithm named ant lion optimizer (ALO), it mimics the hunting mechanism of antlions in nature. One of a famous PSOs, a social learning particle swarm optimization algorithm (SL-PSO) that was proposed by Cheng and Jin [63], is based on the principle of social learning mechanisms incorporated into PSO. Actually, there are numerous invented SI, EA, and NI in the literature that is not mention in this paper yet. In this section, because these algorithms are considered selective for implementation and comparison with our proposed method, hence only the mentioned algorithms are reviewed and compared. These algorithms come from different families, they are advanced, and their performance is reasonably effective and can be considered as state-of-the-art approaches. That is the reason why we select them.

3. Parameters extraction problem formulation

Mathematical modeling represents the electrical characteristics of the solar cell models and the PV module. Then, a definition of the objective function is needed for the parameters extraction of different solar cell models using optimization techniques. This mathematical modeling and the objective function definition are explained as follows:

3.1. Solar cell models

Several equivalent circuit models have been developed to describe the *I-V* characteristic of solar cells in the literature. However, the double diode model and the single diode model are commonly used in practice. Not only the two models but also a photovoltaic module are briefly discussed in the following subsections.

3.1.1. Double diode model

The solar cell is ideally modeled as a current source connected in parallel with a rectifying diode. In the double diode model, the output current of the solar cell can be written as follows [64]:

$$I_L = I_{ph} - I_{d1} - I_{d2} - I_{sh} \tag{1}$$

where I_L is the cell output current, I_{ph} denotes the photo-generated current, I_{d1} is the first diode current, I_{d2} is the second diode currents, and I_{sh} represents the shunt resistor current. The equivalent circuit for this model is illustrated in Fig. 1(a).

Shockley diode equation is used to appropriately model the solar cell. The two diode currents I_{d1} and I_{d2} from Eq. (1) can be formulated as follows:

$$I_{d1} = I_{sd1}\left(\exp\left(\frac{V_L + I_L R_s}{a_1 V_t}\right) - 1\right) \tag{2}$$

$$I_{d2} = I_{sd2}\left(\exp\left(\frac{V_L + I_L R_s}{a_2 V_t}\right) - 1\right) \tag{3}$$

The shunt resistor current I_{sh} is formulated as

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

where V_L is the cell output voltage, I_{sd1} is the diffusion current, I_{sd2} is the saturation current, R_s is the series resistance, R_{sh} denotes the shunt resistance, and a_1 and a_2 denote the diffusion and recombination of diode ideality factors, respectively. The junction thermal voltage V_t in Eqs. (2) and (3) is formulated as:

$$V_t = \frac{kT}{q} \tag{5}$$

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

Seven parameters to be extracted for the double diode model are I_{ph}, I_{sd1}, I_{sd2}, R_s, R_{sh}, a_1, and a_2.

3.1.2. Single diode model

For the case of single diode model, I_{d1} and I_{d2} were combined together under the introduction of a non-physical diode ideality factor a. The single diode model has been widely used in literature because it offers a good compromise between simplicity and accuracy [3,6,17,65]. The equivalent circuit of this model is depicted in Fig. 1(b). In this model, the output current of the cell is calculated as follows:

$$I_L = I_{ph} - I_{sd}\left(\exp\left(\frac{V_L + I_L R_s}{a V_t}\right) - 1\right) - \frac{V_L + I_L R_s}{R_{sh}}. \tag{6}$$

Five parameters to be extracted for the single diode model are I_{ph}, I_{sd}, R_s, R_{sh}, and a.

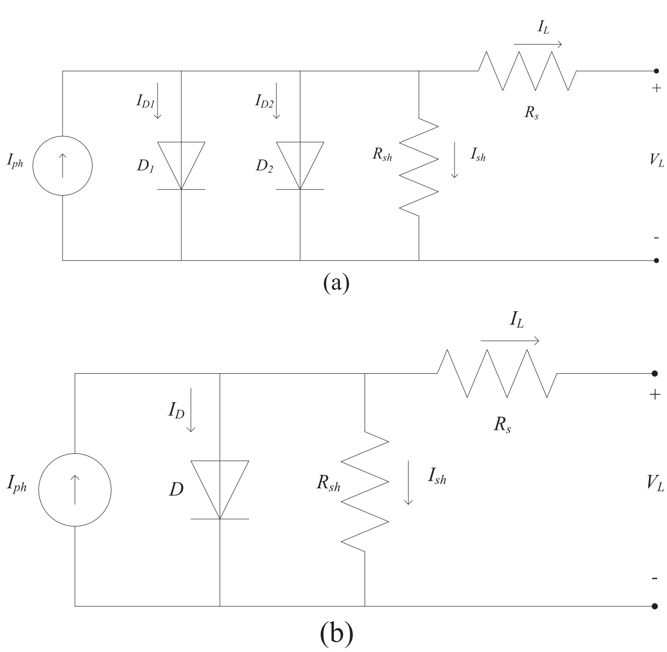

Fig. 1. An equivalent electrical circuit. (a) The double diode model of solar cell; (b) The single diode model of solar cell.

3.1.3. Photovoltaic module

A PV module consists of N_s connected cells in series per string and is modeled as a single diode model. The output current I of the PV module is given as [12]:

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

Since PV module is modeled as a single diode model, five parameters to be extracted for the PV module are I_{ph}, I_{sd}, R_s, R_{sh}, and a.

3.2. Optimization process as the parameters extraction process

The parameters of different solar cell models from the *I-V* data can be extracted using optimization techniques. Before proceeding to the optimization stage, firstly, the objective function is needed to be defined. The output current of a cell in the double diode model and the single diode model is rewritten in the homogeneous form as follows:

For the double diode model

$$f(V_L, I_L, \mathbf{x}) = I_{ph} - I_{sd1}\left(\exp\left(\frac{V_L + I_L R_s}{a_1 V_t}\right) - 1\right) - I_{sd2}\left(\exp\left(\frac{V_L + I_L R_s}{a_2 V_t}\right) - 1\right) - \frac{V_L + I_L R_s}{R_{sh}} - I_L \tag{8}$$

$$\mathbf{x} = \left\{I_{ph}, I_{sd1}, I_{sd2}, R_s, R_{sh}, a_1, a_2\right\}. \tag{9}$$

For the single diode model

$$f(V_L, I_L, \mathbf{x}) = I_{ph} - I_{sd}\left(\exp\left(\frac{V_L + I_L R_s}{a V_t}\right) - 1\right) - \frac{V_L + I_L R_s}{R_{sh}} - I_L \tag{10}$$

$$\mathbf{x} = \left\{I_{ph}, I_{sd}, R_s, R_{sh}, a\right\}. \tag{11}$$

For the parameters that need to be extracted, $\mathbf{x}$ represents the decision vector for the optimization techniques. For each parameter, it is bounded in the search space. Table 1 shows the lower and upper boundaries of each parameter of the double diode model, single diode model, and the PV module.

The root mean square error (*RMSE*) is used as the objective function [2,3,6,11], which is described as:

$$F(\mathbf{x}) = \sqrt{\frac{1}{N}\sum_{k=1}^{N} f_k(V_L, I_L, \mathbf{x})^2}, \tag{12}$$

where N is the number of experimental data.

Eventually, the problem solving of a solar cell parameters extraction is to minimize the objective function $F(\mathbf{x})$, in order to make the model better fit to the data.

Table 1
Ranges of the parameters of the solar cell models.

Parameter	Solar cell models							
	Models		PV module					
	Double and single diode models		PV module		Mono-crystalline: Shell SP70, and Shell Sq85		Thin film: Shell ST40	
	Lower bound	Upper bound	Lower bound	Upper bound	Lower bound	Upper bound	Lower bound	Upper bound
I_{ph}(A)	0	1	0	2	0	10	0	10
I_{sd} (μA)	0	1	0	50	0	50	0	50
R_s(Ω)	0	0.5	0	2	0.001	1	0.001	3
R_{sh}(Ω)	0	100	0	2000	50	200	50	300
a	1	2	1	50	0.5	2	0.5	2

4. Problem analysis

In this section, in order to better understand the behavior of the solar cell parameters extraction problem, this problem is merely analyzed. For the sake of limited space, the double diode model is selected to demonstrate because its parameters cover that of the single diode model and PV module. Several curves of the objective function of the double diode model with respect to two input parameters are visualized in Fig. 2. In Fig. 2(a) and (b), the three-dimensional spaces characterized by the objective function and its contour are projected with the relationship between pairs of (a_1, a_2) parameters, respectively. It can be seen that the minimum objective value can be obtained somewhere in the input space generated from $a_1 \times a_2$, and the minimum objective function is near 2. As can be seen from Fig. 2(c) and (d), we found that the objective function value is slightly changed with respect to I_{ph}, whereas the objective function value is changed a lot if the I_{sd1} changed. That means I_{sd1} has a strong influence on the objective function but I_{ph} has not. The minimum objective value can be obtained when I_{sd1} varies from 0.5 to 0.6. By carefully looking at the Fig. 2(e), (f), (m), (n), (s), and (t), we found that I_{sd2} has a strong influence on the objective function. Fig. 2(g) and (h), we found that I_{ph} has a strong influence on the objective function, a proper I_{ph} can be found somewhere near to 0.

As exemplified in Fig. 2, it is worth mentioning that the parameters extraction of solar cell models is a non-linear and multi-variable model, and might be a multi-modal model. Therefore, many traditional extraction techniques (i.e., SI, EA, and NI based extraction techniques) may not efficient in accurately extract the parameters of this kind of problem. An example of difficult cases of finding the minimum objective value of the double diode model is depicted in Fig. 2(k), (l), (o), and (p), of the double diode model. There is a large flat surface on the top so that some algorithms can rarely produce the minimum objective value (see section 7.3.3). However, the synergy of O(β)R into the R_{cr}-IJADE should effectively overcome this kind of problem.

5. The related ideas of DE, JADE and R_{cr}-IJADE

In this section, the original DE, JADE, and the R_{cr}-IJADE algorithms are briefly described. JADE is an advanced DE variant proposed by Zhang and Sanderson [26], it has a promising performance compared to the original DE and it is a base algorithm for several advanced DE variants and R_{cr}-IJADE algorithm, which was proposed by Gong and Cai [3] for a solar cell parameters extraction, is one of its offspring.

5.1. The idea of JADE and crossover rate repair

DE is a very simple EA for the numerical optimization proposed by Storn and Price [23]. There were DE variants that concerned on *CR* and *F* adaptation [24,27,66,67]. An advanced DE variant, JADE, was proposed by Zhang and Sanderson [25,26]. In JADE, in addition to the adaptation of *F* and *CR*, a new mutation strategy "DE/current-to-*p*best" is implemented with an optional archive. The adaptation techniques proposed in JADE are briefly discussed within the DE framework as follows:

5.1.1. Initialization

The initial population (at generation $t=0$) $\vec{x}_{i,0} = (x_{i,1,0}, x_{i,2,0}, \ldots, x_{i,D,0})$, $i = 1, \ldots, \mu$ is randomly generated so that each dimension (D) is in the search space. $\vec{x}_{i,j,0}$ (at generation $t=0$) can be initialized as follows:

$$x_{i,j,0} = x_{j,\min} + \mathrm{rand}_{i,j}[0,1) \cdot (x_{j,\max} - x_{j,\min}) \tag{13}$$

where $x_{j,\min}$ and $x_{j,\max}$ are the jth minimum and maximum bounds, index $i = 1, \ldots, \mu$ is the ith solution of the population, $j = 1, \ldots, D$, μ is the population size, and $\mathrm{rand}_{i,j}[0, 1)$ is a uniformly distributed random number between 0 and 1.

5.1.2. Mutation operators

At the generation t, the mutation operator is applied to generate the mutant *vector* $\vec{v}_{i,t}$ for each *target vector* $\vec{x}_{i,t}$ in the current population. The mutation is a change or perturbation with a random element in the context of the evolutionary computing paradigm [32]. The role of mutation operator is to construct the mutant vector through a specific mutation scheme based on adding differences between randomly selected elements of the population to another element. Zhang and Sanderson [26] proposed the "DE/current-to-best/1" mutation strategy for JADE. In 2013, Gong and Cai [74] proposed a rank-DE, which the vectors selection is based on the rank of vectors, for balancing the exploration and exploitation abilities of the ordinary DE. Inspired by a survival of species in nature, good species always contain good information, and more likely to be selected to propagate the offspring [74]. R_{cr}-IJADE, which used the quadratic probabilistic assignment model for each vector, was proposed for parameters extraction of a solar cell by Ref. [3].

Charansiriphaisan et al. [33] found that rank-DE has premature convergence when applied to a multi-level image thresholding problem. A higher threshold number is a harder problem. The rank-DE cannot solve the task when the threshold number is greater than 12, i.e., a premature convergence occurred. The algorithm needs more exploration ability to solve this problem. Hence, to alleviate this drawback, Charansiriphaisan et al. proposed O(β)R-DE with the onlooker-ranking-based mutation operator (O(β)R).

The mutation vectors of JADE are generated in the following manner:

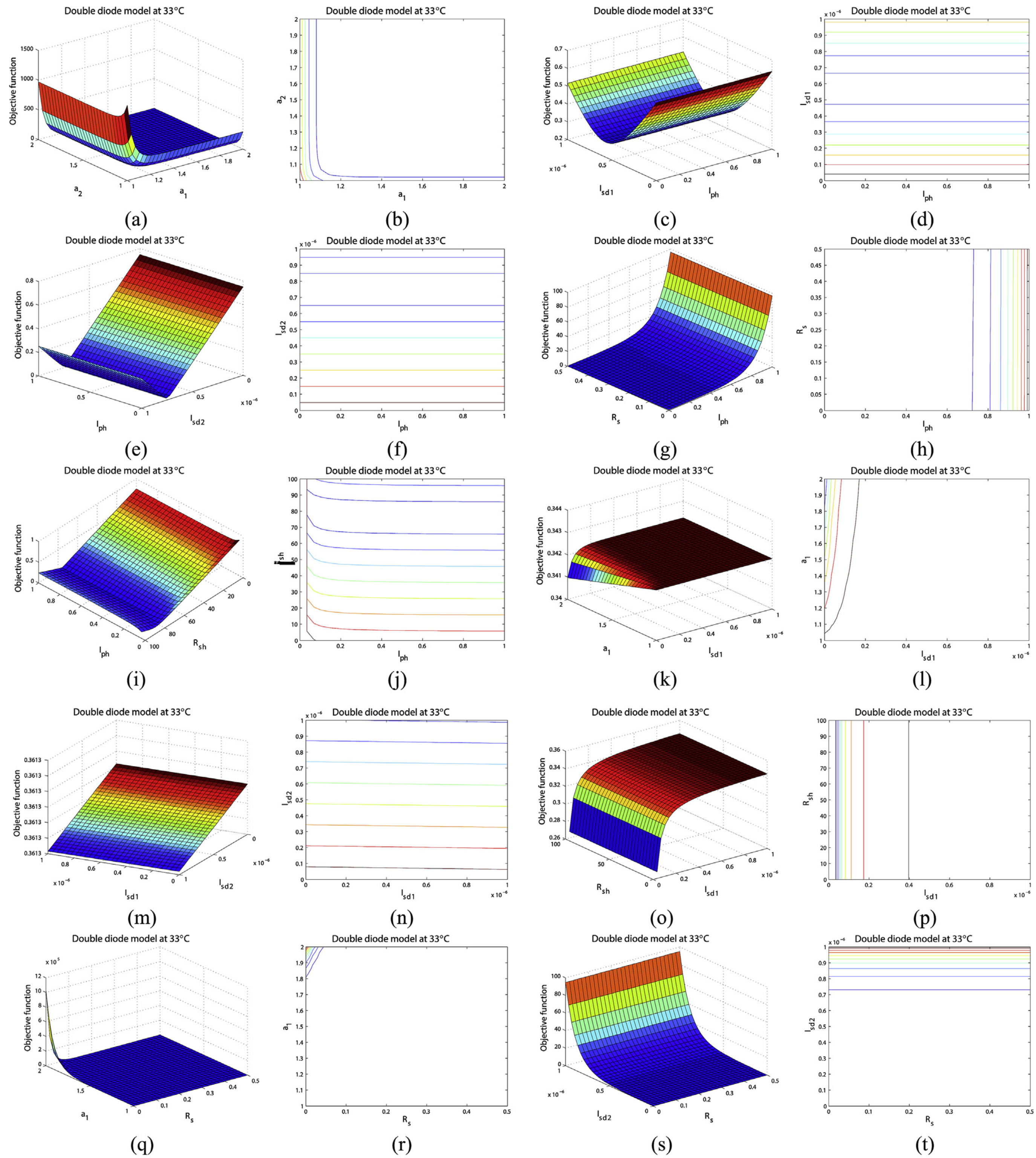

Fig. 2. Visualization of the objective function and its contour of double diode model with the pair difference of its parameters. (a), (b): pair of (a_1, a_2); (c), (d): pair of (I_{ph}, I_{sd1}); (e), (f): pair of (I_{sd2}, I_{ph}); (g), (h): pair of (I_{ph}, R_s); (i), (j): pair of (R_{sh}, I_{ph}); (k), (l): pair of (I_{sd1}, a_1); (m), (n): pair of (I_{sd1}, I_{sd2}); (o), (p): pair of (I_{sd1}, R_{sh}); (q), (r): pair of (R_s, a_1); (s), (t): pair of (R_s, I_{sd2}).

(1) DE/current-to-*p*best/1 (without archive):

$$\vec{v}_{i,t} = \vec{x}_{i,t} + F \cdot \left(\vec{x}^{p}_{best,t} - \vec{x}_{i,t} \right) + F \cdot \left(\vec{x}_{r_1,t} - \vec{x}_{r_2,t} \right) \quad (14)$$

(2) DE/current-to-*p*best/1 (with archive):

$$\vec{v}_{i,t} = \vec{x}_{i,t} + F \cdot \left(\vec{x}^{p}_{best,t} - \vec{x}_{i,t} \right) + F \cdot \left(\vec{x}_{r_1,t} - \vec{x}'_{r_2,t} \right) \quad (15)$$

where F is the mutation scaling factor, the dot operator $(\cdot)$ appeared in the above equations is the scalar multiplication, $\vec{x}_{i,t}$, $\vec{x}_{r_1,t}$, $\vec{x}_{r_2,t}$, and $\vec{x}^{p}_{best,t}$ are selected from the current population $\mathbf{P}$, while $\vec{x}'_{r_2,t}$ is randomly chosen from the union of the current population and the archive, $\mathbf{P} \cup \mathbf{A}$, and $r_1 \neq r_2 \neq i$. In Eq. (15), an archive $\mathbf{A}$ is used to store the inferior solutions. $\vec{x}^{p}_{best,t}$ refers to the $p$best solution, which is randomly chosen from the top of 100p% solutions with $p \in (0, 1]$.

To preserve the population diversity, at generation t, the mutation factor F_i of individual target vector is independently calculated as:

$$F_i = \mathrm{randc}_i(\mu_F, 0.1) \tag{16}$$

and either $F_i > 1.0$ then truncated to be 1.0 or $F_i \le 0$ regenerated. $\mathrm{randc}_i(\mu_F,\ 0.1)$ denotes a random number generated according to the Cauchy distribution with location parameter μ_F and scale parameter 0.1. The location parameter μ_F is updated as follows:

$$\mu_F = (1-c)\cdot\mu_F + c\cdot \mathrm{mean}_L(S_F) \tag{17}$$

where S_F is the set of all successful F_i at generation t; and $\mathrm{mean}_L(\cdot)$ is the Lehmer mean:

$$\mathrm{mean}_L(S_F) = \frac{\sum_{i=1}^{|S_F|} F_i^2}{\sum_{i=1}^{|S_F|} F_i}. \tag{18}$$

In order to minimize optimization, the information on good fitness of the population in DE is utilized. Firstly, the fitness of each vector $\vec{x}_{i,t}$ is sorted in ascending order (i.e., from best to worst). Then, the rank of the ith vector; R_i, is assigned, based on its sorted ordering as follows:

$$R_i = \mu - i, \quad i = 1, 2, ..., \mu. \tag{19}$$

Accordingly, the best vector in the current population will obtain the highest ranking [74].

The quadratic probabilistic assignment model used in $\mathrm{R_{cr}}$-IJADE, the selection probability pv_i of each vector is assigned as follows:

$$pv_i = \left(\frac{R_i}{\mu}\right)^2, \quad i = 1, 2, ..., \mu. \tag{20}$$

To make the premature convergence abate, the onlooker-ranking-based mutation operator for DE (O(β)R-DE), inspired by the onlooker phase of the ABC algorithm [68], was proposed in Ref. [33]. The pseudo-code of the O(β)R vector selection is shown in Algorithm 1.

Algorithm 1
The O(β)R vector selection.

```
Input: β, the target vector index i, and the last index of onlooker r1
Output: The selected vector indices r1, r2, r3
r1 = r1 + 1; if r1 > μ then r1 = 1; end if
while minprop(β) > p_r1 do //onlooker-like selection
  r1 = r1 + 1; if r1 > μ then r1 = 1; end if
end while
Randomly select r2 ∈ { 1, μ } //terminal vector index
while rand_j[0, 1) > p_r2 or r2 == i or r2 == r1 do
  Randomly select r2 ∈ { 1, μ}
end while
Randomly select r3 ∈ { 1, μ } //the other vector index
while r3 == r2 or r3 == r1 or r3 == i do
  Randomly select r3 ∈ { 1, μ }
end while
```

From Algorithm 1, the proper value of β parameter can tread-off the exploration and exploitation abilities. The function minprop(β) generalizes the algorithm in that its output depends on the β parameter and can be either a constant value of [0, 1] or a value generated from uniform random function rand[0, 1). The function minprop(β) can be defined as follows:

$$\mathrm{minprop}(\beta) = \begin{cases} \beta, & \text{if } \beta \text{ is a constant and } 0 \le \beta \le 1 \\ \mathrm{rand}[0,1), & \text{otherwise.} \end{cases} \tag{21}$$

If β is 0, it means a leading vector can be any vector in the population and the algorithm prefers exploration. If β is 1, it means a leading vector is the best vector that obtains the highest probability and the algorithm prefers exploitation. The proper β value is in [0, 1].

5.1.3. Crossover operators

In the DE family, the crossover operator plays an essential role to construct an offspring by mixing components of the current element and of that generated by mutation [69]. The CR is the crossover probability or crossover rate. Its role is to determine how the child inherits parameters form the mutant. It controls the number of components and which components are selected to mutate in each element of the current population. Hence, the new trial element can be constructed starting from the current and mutant elements by crossover operator. The impact the crossover operator and its parameter, the CR, has been analyzed. There are several studies focusing on the influence of crossover [69–71]. Recently, Gong et al. [72] presented a crossover rate repair technique to speed up the convergence rate by encouraging the exploitation ability.

Binomial crossover is the most commonly used crossover operators in the DE algorithm. It is utilized to generate the new trial vectors $\vec{u}_{i,t}$, which can be exchanged between $\vec{x}_{i,t}$ and $\vec{v}_{i,t}$. The trial vector $\vec{u}_{i,t} = (u_{i,1,t}, u_{i,2,t}, ..., u_{i,D,t})$ is created as follows:

$$u_{i,j,t} = \begin{cases} v_{i,j,t} & \text{if } (\mathrm{rand}_j[0,1) \le CR) \text{ or } j = j_{rand} \\ x_{i,j,t} & \text{otherwise} \end{cases} \tag{22}$$

where the crossover rate $CR \in (0,\ 1]$, the index j_{rand} is randomly chosen within $[1,\ D]$, and $\mathrm{rand}_j[0,\ 1)$ is a uniform random number within $[0,\ 1)$. The use of j_{rand} guarantees that at least one parameter of $\vec{u}_{i,t}$ is different from $\vec{x}_{i,t}$.

At each generation t, for each target vector the crossover probability CR_i is independently generated as follows:

$$CR_i = \mathrm{randn}_i(\mu_{CR}, 0.1) \tag{23}$$

and then truncated to the interval $[0,\ 1]$; where $\mathrm{randn}_i(\mu_{CR},\ 0.1)$ is a normal distribution with the mean value μ_{CR} and standard deviation 0.1. The mean μ_{CR} is initialized to be 0.5 and then updated as follows:

$$\mu_{CR} = (1-c)\cdot\mu_{CR} + c\cdot\mathrm{mean}_A(S_{CR}) \tag{24}$$

where c is a constant in $[0,\ 1]$, $\mathrm{mean}_A(\cdot)$ is the usual arithmetic mean, and S_{CR} denotes the set of all successful crossover rates CR_i at generation t.

Let $\mathbf{b}_i$ be a binary string generated for each target vector $\mathbf{x}_i$ as follows:

$$b_{i,j} = \begin{cases} 1, & \text{if} (\mathrm{rand}_j[0,1) < CR \text{ or } j == j_{rand}) \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

Then, the binomial crossover of ordinary DE in Eq. (22) can be reformulated as

$$u_{i,j} = b_{i,j} \cdot v_{i,j} + (1 - b_{i,j}) \cdot x_{i,j}, \tag{26}$$

where $i = 1, \ldots, \mu$ and $j = 1, \ldots, D$. According to Eqs. (25) and (26), the successful trial vector $\mathbf{u}_i$ is directly related to its binary string $\mathbf{b}_i$. The crossover rate is repaired by its corresponding binary string using the average number of components taken from the mutant. The repaired crossover rate is calculated as follows:

$$CR_i' = \frac{\sum_{j=1}^{D} b_{i,j}}{D}, \tag{27}$$

where $\mathbf{b}_i$ is the binary string calculated in Eq. (25) based on CR_i, $i = 1, \ldots, \mu$ and $j = 1, \ldots, D$. Therefore, the crossover rate is repaired. If the trial vector $\mathbf{u}_i$ is a successful vector, CR_i' will be stored in S_{CR} instead of storing CR_i.

5.1.4. Selection

Finally, for the selection step, for minimization, a greedy selection operation is used to determine whether the target $\vec{x}_{i,t}$ or the trial vector $\vec{u}_{i,t}$ survives to the next generation; $t+1$. For the trial vector $\vec{u}_{i,t}$, the selection operation selects the lower fitness value $f(\cdot)$ of each individual of $\vec{x}_{i,t}$ and $\vec{u}_{i,t}$. The selection operation is described as:

$$\vec{x}_{i,t+1} = \begin{cases} \vec{u}_{i,t}, & \text{if } f(\vec{u}_{i,t}) \le f(\vec{x}_{i,t}) \\ \vec{x}_{i,t}, & \text{otherwise.} \end{cases} \tag{28}$$

Therefore, if the objective of the new trial vector $f(\vec{u}_{i,t})$ is equal to or lower than the objective of the old trial vector $f(\vec{x}_{i,t})$ then $\vec{x}_{i,t+1}$ is set to $\vec{u}_{i,t}$. Otherwise, the old value $\vec{x}_{i,t}$, is retained.

6. The proposed OR$_{cr}$-IJADE algorithm

In this section, the proposed algorithm is an advanced onlooker-ranking-based mutation operator (O(β)R) which is synergized into the R$_{cr}$-IJADE, called OR$_{cr}$-IJADE for short. The excellent OR$_{cr}$-IJADE is then applied to extract the parameters of different solar cell models.

However, O(β)R-DE has too much exploration ability as the speed of O(β)R-DE was slower than the other.

6.1. The proposed OR$_{cr}$-IJADE

Hereafter, the procedures mentioned in previous-sections including the crossover rate repairing technique, the O(β)R mutation operator, and R$_{cr}$-IJADE algorithm are combined together to develop OR$_{cr}$-IJADE.

The "DE/rand-to-*p*best/1" mutation scheme of R$_{cr}$-IJADE was proposed by Gong and Cai [3] and it is formulated as:

$$\vec{v}_{i,t} = \vec{x}_{r_1,t} + F \cdot \left(\vec{x}^{\,p}_{best,t} - \vec{x}_{r_1,t}\right) + F \cdot \left(\vec{x}_{r_2,t} - \vec{x}_{r_3,t}\right) \tag{29}$$

where the indexes $r_1, r_2, r_3 \in \{1, 2, \ldots, \mu\}$ and $r_1 \ne r_2 \ne r_3 \ne i$. The vectors $\vec{x}_{r_1,t}$, $\vec{x}_{r_2,t}$ and $\vec{x}_{r_3,t}$ were chosen using rank-DE by Algorithm 2 in Ref. [74]. For the proposed OR$_{cr}$-IJADE algorithm, the "DE/onlooker-to-*p*best/1" mutation scheme is computed by Eq. (29) but the three indices selection is performed by Algorithm 1, i.e., r_1 is selected with onlooker-ranking-based, r_2 is selected with ranking-based, and r_3 is randomly selected. The pseudo-code of OR$_{cr}$-IJADE is shown in Algorithm 2; while the different steps between R$_{cr}$-IJADE and OR$_{cr}$-IJADE are highlighted by "⇐."

For the termination criterion, this work used the maximum of number of function evaluations (*Max_NFEs*) as the termination criterion, where *NFEs* is the number of function evaluations. The maximal generations are unsuitable to use as the termination criterion; since at the same generation, the different algorithms are consumed different number of function evaluations (*NFEs*).

Algorithm 2
The OR$_{cr}$-IJADE algorithm.

Input: Three control parameters; population size μ, the maximum of number of function evaluations *Max_NFEs*, the β value. ⇐
Output: $\vec{x}_{best}$: the best solution in the final population
Set $\mu_{CR} = 0.5$, $\mu_F = 0.5$, $c = 0.1$, $p = 0.05$, $A = \phi$
Initialize the population randomly
Evaluate the objective function value of each solution in P_0
$NFEs = \mu$; $t = 1$
while $NFEs < Max\_NFEs$ **do**
  $S_{CR} = \phi$, $S_F = \phi$, $r_1 = 0$
  Sort the population from the best to the worst
  for $i = 1$ to μ **do**
    Generate CR_i and F_i with Eqs. (23) and (16), respectively
    Calculate the ranking R_i and the selection probability pv_i
  end for
  for $i = 1$ to μ **do**
    Select the vector indexes with onlooker-ranking-based select r_1, ranking-based select r_2 and randomly select r_3 according to Algorithm 1 ⇐
    Produce the mutant vector $\vec{v}_{i,t}$ with adaptive onlooker-ranking-based $\vec{v}_{i,t} = \vec{x}_{r_1,t} + F \cdot (\vec{x}^{\,p}_{best,t} - \vec{x}_{r_1,t}) + F \cdot (\vec{x}_{r_2,t} - \vec{x}_{r_3,t})$; "DE/onlooker-to-*p*best/1"
    Get the binary string $\boldsymbol{b}_i$ as shown in Eq. (25)
    Calculate the repaired crossover rate CR_i' with Eq. (27)
    for $j = 1$ to D **do**
      $u_{i,j} = b_{i,j} \cdot v_{i,j} + (1 - b_{i,j}) \cdot x_{i,j}$
    end for
    Evaluate the objective function value of $\boldsymbol{u}_i$
  end for
  for $i = 1$ to μ **do**
    if $\boldsymbol{u}_i$ is better than its parent $\vec{x}_i$ **then**
      Update the archive **A** with the inferior solution $\vec{x}_i$
      $\vec{x}_i = \boldsymbol{u}_i$
      $CR_i' \to S_{CR}$, $F_i \to S_F$
    end if
  end for
  Update the μ_F and μ_{CR} with Eqs. (17) and (24), respectively
  $NFEs = NFEs + \mu$; $t = t + 1$
end while
output the best solution $\vec{x}_{best}$ in the final population

6.2. Boundary-handling technique

The values of some generated mutation vector $\vec{v}_{i,t}$ may violate boundary constraints. To respond to this problem, we use the *reinitialization* method, which will uniformly reinitialize the violating components within the corresponding bounds [3]. It can be calculated as

$$v_{i,j,t} = x_{j,\min} + \text{rand}_{i,j}[0,1) \cdot (x_{j,\max} - x_{j,\min}) \tag{30}$$

where $v_{i,j,t}$ denote the jth components of the mutation vector $\vec{v}_{i,t}$, $x_{j,\min}$, and $x_{j,\max}$ are the jth minimum and maximum bounds, index $i = 1, \ldots, \mu$ is the ith solution of the population, μ is the population size, $j = 1, \ldots, D$, and $\text{rand}_{i,j}[0, 1)$ is a uniformly distributed random number between 0 and 1.

There is another technique for handling the bound. Zhang and Sanderson [26] set the violating components to be in the middle between the violated bounds and its corresponding component of the parent. This method is *bounce-back* boundary-handling technique [73]. If the optimal solution is located near the boundary, this method is suitable for handling the violating components.

7. Experimental studies and results

This section aims to investigate the ability of the proposed OR_{cr}-IJADE algorithm and the effects of the O(β)R operator when combined with the R_{cr}-IJADE algorithm. Hereafter, it is applied to the real world problem; especially, in the field of photovoltaic engineering. The OR_{cr}-IJADE algorithm will be evaluated for parameters extraction of different solar cell models and PV modules from several manufacturers; single diode, double diode, and different types of PV modules (mono-crystalline and thin-film). Accordingly, the synthetic current-voltage (*I-V*) data of a solar cell and PV module are used in the evaluation. Then, these problems are conducted for comparative study among different parameters extraction techniques. These data are obtained from Ref. [3] and are widely used literature [1,6,7,17] including a 57 mm diameter commercial (R.T.C France) silicon solar cell (at 33 °C) and a solar module (Photowatt-PWP 201, at 45 °C) in which 36 polycrystalline silicon cells are connected in series. Furthermore, the three simulated current-voltage (*I-V*) PV module datasets are generated, based on different technologies, that are Shell SP70, Shell SQ85, and Shell ST40 [12], which have also been investigated with varied temperatures at five different degrees: $T = -25$ °C, 0 °C, 25 °C, 50 °C, and 75 °C.

7.1. Experimental setup and parameter settings

For all experiments, the parameters are used as same as in their original article unless a change is mentioned for all parameters extraction problems of different solar cell models. Since the boundary-handling method has a significant influence on the performance of the algorithms; and for a fair in comparison between OR_{cr}-IJADE and R_{cr}-IJADE, the *reinitialization* method is used for both algorithms, as same as in Ref. [3]. However, for JADE, this research employs the commonly-used boundary-handling method as in its original article. In other words, the *bounce-back* [26] boundary-handling technique handles the boundary constraint when one of the decision variables is violating its boundary constraint. Meanwhile, other algorithms used the boundary-handling method as used in their original articles. The population size $\mu = 50$ is used for all algorithms and all experiments. For the parameters μ_{CR}, μ_F, c, and p used in JADE, R_{cr}-IJADE, and OR_{cr}-IJADE are the default values as appeared in their original literature. For the other competitors SI, EA, and other NI algorithms, all their parameters are as same as they were used in their original literature.

For fairness in comparison of each optimizer, the *Max_NFEs* is set to be 150,000 for all algorithms and all problems. The high number of *Max_NFEs* is set in order to ensure that at least one algorithm can produce the global optimum solution. For the statistical results comparisons, all algorithms are run for 100 independent runs. In each one of 100 runs, every algorithm starts from the same initial position to evaluate different algorithms in a similar way [3,72,74]. All methods are coded in MATLAB, run on a computer with a CPU of 3.4 GHz, 8 GB RAM, on Windows 7 64-bit Professional platform.

7.2. Performance evaluation criteria

The performance criteria are an important issue used in measuring the performance between different algorithms. These algorithms will be evaluated by the following performance criteria which are similar to that used in Ref. [3].

7.2.1. Root mean square error (RMSE)

This criterion is used as the objective function of the optimization techniques to measure the performance of the algorithm while measuring the quality of the fit between the actual data and the expected data. A lower *RMSE* is better.

7.2.2. Number of function evaluations ($NFEs_\varepsilon$)

The $NFEs_\varepsilon$ is used for recording the number of function evaluations in each run to search for the solution of satisfying the condition: $f(\vec{x}) - f(\vec{x}*) \le \varepsilon$, where $\vec{x}*$ is the known-optimal solution and $f(\vec{x}*) = 0$ in this work. While the objective function $(f(\vec{x}))$ has not satisfied the condition, the recording is continued; then the searching is stopped when the condition is satisfied and the $NFEs_\varepsilon$ number is reported. The ε is a very small positive value to be pre-given for different problems. It can be given in Table 2, where a lower $NFEs_\varepsilon$ is better.

7.2.3. Success rate (SR)

There is one crucial evaluation criterion in measuring the efficiency and reliability of the algorithm. The higher *SR* means higher reliability. It is defined as follows:

$$SR = \frac{(\text{number of successful runs})}{\text{total number of runs}}, \tag{31}$$

where the successful run is the run that reaches the solution satisfying the condition of $f(\vec{x}) - f(\vec{x}*) \le \varepsilon$ within *Max_NFEs*.

7.2.4. Acceleration rate (AR)

This criterion is very important in considering the ability of each algorithm. The *AR* is mainly used to compare the convergence speed between two algorithms [75,76,77,78]. It is defined as follows:

$$AR = \frac{\text{ANFEs}_{\varepsilon,A}/SR_A}{\text{ANFEs}_{\varepsilon,B}/SR_B}, \tag{32}$$

where $\text{ANFEs}_{\varepsilon,A}$ and SR_A are the average NFEs_ε and *SR* values of algorithm A, respectively. $AR < 1$ means algorithm B converges faster than algorithm A.

7.2.5. Convergence graphs

The convergence graphs show the *median* error performance, $f(\vec{x}) - f(\vec{x}*)$, of the total run with termination by the *Max_NFEs*. The graphs show in log10 scale of $f(\vec{x}) - f(\vec{x}*)$ vs. *NFEs* for each problem, where $\vec{x}$ is the best solution found.

Table 2
The pre-given ε value for all problems.

Problem	The ε values at different temperatures						
	−25 °C	0 °C	25 °C	33 °C	45 °C	50 °C	75 °C
Single diode model	–	–	–	10^{-3}	–	–	–
Double diode model	–	–	–	10^{-3}	–	–	–
PV module	–	–	–	–	10^{-2}	–	–
Shell SP70	10^{-9}	10^{-8}	10^{-8}	–	–	10^{-8}	10^{-8}
Shell SQ85	10^{-8}	10^{-9}	10^{-8}	–	–	10^{-9}	10^{-9}
Shell ST40	10^{-8}	10^{-7}	10^{-8}	–	–	10^{-7}	10^{-7}

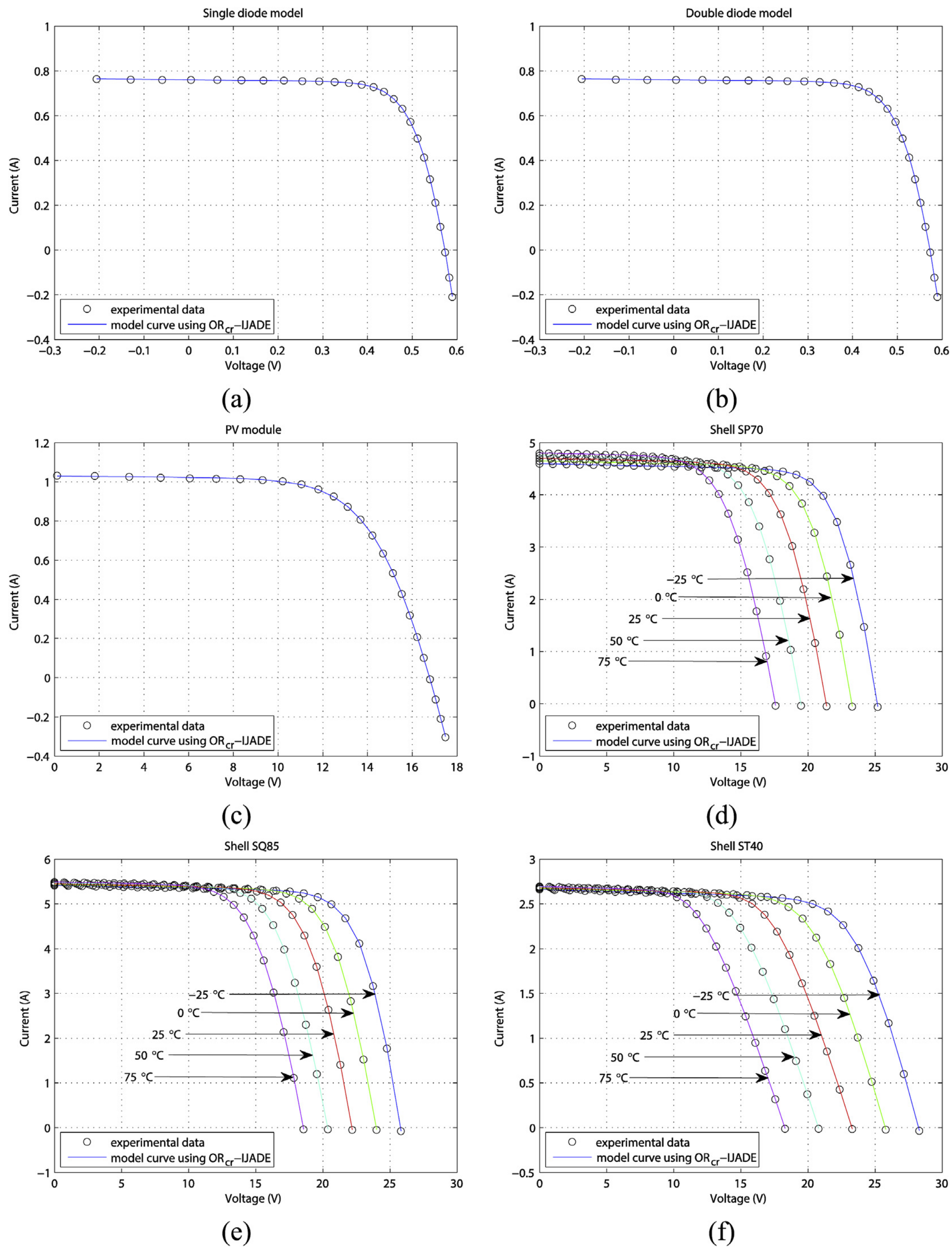

Fig. 3. The fitting result on the *I-V* characteristic of simulated data obtained by OR_{cr}-IJADE with $\beta = 0.1$ and experimental data. (a): single diode model; (b): double diode model; (c) PV module; (d): SP70; (e): Shell SQ85; (f): Shell ST40.

7.3. Experimental results

7.3.1. The effectiveness of β value in OR_{cr}-IJADE

The *I-V* characteristic produced by OR_{cr}-IJADE, both the experimental data and the model outcome, of single diode model, double diode model, PV module, Shell SP70, Shell SQ85, and Shell ST40 are shown in Fig. 3. As shown in Fig. 3, it is clear that the *I-V* characteristics generated by OR_{cr}-IJADE highly coincide with the experimental data in all solar cell models.

The parameter β in O(β)R operator plays a very important role in treading-off exploration and exploitation throughout the optimization process. The properly set β value can enhance the performance of the proposed algorithm. In this subsection, the influence of different β values in OR_{cr}-IJADE is analyzed and discussed. Firstly, the OR_{cr}-IJADE with varying β values are evaluated and then compared to the existing R_{cr}-IJADE algorithm. In order to find a proper β value, the *I-V* characteristic and the individual absolute error between the experimental data and model outcome

Table 3
Comparison of statistical results between R_{cr}-IJADE and OR_{cr}-IJADE with 50 of popsize, 150,000 NFEs, 100 runs, on different performance criteria for the single diode model. The lowest values of *RMSE*, $NFEs_{\varepsilon}$, and *AR*, and the highest values of *SR* are highlighted in **boldface**.

Algorithm	*RMSE*						$NFEs_{\varepsilon}(\varepsilon = 0.001)$		*SR*	*AR*
	Min	Median	Max	Mean	Std	Sig.	Mean	Std		
R_{cr}-IJADE	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	4.054469E-17		7679.97	2417.67	**1.00**	1.000
OR_{cr}-IJADE (0.0)	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	4.509089E-17	=	**5835.66**	**1671.71**	**1.00**	**0.760**
OR_{cr}-IJADE (0.1)	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	3.994841E-17	=	6299.54	1993.26	**1.00**	0.820
OR_{cr}-IJADE (0.2)	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	3.540291E-17	+	6652.96	2059.85	**1.00**	0.866
OR_{cr}-IJADE (0.3)	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**3.299984E-17**	=	6418.49	2288.72	**1.00**	0.836
OR_{cr}-IJADE (0.4)	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	3.793976E-17	=	6637.59	2329.88	**1.00**	0.864
OR_{cr}-IJADE (0.5)	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	3.801774E-17	=	6474.89	2511.68	**1.00**	0.843
OR_{cr}-IJADE (0.6)	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	3.967047E-17	=	7280.49	2348.10	**1.00**	0.948
OR_{cr}-IJADE (0.7)	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	3.971002E-17	=	7259.85	2506.66	**1.00**	0.945
OR_{cr}-IJADE (0.8)	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	4.014309E-17	+	7036.93	2645.27	**1.00**	0.916
OR_{cr}-IJADE (0.9)	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	4.036259E-17	=	6820.56	3192.41	**1.00**	0.888
OR_{cr}-IJADE (rand)	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	4.448812E-17	=	6607.29	2121.94	**1.00**	0.860

The number in parenthesis indicated the parameter β value.

Table 4
Comparison of the ranked *AR* results between OR_{cr}-IJADE and R_{cr}-IJADE with β-varying value in all solar cell models.

No.	*Solar cell models*	Ranks of acceleration rate (*AR*)											
		R_{cr}-IJADE	OR_{cr}-IJADE										
			$\beta = 0.0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.4$	$\beta = 0.5$	$\beta = 0.6$	$\beta = 0.7$	$\beta = 0.8$	$\beta = 0.9$	$\beta =$ rand
1	Single diode model	12	1	2	7	3	5	4	11	10	9	8	6
2	Double diode model	11	12	9	3	8	10	1	5	7	4	2	6
3	PV module	10	12	11	9	6	8	5	4	3	1	2	7
4	SP70(-25)	7	1	2	3	4	6	8	9	10	11	12	4
5	SP70(0)	7	1	2	3	4	6	8	9	10	11	12	4
6	SP70(25)	6	1	2	4	5	8	7	9	10	11	12	3
7	SP70(50)	8	1	2	4	6	5	7	9	10	11	12	3
8	SP70(75)	11	1	2	4	3	5	8	7	9	9	12	6
9	SQ85 (-25)	7	1	2	3	5	6	8	9	10	11	12	4
10	SQ85 (0)	7	1	2	3	5	6	8	9	10	11	12	4
11	SQ85 (25)	6	1	2	4	5	7	8	9	10	11	12	3
12	SQ85 (50)	8	1	2	3	5	6	7	9	10	11	12	4
13	SQ85 (75)	9	1	2	3	5	7	8	6	10	11	12	4
14	ST40 (-25)	7	1	3	2	4	6	8	9	10	11	12	5
15	ST40 (0)	7	1	2	3	5	6	8	9	10	11	12	4
16	ST40 (25)	6	1	2	3	5	7	8	9	10	11	12	4
17	ST40 (50)	7	1	2	3	5	6	8	9	10	11	12	4
18	ST40 (75)	7	1	2	3	5	6	8	9	10	11	12	4
Average rankings[a]		7.94	2.22	2.94	3.72	4.94	6.44	7.06	8.33	9.42	9.86	10.67	4.44

[a] Average rankings indicated that the average rankings of all algorithms by the Friedman test.

(individual absolute error: IAE) [1,3,7,17] of both OR_{cr}-IJADE and R_{cr}-IJADE must be obtained and compared. However, for the sake of space limitation, the experimental *I-V* data of the single diode model taken from Ref. [3] is shown as an example, the rest of results are in the supplementary data. In order to reflect the effect of β, R_{cr}-IJADE and OR_{cr}-IJADE with 11 different β values extract the parameters and return them to the single diode model to reconstruct the *I-V* characteristic and to compare with the experimental data. The min, median, max, mean, and the associated standard deviation (Std.) of *RMSE*, mean *NFEs* and the associated Std., the paired Wilcoxon signed-rank with $\alpha = 0.05$ compares the significance of *RMSE* values between R_{cr}-IJADE and OR_{cr}-IJADE with 11 different β values, the *SR*, and the *AR* compared with the convergence speed between R_{cr}-IJADE and OR_{cr}-IJADE, are recorded and tabulated in Table 3. The cases that are marked with "+" and "=," indicate that OR_{cr}-IJADE is significantly better than or similar to the R_{cr}-IJADE. The cases that $AR < 1$ indicate that OR_{cr}-IJADE is faster than R_{cr}-IJADE. From Table 3, both OR_{cr}-IJADE and R_{cr}-IJADE achieve the same accuracy, but OR_{cr}-IJADE is faster than R_{cr}-IJADE.

For the overall convergent rate, the ranks of *ARs* produced from R_{cr}-IJADE and 11 variations of OR_{cr}-IJADE conducted on 18 datasets are shown in Table 4. The average Friedman test of *ARs* are shown in the last row of Table 4. As seen from the table, the algorithms are ranked from best to worst in the following order: OR_{cr}-IJADE (0.0), OR_{cr}-IJADE (0.1), OR_{cr}-IJADE (0.2), OR_{cr}-IJADE (rand), OR_{cr}-IJADE (0.3), OR_{cr}-IJADE (0.4), OR_{cr}-IJADE (0.5), R_{cr}-IJADE, OR_{cr}-IJADE (0.6), OR_{cr}-IJADE (0.7), OR_{cr}-IJADE (0.8), and OR_{cr}-IJADE (0.9). OR_{cr}-IJADE with $\beta = 0.0$ to 0.5 and β with a random are faster than R_{cr}-IJADE, which we conclude based on the convergent rate.

The IAEs produced by OR_{cr}-IJADE with 11 different β values and R_{cr}-IJADE conducted on single diode model are reported in Table 5. The best of sum of IAE is highlighted in **boldface**. As can be seen from the table, the IAEs produced by OR_{cr}-IJADE with 11 different β values are lower or equal to that produced by R_{cr}-IJADE and the best IAE is produced by OR_{cr}-IJADE. For the other datasets, the conclusion does not change.

For the overall IAEs, the ranks of IAEs produced from R_{cr}-IJADE and 11 variations of OR_{cr}-IJADE conducted on 18 datasets are shown

Table 5
Curve fitting comparison results between OR_{cr}-IJADE and R_{cr}-IJADE with β-varying value of the extracted single diode model parameters. The lowest sums of IAE are highlighted in **boldface**.

Item	V_L(V)	I_L measured (A)	Individual absolute error (IAE)											
			R_{cr}-IJADE	OR_{cr}-IJADE										
				$\beta = 0.0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.4$	$\beta = 0.5$	$\beta = 0.6$	$\beta = 0.7$	$\beta = 0.8$	$\beta = 0.9$	$\beta = \text{rand}$
1	−0.2057	0.7640	0.00008770	0.00008770	0.00008770	0.00008770	0.00008770	0.00008770	0.00008770	0.00008770	0.00008770	0.00008770	0.00008770	0.00008770
2	−0.1291	0.7620	0.00066309	0.00066309	0.00066309	0.00066309	0.00066309	0.00066309	0.00066309	0.00066309	0.00066309	0.00066309	0.00066309	0.00066309
3	−0.0588	0.7605	0.00085531	0.00085531	0.00085531	0.00085531	0.00085531	0.00085531	0.00085531	0.00085531	0.00085531	0.00085531	0.00085531	0.00085531
4	0.0057	0.7605	0.00034601	0.00034601	0.00034601	0.00034601	0.00034601	0.00034601	0.00034601	0.00034601	0.00034601	0.00034601	0.00034601	0.00034601
5	0.0646	0.7600	0.00094479	0.00094479	0.00094479	0.00094479	0.00094479	0.00094479	0.00094479	0.00094479	0.00094479	0.00094479	0.00094479	0.00094479
6	0.1185	0.7590	0.00095765	0.00095765	0.00095765	0.00095765	0.00095765	0.00095765	0.00095765	0.00095765	0.00095766	0.00095765	0.00095765	0.00095765
7	0.1678	0.7570	0.00009165	0.00009165	0.00009165	0.00009165	0.00009165	0.00009165	0.00009165	0.00009165	0.00009165	0.00009165	0.00009165	0.00009165
8	0.2132	0.7570	0.00085864	0.00085864	0.00085864	0.00085864	0.00085864	0.00085864	0.00085864	0.00085864	0.00085864	0.00085864	0.00085864	0.00085864
9	0.2545	0.7555	0.00041313	0.00041313	0.00041313	0.00041313	0.00041313	0.00041313	0.00041313	0.00041313	0.00041313	0.00041313	0.00041313	0.00041313
10	0.2924	0.7540	0.00033612	0.00033612	0.00033612	0.00033612	0.00033612	0.00033612	0.00033612	0.00033612	0.00033612	0.00033612	0.00033612	0.00033612
11	0.3269	0.7505	0.00089097	0.00089097	0.00089097	0.00089097	0.00089097	0.00089097	0.00089097	0.00089097	0.00089097	0.00089097	0.00089097	0.00089097
12	0.3585	0.7465	0.00085385	0.00085385	0.00085385	0.00085385	0.00085385	0.00085385	0.00085385	0.00085385	0.00085385	0.00085385	0.00085385	0.00085385
13	0.3873	0.7385	0.00161722	0.00161722	0.00161722	0.00161722	0.00161722	0.00161722	0.00161722	0.00161722	0.00161722	0.00161722	0.00161722	0.00161722
14	0.4137	0.7280	0.00061777	0.00061778	0.00061777	0.00061777	0.00061777	0.00061778	0.00061778	0.00061778	0.00061778	0.00061778	0.00061777	0.00061777
15	0.4373	0.7065	0.00047265	0.00047265	0.00047265	0.00047265	0.00047265	0.00047265	0.00047265	0.00047265	0.00047265	0.00047265	0.00047265	0.00047265
16	0.4590	0.6755	0.00021985	0.00021985	0.00021985	0.00021985	0.00021985	0.00021985	0.00021985	0.00021985	0.00021985	0.00021985	0.00021985	0.00021985
17	0.4784	0.6320	0.00124173	0.00124173	0.00124173	0.00124173	0.00124173	0.00124173	0.00124173	0.00124173	0.00124173	0.00124173	0.00124173	0.00124173
18	0.4960	0.5730	0.00107164	0.00107164	0.00107164	0.00107164	0.00107164	0.00107164	0.00107164	0.00107164	0.00107164	0.00107164	0.00107164	0.00107164
19	0.5119	0.4990	0.00060702	0.00060702	0.00060702	0.00060702	0.00060702	0.00060702	0.00060702	0.00060702	0.00060702	0.00060702	0.00060702	0.00060702
20	0.5265	0.4130	0.00064879	0.00064879	0.00064879	0.00064879	0.00064879	0.00064879	0.00064879	0.00064879	0.00064879	0.00064879	0.00064879	0.00064879
21	0.5398	0.3165	0.00101011	0.00101011	0.00101011	0.00101011	0.00101011	0.00101011	0.00101011	0.00101011	0.00101011	0.00101011	0.00101011	0.00101011
22	0.5521	0.2120	0.00015494	0.00015494	0.00015494	0.00015494	0.00015494	0.00015494	0.00015494	0.00015494	0.00015494	0.00015494	0.00015494	0.00015494
23	0.5633	0.1035	0.00124869	0.00124869	0.00124869	0.00124869	0.00124869	0.00124869	0.00124869	0.00124869	0.00124869	0.00124869	0.00124869	0.00124869
24	0.5736	−0.0100	0.00128246	0.00128246	0.00128246	0.00128246	0.00128246	0.00128246	0.00128246	0.00128246	0.00128246	0.00128246	0.00128246	0.00128246
25	0.5833	−0.1230	0.00250741	0.00250741	0.00250741	0.00250741	0.00250741	0.00250741	0.00250741	0.00250741	0.00250741	0.00250741	0.00250741	0.00250741
26	0.5900	−0.2100	0.00152767	0.00152767	0.00152767	0.00152767	0.00152767	0.00152767	0.00152767	0.00152767	0.00152767	0.00152767	0.00152767	0.00152767
Sum of IAE			0.02152687	0.02152687	0.02152687	0.02152687	**0.02152686**	0.02152687	0.02152687	0.02152687	0.02152687	0.02152687	**0.02152686**	0.02152687
Rank			2	2	2	2	1	2	2	2	2	2	1	2

Table 6
Comparison of the ranked IAE results between OR$_{cr}$-IJADE and R$_{cr}$-IJADE with β-varying value in all solar cell models.

Problem No	*Solar cell models*	R$_{cr}$-IJADE	Ranks of individual absolute error (IAE)										
			OR-JADE										
			$\beta = 0.0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.4$	$\beta = 0.5$	$\beta = 0.6$	$\beta = 0.7$	$\beta = 0.8$	$\beta = 0.9$	$\beta =$ rand
1	Single diode model	2	2	2	2	1	2	2	2	2	2	1	2
2	Double diode model	1	12	1	1	1	1	8	8	1	8	8	1
3	PV module	5	5	1	5	5	5	1	5	1	1	5	5
4	SP70(-25)	6	8	1	1	1	1	6	9	10	12	11	1
5	SP70(0)	1	8	2	2	2	2	7	9	10	12	11	2
6	SP70(25)	2	9	2	2	1	2	7	8	10	11	12	2
7	SP70(50)	7	2	4	4	8	2	1	9	10	11	12	4
8	SP70(75)	4	1	3	4	4	9	11	9	2	4	12	4
9	SQ85 (-25)	1	7	2	2	2	2	8	9	10	12	11	2
10	SQ85 (0)	7	8	1	1	1	1	1	9	11	10	12	1
11	SQ85 (25)	5	8	2	1	5	2	5	9	10	11	12	2
12	SQ85 (50)	3	9	1	1	3	3	3	8	10	11	12	3
13	SQ85 (75)	3	1	6	6	3	6	6	6	3	2	12	6
14	ST40 (-25)	6	8	1	1	1	6	1	11	12	10	9	1
15	ST40 (0)	1	9	3	3	3	3	2	8	10	12	11	3
16	ST40 (25)	4	9	4	1	1	1	7	8	10	12	11	4
17	ST40 (50)	4	9	5	5	1	1	7	8	10	11	12	1
18	ST40 (75)	5	9	1	5	1	1	1	8	10	11	12	5
Average rankings[a]		3.72	6.89	2.33	2.61	2.44	2.78	4.67	7.94	7.89	9.06	10.33	2.72
		6	8	1	3	2	5	7	10	9	11	12	4

[a] Average rankings indicated that the average rankings of all algorithms by the Friedman test.

Table 7
Comparison of different parameters extraction techniques for the single diode model. The lowest *RMSE* as well as the extracted parameters are highlighted in **boldface**.

Algorithm	Item (extracted from the best solution)					*RMSE*
	I_{ph}(A)	I_{sd} (μA)	R_s(Ω)	R_{sh}(Ω)	a	
GL25	0.760757	0.327267	0.036321	54.122450	1.482500	9.863852E-04
GSA	0.759768	0.647281	0.031976	67.732890	1.554636	2.887716E-03
TLBO	0.760778	0.323025	0.036377	53.701910	1.481185	9.860229E-04
WCA	**0.760776**	**0.323002**	**0.036377**	**53.716310**	**1.481178**	**9.860219E-04**
SFS	**0.760776**	**0.323021**	**0.036377**	**53.718520**	**1.481184**	**9.860219E-04**[a]
HeCoS	**0.760776**	**0.323021**	**0.036377**	**53.718530**	**1.481184**	**9.860219E-04**[a]
ACiD	**0.760776**	**0.323021**	**0.036377**	**53.718520**	**1.481184**	**9.860219E-04**
GWO	0.760005	0.374088	0.035914	73.657170	1.496049	1.172427E-03
LCA	0.760956	0.353942	0.035678	47.588620	1.490812	1.246812E-03
ALO	0.790658	6.260380	0.003928	13.519090	1.860577	2.966183E-02
PSO	**0.760776**	**0.323021**	**0.036377**	**53.718520**	**1.481184**	**9.860219E-04**
CLPSO	0.761130	0.277432	0.036991	49.453160	1.465785	1.295102E-03
QPSO	0.760768	0.324482	0.036358	53.878690	1.481637	9.860747E-04
WQPSO	0.760763	0.325555	0.036347	54.050590	1.481970	9.861785E-04
SPSO2011	0.761228	0.999962	0.031278	84.029930	1.604657	2.511754E-03
DNLPSO	0.760772	0.325615	0.036347	53.949820	1.481989	9.861433E-04
DMP-PSO	0.759643	0.200105	0.038462	57.374680	1.434152	1.622726E-03
SL-PSO	0.760694	0.387861	0.035642	59.263180	1.499830	1.046227E-03
ABC	0.761164	0.354267	0.035862	47.582160	1.490845	1.181361E-03
FIABCps	0.760406	0.290579	0.036987	55.632350	1.470483	1.076627E-03
SiPABC_Sf	0.760803	0.321689	0.036443	54.462590	1.480736	9.901651E-04
CMAES	0.759615	0.437174	0.035397	92.654740	1.511998	1.370234E-03
SaDE	**0.760776**	**0.323021**	**0.036377**	**53.718530**	**1.481184**	**9.860219E-04**[a]
EPSDE	**0.760776**	**0.323021**	**0.036377**	**53.718520**	**1.481184**	**9.860219E-04**[a]
CoDE	**0.760776**	**0.323021**	**0.036377**	**53.718530**	**1.481184**	**9.860219E-04**[a]
OXDE	**0.760776**	**0.323021**	**0.036377**	**53.718520**	**1.481184**	**9.860219E-04**[a]
CoBiDE	**0.760776**	**0.323021**	**0.036377**	**53.718530**	**1.481184**	**9.860219E-04**[a]
jDE	**0.760776**	**0.323021**	**0.036377**	**53.718520**	**1.481184**	**9.860219E-04**[a]
JADE	**0.760776**	**0.323021**	**0.036377**	**53.718530**	**1.481184**	**9.860219E-04**
DE-EIG	**0.760776**	**0.323021**	**0.036377**	**53.718520**	**1.481184**	**9.860219E-04**[a]
R$_{cr}$-IJADE	**0.760776**	**0.323021**	**0.036377**	**53.718530**	**1.481184**	**9.860219E-04**[a]
OR$_{cr}$-IJADE	**0.760776**	**0.323021**	**0.036377**	**53.718520**	**1.481184**	**9.860219E-04**[a]

[a] The algorithms that reach the global solution as their successful rate *SRs* are 1.

in Table 6. The average Friedman test of IAEs are shown in the last row of Table 6. As can be seen from the table, the algorithms are ranked from best to worst in the following order: OR$_{cr}$-IJADE (0.1), OR$_{cr}$-IJADE (0.3), OR$_{cr}$-IJADE (0.2), OR$_{cr}$-IJADE (rand), OR$_{cr}$-IJADE (0.4), R$_{cr}$-IJADE, OR$_{cr}$-IJADE (0.5), OR$_{cr}$-IJADE (0.0), OR$_{cr}$-IJADE (0.7), OR$_{cr}$-IJADE (0.6), OR$_{cr}$-IJADE (0.8), and OR$_{cr}$-IJADE (0.9). OR$_{cr}$-IJADE with $\beta = 0.1$ to 0.4 and β with a random are faster than R$_{cr}$-IJADE, which we conclude based on the accuracy of extracted parameters.

Table 8
Comparison of statistical results among OR_{cr}-IJADE and other state-of-the-art algorithms with 50 of popsize, 150,000 NFEs, 100 runs, on different performance criteria for the single diode model; "NA" means not available. The lowest values of *RMSE*, $NFEs_{\varepsilon,}$ and *AR*, and the highest values of *SR* are highlighted in **boldface**.

Algorithm	*RMSE*						$NFEs_{\varepsilon}(\varepsilon = 0.001)$		*SR*	*AR*
	Min	Median	Max	Mean	Std.	Sig.	Mean	Std		
GL25	9.863852E-04	9.967828E-04	1.050642E-03	1.000315E-03	1.245600E-05	+	127964.70	12892.18	0.59	37.166
GSA	1.841970E-03	5.108860E-03	1.661451E-02	5.884575E-03	3.379586E-03	+	NA	NA	NA	NA
TLBO	9.860229E-04	1.015458E-03	1.560309E-03	1.066598E-03	1.060966E-04	+	74592.72	44427.87	0.32	39.944
WCA	**9.860219E-04**	9.861413E-04	1.954508E-02	1.414883E-03	2.023818E-03	+	20584.73	14265.79	0.67	5.265
SFS	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	2.554108E-17	+	42376.13	10469.00	**1.00**	7.262
HeCoS	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	1.619619E-17	+	28280.96	7204.43	**1.00**	4.846
ACiD	**9.860219E-04**	1.397284E-03	3.001682E-01	3.814806E-02	7.433567E-02	+	6911.83	6731.88	0.23	5.150
GWO	1.172427E-03	4.818938E-03	9.412183E-03	4.737658E-03	2.091550E-03	+	NA	NA	NA	NA
LCA	9.886414E-04	1.153185E-03	1.676825E-03	1.169749E-03	1.254779E-04	+	81364.50	17687.00	0.04	348.566
ALO	2.966183E-02	1.687750E-01	2.228614E-01	1.555190E-01	7.242538E-02	+	NA	NA	NA	NA
PSO	**9.860219E-04**	**9.860219E-04**	2.292347E-03	1.057431E-03	2.663888E-04	+	122196.70	14598.69	0.91	23.011
CLPSO	9.866995E-04	9.966351E-04	1.188693E-03	1.005572E-03	2.866523E-05	+	106978.20	24082.75	0.59	31.071
QPSO	9.860747E-04	1.058286E-03	1.328805E-03	1.071335E-03	6.970058E-05	+	60106.74	40198.94	0.19	54.210
WQPSO	9.861785E-04	1.065170E-03	1.235879E-03	1.069339E-03	5.824776E-05	+	76079.23	40640.17	0.13	100.284
SPSO2011	2.511754E-03	3.453440E-03	2.170608E-02	4.343140E-03	2.835247E-03	+	NA	NA	NA	NA
DNLPSO	9.861433E-04	1.532432E-03	3.179929E-03	1.583429E-03	4.687482E-04	+	56341.25	29112.14	0.04	241.366
DMP-PSO	1.603598E-03	3.148979E-03	2.475801E-02	4.735140E-03	4.297515E-03	+	NA	NA	NA	NA
SL-PSO	1.046227E-03	1.859234E-03	2.390558E-03	1.877306E-03	3.493188E-04	+	NA	NA	NA	NA
ABC	9.886004E-04	1.051587E-03	1.471266E-03	1.082023E-03	9.206701E-05	+	65861.56	35722.48	0.09	125.401
FlABCps	1.076627E-03	1.664415E-03	2.988219E-03	1.685219E-03	3.198904E-04	+	148834.40	8567.12	NA	NA
SiPABC_Sf	9.901651E-04	1.302008E-03	1.912942E-03	1.323229E-03	2.158335E-04	+	113234.00	39896.02	0.03	646.793
CMAES	1.370234E-03	3.481143E-03	2.100034E-02	4.638617E-03	3.387165E-03	+	NA	NA	NA	NA
SaDE	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**1.521251E-17**	+	19311.27	5665.78	**1.00**	3.309
EPSDE	**9.860219E-04**	**9.860219E-04**	9.881338E-04	9.860430E-04	2.111942E-07	+	15992.42	8177.27	**1.00**	2.740
CoDE	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	2.297349E-17	+	26178.55	2028.37	**1.00**	4.486
OXDE	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	2.762568E-17	+	19212.76	**1483.39**	**1.00**	3.292
CoBiDE	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	2.460971E-17	+	15684.91	1902.03	**1.00**	2.688
jDE	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	1.852958E-17	+	21831.59	5277.56	**1.00**	3.741
JADE	**9.860219E-04**	**9.860219E-04**	1.074669E-03	9.881007E-04	1.121451E-05	=	24230.07	26349.77	0.98	4.237
DE-EIG	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	1.526591E-17	+	17978.52	2455.69	**1.00**	3.081
R_{cr}-IJADE	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	4.054469E-17	=	7679.97	2417.67	**1.00**	1.316
OR_{cr}-IJADE	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	**9.860219E-04**	4.509089E-17		**5835.66**	1671.71	**1.00**	**1.000**

To conclude the proper value of β, OR_{cr}-IJADE is faster and produces a more accurate solution than R_{cr}-IJADE does when $\beta = 0.1$ to 0.4 and β with a random, which we conclude the β.

We observed that the *RMSEs* of each pair of algorithms are much closed, but their IAEs might not be the same, e.g., Shell SP70 at 0 °C, from Table S-4 in the supplementary file, the *RMSE* of both algorithm is 2.200606E-09, but the IAEs are 3.0833E-08 (R_{cr}-IJADE) and 1.4763E-05(OR_{cr}-IJADE)—see Table S-20 (0 ℃) in the supplementary file. That means the search surface of the problem might contain a basin or the model is multi-modal.

7.3.2. Compared with the state-of-the-art algorithms

In this subsection, the proposed OR_{cr}-IJADE is compared with other state-of-the-art SI, EA, and NI based algorithms. In order to avoid the bias comparison results, this study attempts to compare the proposed algorithm with 31 selected state-of-the-art EA, SI, and NI algorithms. We roughly grouped these algorithms into five families; including:

1) group of DE family; SaDE [27], jDE [24], JADE [25,26], CoDE [28], EPSDE [29], OXDE [30], R_{cr}-IJADE [3], CoBiDE [56], and DE-EIG [59];
2) group of GA family; GL25 [46];
3) group of PSO family; PSO [79], QPSO [44], CLPSO [45], WQPSO [47], SPSO2011 [49], DNLPSO [52], DMP-PSO [54], and SL-PSO [63];
4) group of ABC family; ABC [68], FlABCps [53], and *Si*PABC_*Sf* [55]; and
5) group of other nature-inspired family; CMA-ES [43], GSA [48], ACiD [50], WCA [51], TLBO [21], GWO [57], LCA [58], SFS [60], HeCoS [61], and ALO [62].

For a fair comparison, all methods are re-coded in MATLAB. For the sake of brevity, only the results of the comparison of extracted parameters and statistical results from the single diode model are chosen as representative of the whole problems, and they are shown in Tables 7 and 8, respectively.

To verify the speed analysis and statistical results, the statistical results from the state-of-the-art SI, EA, and NI algorithms are carried out to compare and analyze. For fairness of comparison, algorithms are optimized by 100 independent runs, while all other parameters are kept the same as in the previous section. The min, median, max, mean, and the associated Std. of *RMSE*, mean *NFEs* and the associated Std., the paired Wilcoxon signed-rank with $\alpha = 0.05$ compares the significance of *RMSE* values between the competitors and OR_{cr}-IJADE with $\beta = 0.1$, the *SR*, and the *AR* compare with the convergence speed between the competitors and OR_{cr}-IJADE with $\beta = 0.1$, are recorded and tabulated in Tables 7–9. The cases that are marked with "+" and "=," indicate that OR_{cr}-IJADE with $\beta = 0.1$ is significantly better than or similar to the competitors. The cases that $AR < 1$ indicate that OR_{cr}-IJADE is faster than the competitors. It can be concluded from the table that SFS, HeCoS, SaDE, EPSDE, CoDE, OXDE, CoBiDE, jDE, DE-EIG, and R_{cr}-IJADE can achieve the same accuracy as OR_{cr}-IJADE can as their *SR*s are 1.0, but OR_{cr}-IJADE is faster than all competitors.

Table 9
Comparison of the ranked *AR* results of all algorithms in all solar cell models.

Problem No.	*Solar cell models*	Ranks of acceleration rate (*AR*)										
		GL25	GSA	TLBO	WCA	SFS	HeCoS	ACiD	GWO	LCA	ALO	PSO
1	Single diode model	17	25	18	13	14	11	12	25	23	25	15
2	Double diode model	14	24	18	6	10	9	11	34	21	24	17
3	PV module	20	29	15	1	19	10	3	26	21	32	28
4	SP70(-25)	13	13	13	13	13	11	12	13	13	13	13
5	SP70(0)	13	13	13	13	13	11	12	13	13	13	13
6	SP70(25)	13	13	13	13	13	12	11	13	13	13	13
7	SP70(50)	14	14	14	14	13	12	8	14	14	14	14
8	SP70(75)	15	15	15	15	13	12	4	15	15	15	14
9	SQ85 (-25)	14	14	14	14	13	11	12	14	14	14	14
10	SQ85 (0)	13	13	13	13	13	11	12	13	13	13	13
11	SQ85 (25)	13	13	13	13	13	12	11	13	13	13	13
12	SQ85 (50)	13	13	13	13	13	11	12	13	13	13	13
13	SQ85 (75)	15	15	15	15	12	4	15	15	15	15	14
14	ST40 (-25)	14	14	14	14	12	11	13	14	14	14	14
15	ST40 (0)	13	13	13	13	13	11	12	13	13	13	13
16	ST40 (25)	10	10	10	10	10	10	8	10	10	10	10
17	ST40 (50)	11	11	11	11	11	11	9	11	11	11	11
18	ST40 (75)	11	11	11	11	11	11	9	11	11	11	11
Average rankings[a]		21.56	23.44	21.56	19.83	18.19	12.39	10.81	23.53	22.33	23.61	20.97

Problem No.	*Solar cell models*	Ranks of acceleration rate (*AR*)										
		CLPSO	QPSO	WQPSO	SPSO2011	DNLPSO	DMP-PSO	SL-PSO	ABC	FlABCps	SiPABC_Sf	CMAES
1	Single diode model	16	19	20	25	22	25	25	21	25	24	25
2	Double diode model	15	22	20	24	23	24	24	19	24	24	24
3	PV module	23	8	9	31	25	30	7	22	24	18	27
4	SP70(-25)	13	13	13	13	13	13	13	13	13	13	13
5	SP70(0)	13	13	13	13	13	13	13	13	13	13	13
6	SP70(25)	13	13	13	13	13	13	13	13	13	13	13
7	SP70(50)	14	14	14	14	14	14	14	14	14	14	14
8	SP70(75)	15	15	15	15	15	15	15	15	15	15	15
9	SQ85 (-25)	14	14	14	14	14	14	14	14	14	14	14
10	SQ85 (0)	13	13	13	13	13	13	13	13	13	13	13
11	SQ85 (25)	13	13	13	13	13	13	13	13	13	13	13
12	SQ85 (50)	13	13	13	13	13	13	13	13	13	13	13
13	SQ85 (75)	15	15	15	15	15	15	15	15	15	15	15
14	ST40 (-25)	14	14	14	14	14	14	14	14	14	14	14
15	ST40 (0)	13	13	13	13	13	13	13	13	13	13	13
16	ST40 (25)	10	10	10	10	10	10	10	10	10	10	10
17	ST40 (50)	11	11	11	11	11	11	11	11	11	11	11
18	ST40 (75)	11	11	11	11	11	11	11	11	11	11	11
Average rankings[a]		21.72	21.44	21.44	23.56	22.61	23.50	22.22	22.17	23.17	22.58	23.33

Problem No.	*Solar cell models*	Ranks of acceleration rate (*AR*)									
		SaDE	EPSDE	CoDE	OXDE	CoBiDE	jDE	JADE	DE-EIG	Rcr-IJADE	OR$_{cr}$-IJADE
1	Single diode model	7	4	10	6	3	8	9	5	2	1
2	Double diode model	3	16	8	7	5	4	13	12	2	1
3	PV module	11	14	17	16	12	6	5	13	3	2
4	SP70(-25)	9	3	7	6	5	8	4	10	2	1
5	SP70(0)	10	3	8	6	5	9	4	7	2	1
6	SP70(25)	9	3	8	5	4	10	6	7	2	1
7	SP70(50)	9	3	10	6	5	11	4	7	2	1
8	SP70(75)	9	3	11	8	5	10	6	7	2	1
9	SQ85 (-25)	10	3	9	6	5	8	4	7	2	1
10	SQ85 (0)	9	3	7	6	5	8	4	10	2	1
11	SQ85 (25)	9	3	8	5	4	10	6	7	2	1
12	SQ85 (50)	9	3	8	6	4	10	5	7	2	1
13	SQ85 (75)	10	3	11	8	6	9	5	7	2	1
14	ST40 (-25)	8	3	10	6	5	9	4	7	2	1
15	ST40 (0)	10	3	8	6	5	9	4	7	2	1
16	ST40 (25)	10	3	7	5	6	9	4	10	2	1
17	ST40 (50)	11	3	8	4	5	10	6	7	2	1
18	ST40 (75)	11	3	8	5	6	10	4	7	2	1
Average rankings[a]		10.89	4.39	9.06	6.50	5.28	8.78	5.39	8.61	2.08	1.06

[a] Average rankings indicated that the average rankings of all algorithms by the Friedman test.

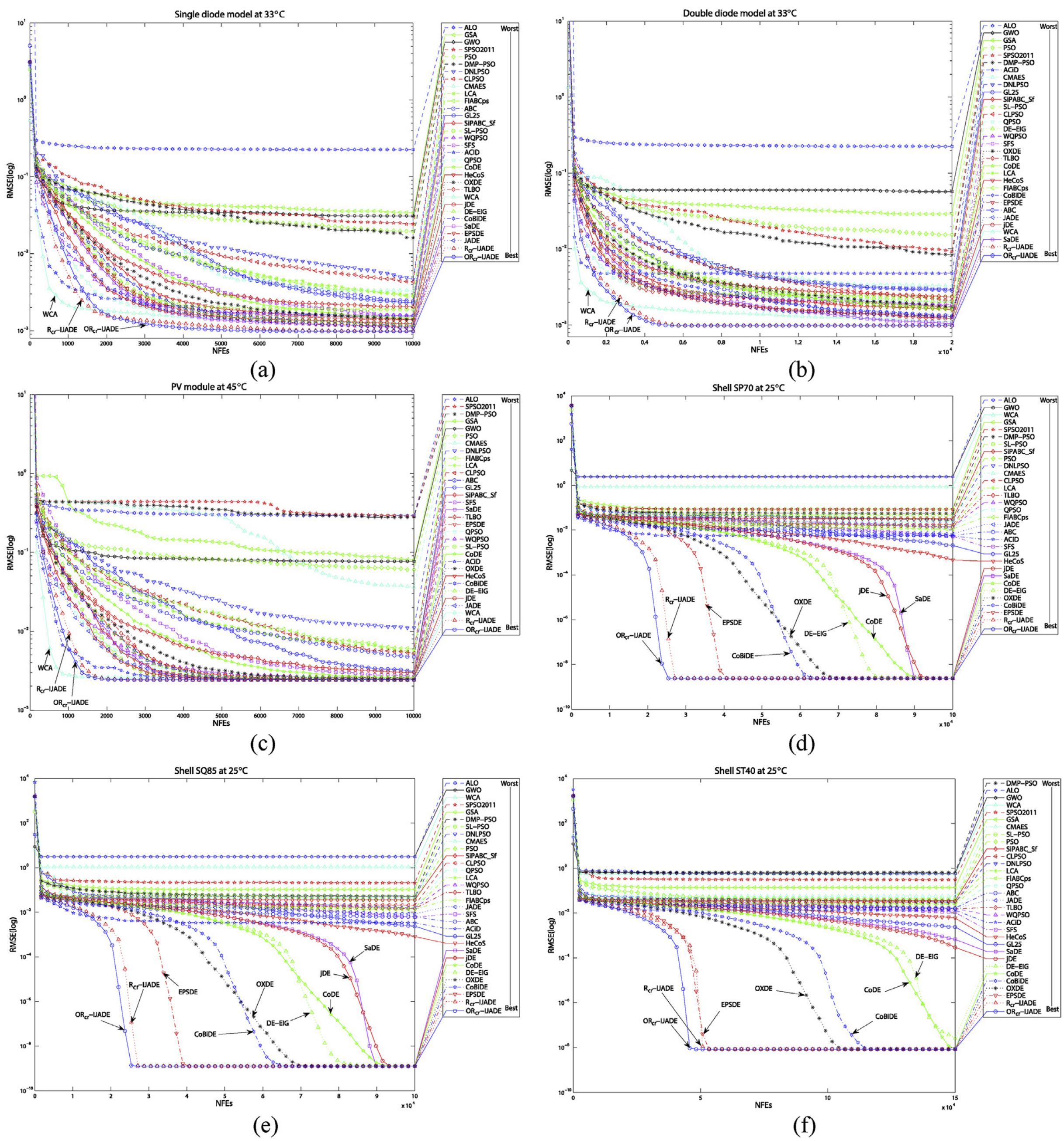

Fig. 4. Convergence graphs of OR_{cr}-IJADE compared to the other state-of-the-art SI, EA, and NI-based algorithms for the selected problems. (a): single diode model; (b): double diode model; (c) PV module; (d): Shell SP70 at 25 °C; (e): Shell SQ85 at 25 °C; (f): Shell ST40 at 25 °C. The legends of the algorithms have been ranked from best to worst (bottom-to-top).

7.3.3. The convergence curve

In this subsection, in order to gain a better knowledge on the convergence rate, the convergence curves of all algorithms are plotted in Fig. 4. Note that the legends of the algorithms have been ranked from best to worst (bottom-to-top). For the sake of limited space, only six out of 18 problems are chosen: the single diode model, double diode model, PV module, Shell SP70, Shell SQ85, and Shell ST40 with the temperature at 25 °C. The y-axis of the convergence graphs shows the *median* error, Eq. (12), of the solution for 100 runs and the x-axis shows *NFEs*. As depicted in Fig. 4, it is clearly observed that the OR_{cr}–IJADE algorithm outperforms the other state-of-the-art SI, EA, and NI-based algorithms in all cases as it can reach the best solution with a fewer *NFEs*. By carefully looking at the curves Fig. 4(a), (b), and (c); we found that the WCA algorithm seems to converge very fast in the early state. However, WCA is eventually trapped in local minima, leading to failure to reach the lowest *RMSE*. Meanwhile, OR_{cr}–IJADE can properly trade-off the exploration and exploitation abilities and successfully reach the lowest *RMSE* faster than the competitors reach. Considering the Fig. 4(d), (e), and (f), we observed that most of the fastest convergence speed produces by the DE family. It means that the advanced DE variants might have a better trade-off between exploration and

Table 10
Comparison of the reported results in literature among different parameters extraction techniques with the proposed techniques for the single diode model. The lowest *RMSE* is highlighted in **boldface**.

Item	GA [17]	CPSO [9]	PS [1]	SA [7]	HS [37]	GGHS [37]	IGHS [37]	ABSO [6]	BMO [39]	BFA [4]	ABC-DE [38]	TLBO [21]	ABC [22]	R_{cr}-IJADE [3]	OR_{cr}-IJADE
$I_{ph}(A)$	0.7619	0.7607	0.7617	0.7620	0.76070	0.76092	0.76077	0.76080	0.76077	0.7602	0.76077	0.7608	0.7608	0.760776	0.760776
I_{sd} (μA)	0.8087	0.4000	0.9980	0.4798	0.30495	0.32620	0.34351	0.30623	0.32479	0.8000	0.32302	0.3223	0.3251	0.323021	0.323021
$R_s(\Omega)$	0.0299	0.0354	0.0313	0.0345	0.03663	0.03631	0.03613	0.03659	0.03636	0.0325	0.03637	0.0364	0.0364	0.036377	0.036377
$R_{sh}(\Omega)$	42.3729	59.0120	64.1026	43.1034	53.5946	53.0647	53.2845	52.2903	53.8716	50.8691	53.7185	53.76027	53.6433	53.718526	53.718523
a	1.5751	1.5033	1.6000	1.5172	1.47538	1.48217	1.48740	1.47583	1.48173	1.6951	1.47986	1.4837	1.4817	1.481184	1.481184
RMSE	1.90780E-02	1.38608E-03	1.49364E-02	1.89978E-02	9.95146E-04	9.90889E-04	1.03345E-03	9.91252E-04	9.86223E-04	2.18866E-01	4.85483E-03	9.69602E-03	1.09667E-03	9.86023136E-04	**9.86023135E-04**

exploitation ability than the other algorithms.

7.3.4. Compared with existing results from the literature

In this subsection, in order to confirm that OR_{cr}-IJADE with $\beta = 0.1$ is better than the other algorithms, the performance of OR_{cr}-IJADE with $\beta = 0.1$ is further investigated by comparison with the reported results presented in the literature. The parameters of single diode, double diode, and the PV module are extracted from the literature, and the *RMSE* values produced by the algorithms mentioned in the literature are re-calculated by using the reported parameters. And hereafter, OR_{cr}-IJADE refers to OR_{cr}-IJADE with $\beta = 0.1$, for short.

To compare the results of single diode model, the GA [17], CPSO [9], PS [1], SA [7], HS [37], GGHS [37], IGHS [37], ABSO [6], BMO [39], BFA [4], ABC-DE [38], R_{cr}-IJADE [3], TLBO [21], and ABC [22] are chosen methods for comparison with OR_{cr}-IJADE. The computational results are tabulated in Table 10. The overall best *RMSE* values are highlighted in **boldface**. The results in Table 10 clearly reveal that OR_{cr}-IJADE provides the best *RMSE* values among all competitors. However, the *RMSE* values of OR_{cr}-IJADE and R_{cr}-IJADE are very close to each other.

To compare the results of double diode model, the extracted parameters are indirectly compared with those of PS [1], SA [7], HS [37], GGHS [37], IGHS [37], ABSO [6], BMO [39], BFA [4], ABC-DE [38], R_{cr}-IJADE [3], and ABC [22] parameters extraction algorithms. The extracted parameters and the re-calculated *RMSE* values of each method are elucidated in Table 11. From Table 11, it is clear that OR_{cr}-IJADE provides the lowest *RMSE* value and the *RMSE* values of OR_{cr}-IJADE and R_{cr}-IJADE are very close to each other.

To compare the results of PV module, the results taken from Newton's study [80], the results taken from Ref. [81], CPSO [9], the results taken from Ref. [82], PS [1], SA [7], R_{cr}-IJADE [3], and TLBO [21] are chosen. The reported results compared with the proposed method of OR_{cr}-IJADE. The re-calculated *RMSE* values of each method are elucidated in Table 12. From Table 12, it can be observed that OR_{cr}-IJADE provides the lowest *RMSE,* and the *RMSE* values of OR_{cr}-IJADE and R_{cr}-IJADE are very close to each other.

8. Conclusions

In this paper, we proposed a new method for a faster and more accurate solar cell parameters extraction. The proposed method embedded an onlooker-ranking-based mutation operator (O(β)R) into an existing R_{cr}-IJADE. The O(β)R operator is an effective and efficient vectors selection mechanism for enhancing the DE when the objective function contains a flat basin. Accordingly, the O(β)R operator has a potential that can be further used to encourage the R_{cr}-IJADE algorithm. This new approach is an advanced DE variant and named as OR_{cr}-IJADE for short. The proposed method can improve the quality of final solutions, the success rate, and the convergence speed of the conventional method (R_{cr}-IJADE algorithm) quite effectively in all cases. In these experiments, the performance of the proposed method was verified. The different types of solar cell models and PV modules from several manufacturers were used to test the ability of the proposed method, which is composed of the single diode model, the double diode model, and the PV module which are acquired form Gong and Cai [3]. Furthermore, the three generated PV modules of different technologies are also included which are Shell SP70, Shell SQ85, and Shell ST40 are simulated according to Soon and Low [12], with the temperature variants. In addition, in order to achieve a meaningful conclusion, the performance of this proposed method is evaluated directly and indirectly in comparison with 31 state-of-the-art EA, SI, and other NI-based methods, along with other methods, which reported their results are in the literature. Experimental results

Table 11
Comparison of the reported results in literature among different parameters extraction techniques with the proposed techniques for the double diode model. The lowest *RMSE* is highlighted in **boldface**.

Item	PS [1]	SA [7]	HS [37]	GGHS [37]	IGHS [37]	ABSO [6]	BMO [39]	BFA [4]	ABC-DE [38]	ABC [22]	R_{cr}-IJADE [3]	OR_{cr}-IJADE
I_{ph}(A)	0.7602	0.7623	0.76176	0.76056	0.76079	0.7608	0.76078	0.7609	0.76078	0.7608	0.760781	0.760781
I_{sd1} (μA)	0.9889	0.4767	0.12545	0.37014	0.97310	0.2671	0.21110	0.0094	0.22599	0.0407	0.225974	0.225974
$R_s(\Omega)$	0.0320	0.0345	0.03545	0.03562	0.03690	0.0366	0.03682	0.0351	0.03674	0.0364	0.036740	0.036740
$R_{sh}(\Omega)$	81.3008	43.1034	46.8269	62.7899	56.8368	54.6219	55.8081	60.0000	55.4921	53.7804	55.485443	55.485438
a_1	1.6000	1.5172	1.49439	1.49638	1.92126	1.4651	1.44533	1.3809	1.44972	1.4495	1.451017	1.451017
I_{sd2} (μA)	0.0001	0.0100	0.25470	0.13504	0.16791	0.3819	0.87688	0.0453	0.75437	0.2874	0.749347	0.749348
a_2	1.1920	2.0000	1.49989	1.92998	1.42814	1.9815	1.99997	1.5255	1.99998	1.4885	2.000000	2.000000
RMSE	1.517666E-02	1.664353E-02	1.259652E-03	1.068370E-03	9.865724E-04	9.857451E-04	9.826615E-04	2.982676E-01	4.852788E-03	1.114580E-03	9.824859E-04	**9.824858E-04**

Table 12
Comparison of the reported results in literature among different parameters extraction techniques with the proposed techniques for the PV module. The lowest *RMSE* is highlighted in **boldface**.

Item	Newton [80]	Method in Ref. [81]	CPSO [9]	Method in Ref. [82]	PS [1]	SA [7]	TLBO [21]	R_{cr}-IJADE [3]	OR_{cr}-IJADE
I_{ph}(A)	1.0318	1.0339	1.0286	1.0310	1.0313	1.0331	1.031805	1.030514	1.030514
I_{sd} (μA)	3.2875	3.0760	8.3010	3.8236	3.1756	3.6642	3.280945	3.482263	3.482263
$R_s(\Omega)$	1.2057	1.2030	1.0755	1.0920	1.2053	1.1989	1.206	1.201271	1.201271
$R_{sh}(\Omega)$	555.5556	555.5556	1850.1000	689.6600	714.2857	833.3333	548.666	981.982240	981.982241
a	48.4500	48.1862	52.2430	48.9300	48.2889	48.8211	48.44228	48.642835	48.642835
RMSE	6.313257E-03	4.786696E-03	4.212772E-03	1.973359E-02	4.507511E-03	4.169322E-03	6.567087E-03	2.425074886073E-03	**2.425074886071E-03**

showed the superiority of the OR_{cr}-IJADE method when being compared to its competitors. The performance of the proposed method successfully solves the parameters extraction problem for solar cell models, and the OR_{cr}-IJADE with $\beta = 0.1$ to 0.4 and $\beta = \text{rand}[0, 1)$ is able to extract the parameters of different solar cell models quickly and accurately.

Acknowledgments

This research was financially supported by Mahasarakham University research support and development fund. In addition, the authors would like to thank Mr. John Coby Davies from Mahasarakham Business School, Mahasarakham University, Thailand, for proofreading.

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.renene.2018.09.017.

References

[1] M.F. AlHajri, K.M. El-Naggar, M.R. AlRashidi, A.K. Al-Othman, Optimal extraction of solar cell parameters using pattern search, Renew. Energy 44 (2012) 238–245.
[2] K. Ishaque, Z. Salam, S. Mekhilef, A. Shamsudin, Parameter extraction of solar photovoltaic modules using penalty-based differential evolution, Appl. Energy 99 (2012) 297–308.
[3] W. Gong, Z. Cai, Parameter extraction of solar cell models using repaired adaptive differential evolution, Sol. Energy 94 (2013b) 209–220.
[4] N. Rajasekar, N. Krishna Kumar, R. Venugopalan, Bacterial Foraging Algorithm based solar PV parameter estimation, Sol. Energy 97 (2013) 255–265.
[5] K. Ishaque, Z. Salam, H. Taheri, Simple, fast and accurate two-diode model for photovoltaic modules, Sol. Energy Mater. Sol. Cell. 95 (2011a) 586–594.
[6] A. Askarzadeh, A. Rezazadeh, Artificial bee swarm optimization algorithm for parameters identification of solar cell models, Appl. Energy 102 (2013a) 943–949.
[7] K.M. El-Naggar, M.R. AlRashidi, M.F. AlHajri, A.K. Al-Othman, Simulated Annealing algorithm for photovoltaic parameters identification, Sol. Energy 86 (2012) 266–274.
[8] E.Q.B. Macabebe, C.J. Sheppard, E. E. v. Dyk, Parameter extraction from I– V characteristics of PV devices, Sol. Energy 85 (2011) 12–18.
[9] W. Huang, C. Jiang, L. Xue, D. Song, Extracting solar cell model parameters based on chaos particle swarm algorithm, in: 2011 International Conference on Electric Information and Control Engineering (ICEICE), 2011, pp. 398–402.
[10] L. Sandrolini, M. Artioli, U. Reggiani, Numerical method for the extraction of photovoltaic module double-diode model parameters through cluster analysis, Appl. Energy 87 (2010) 442–451.
[11] M. Ye, X. Wang, Y. Xu, Parameter extraction of solar cells using particle swarm optimization, J. Appl. Phys. 105 (2009) 094502–094508.
[12] S. Jing Jun, L. Kay-Soon, Photovoltaic model identification using particle swarm optimization with inverse barrier constraint, IEEE Trans. Power Electron. 27 (2012) 3975–3983.
[13] V. Khanna, B.K. Das, D. Bisht, Vandana, P.K. Singh, A three diode model for industrial solar cells and estimation of solar cell parameters using PSO algorithm, Renew. Energy 78 (2015) 105–113.
[14] W.T. da Costa, J.F. Fardin, D.S.L. Simonetti, L.d.V.B.M. Neto, Identification of photovoltaic model parameters by Differential Evolution, in: 2010 IEEE International Conference on Industrial Technology (ICIT), 2010, pp. 931–936.
[15] K. Ishaque, Z. Salam, An improved modeling method to determine the model parameters of photovoltaic (PV) modules using differential evolution (DE), Sol. Energy 85 (2011) 2349–2359.
[16] K. Ishaque, Z. Salam, H. Taheri, A. Shamsudin, A critical evaluation of EA computational methods for Photovoltaic cell parameter extraction based on two diode model, Sol. Energy 85 (2011b) 1768–1779.
[17] M.R. AlRashidi, M.F. AlHajri, K.M. El-Naggar, A.K. Al-Othman, A new estimation approach for determining the I–V characteristics of solar cells, Sol. Energy 85 (2011) 1543–1550.
[18] M.S. Ismail, M. Moghavvemi, T.M.I. Mahlia, Characterization of PV panel and global optimization of its model parameters using genetic algorithm, Energy Convers. Manag. 73 (2013) 10–25.
[19] A.J. Joseph, B. Hadj, A.-L. Ali, Solar cell parameter extraction using genetic algorithms, Meas. Sci. Technol. 12 (2001) 1922.
[20] M. Zagrouba, A. Sellami, M. Bouaïcha, M. Ksouri, Identification of PV solar cells and modules parameters using the genetic algorithms: application to maximum power extraction, Sol. Energy 84 (2010) 860–866.
[21] S.J. Patel, A.K. Panchal, V. Kheraj, Extraction of solar cell parameters from a single current–voltage characteristic using teaching learning based optimization algorithm, Appl. Energy 119 (2014) 384–393.
[22] D. Oliva, E. Cuevas, G. Pajares, Parameter identification of solar cells using artificial bee colony optimization, Energy 72 (2014) 93–102.
[23] R. Storn, K. Price, Differential evolution- a simple and efficient adaptive scheme for global optimization over continuous spaces, in: Tech. Rep. TR-95-012. Berkeley, CA, USA: Int. Comput. Sci. Inst, 1995.
[24] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-Adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (2006) 646–657.
[25] J. Zhang, A.C. Sanderson, JADE: self-adaptive differential evolution with fast and reliable convergence performance, in: IEEE Congress on Evolutionary Computation, 2007. CEC 2007, 2007, pp. 2251–2258.

[26] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (2009) 945–958.
[27] A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: The 2005 IEEE Congress on Evolutionary Computation, 2005, vol. 2, 2005, pp. 1785–1791.
[28] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (2011) 55–66.
[29] R. Mallipeddi, P.N. Suganthan, Q.K. Pan, M.F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2011) 1679–1696.
[30] Y. Wang, Z. Cai, Q. Zhang, Enhancing the search ability of differential evolution through orthogonal crossover, Inf. Sci. 185 (2012) 153–177.
[31] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution – an updated survey, Swarm Evol. Comput. 27 (2016) 1–30.
[32] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (2011) 4–31.
[33] K. Charansiriphaisan, S. Chiewchanwattana, K. Sunat, A global multilevel thresholding using differential evolution approach, Math. Probl Eng. 2014 (2014) 1–23.
[34] D.H. Wolpert, W.G. Macready, No free lunch theorems for search, in: Tech. Rep. SFI-TR-95-02-010, Santa Fe Institute, 1995.
[35] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
[36] D.T. Cotfas, P.A. Cotfas, S. Kaplanis, Methods to determine the dc parameters of solar cells: a critical review, Renew. Sustain. Energy Rev. 28 (2013) 588–596.
[37] A. Askarzadeh, A. Rezazadeh, Parameter identification for solar cell models using harmony search-based algorithms, Sol. Energy 86 (2012) 3241–3249.
[38] O. Hachana, K.E. Hemsas, G.M. Tina, C. Ventura, Comparison of different metaheuristic algorithms for parameter identification of photovoltaic cell/module, J. Renew. Sustain. Energy 5 (2013), 053122.
[39] A. Askarzadeh, A. Rezazadeh, Extraction of maximum power point in solar cells using bird mating optimizer-based parameters identification approach, Sol. Energy 90 (2013b) 123–133.
[40] A. Askarzadeh, A. Rezazadeh, A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: bird mating optimizer, Int. J. Energy Res. 37 (2013c) 1196–1204.
[41] D.H. Muhsen, A.B. Ghazali, T. Khatib, I.A. Abed, A comparative study of evolutionary algorithms and adapting control parameters for estimating the parameters of a single-diode photovoltaic module's model, Renew. Energy 96 (2016) 377–389.
[42] X. Gao, Y. Cui, J. Hu, G. Xu, Z. Wang, J. Qu, H. Wang, Parameter extraction of solar cell models using improved shuffled complex evolution algorithm, Energy Convers. Manag. 157 (2018) 460–479.
[43] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2001) 159–195.
[44] S. Jun, X. Wenbo, F. Bin, A global search strategy of quantum-behaved particle swarm optimization, in: 2004 IEEE Conference on Cybernetics and Intelligent Systems, vol. 1, 2004, pp. 111–116.
[45] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (2006) 281–295.
[46] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, A.M. Sánchez, Global and local real-coded genetic algorithms based on parent-centric crossover operators, Eur. J. Oper. Res. 185 (2008) 1088–1113.
[47] M. Xi, J. Sun, W. Xu, An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position, Appl. Math. Comput. 205 (2008) 751–759.
[48] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (2009) 2232–2248.
[49] M. Clerc, Standard Particle Swarm Optimisation, in: Tech. Rep. Particle Swarm Central, 2012.
[50] I. Loshchilov, M. Schoenauer, M. Sebag, Adaptive coordinate descent, in: K. Natalio, L. Pier Luca (Eds.), Genetic and Evolutionary Computation Conference (GECCO 2011), ACM Press, Dublin, Ireland, 2011, pp. 885–992.
[51] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110–111 (2012) 151–166.
[52] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, P.N. Suganthan, A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization, Inf. Sci. 209 (2012) 16–36.
[53] S. Das, S. Biswas, S. Kundu, Synergizing fitness learning with proximity-based food source selection in artificial bee colony algorithm for numerical optimization, Appl. Soft Comput. 13 (2013) 4676–4694.
[54] R. Kundu, S. Das, R. Mukherjee, S. Debchoudhury, An improved particle swarm optimizer with difference mean based perturbation, Neurocomputing 129 (2014) 315–333.
[55] S. Das, S. Biswas, B.K. Panigrahi, S. Kundu, D. Basu, A spatially informative optic flow model of bee colony with saccadic flight strategy for global optimization, IEEE Trans. Cybernet. 44 (2014) 1884–1897.
[56] Y. Wang, H.-X. Li, T. Huang, L. Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, Appl. Soft Comput. 18 (2014) 232–247.
[57] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Software 69 (2014) 46–61.
[58] A. Husseinzadeh Kashan, League Championship Algorithm (LCA): an algorithm for global optimization inspired by sport championships, Appl. Soft Comput. 16 (2014) 171–200.
[59] G. Shu-Mei, Y. Chin-Chang, Enhancing differential evolution utilizing eigenvector-based crossover operator, IEEE Trans. Evol. Comput. 19 (2015) 31–49.
[60] H. Salimi, Stochastic Fractal Search: a powerful metaheuristic algorithm, Knowl. Base Syst. 75 (2015) 1–18.
[61] X. Ding, Z. Xu, N.J. Cheung, X. Liu, Parameter estimation of Takagi–Sugeno fuzzy system using heterogeneous cuckoo search algorithm, Neurocomputing 151 (3) (2015) 1332–1342.
[62] S. Mirjalili, The ant lion optimizer, Adv. Eng. Software 83 (2015) 80–98.
[63] R. Cheng, Y. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Inf. Sci. 291 (2015) 43–60.
[64] M. Wolf, G.T. Noel, R.J. Stirn, Investigation of the double exponential in the current-Voltage characteristics of silicon solar cells, IEEE Trans. Electron. Dev. 24 (1977) 419–428.
[65] M.G. Villalva, J.R. Gazoli, E.R. Filho, Comprehensive approach to modeling and simulation of photovoltaic arrays, IEEE Trans. Power Electron. 24 (2009) 1198–1208.
[66] J. Liu, J. Lampinen, A fuzzy adaptive differential evolution algorithm, Soft Comput. 9 (2005) 448–462.
[67] J. Teo, Exploring dynamic self-adaptive populations in differential evolution, Soft Comput. 10 (2006) 673–686.
[68] D. Karaboga, An idea based on honey bee swarm for numerical optimization, in: Tech. Rep. TR06, Erciyes University, 2005.
[69] D. Zaharie, Influence of crossover on the behavior of differential evolution algorithms, Appl. Soft Comput. 9 (2009) 1126–1138.
[70] C. Lin, A. Qing, Q. Feng, A comparative study of crossover in differential evolution, J. Heuristics 17 (2011) 675–703.
[71] D. Zaharie, A comparative analysis of crossover variants in differential evolution, in: Proceedings of IMCSIT 2007, 2007, pp. 171–181.
[72] W. Gong, Z. Cai, Y. Wang, Repairing the crossover rate in adaptive differential evolution, Appl. Soft Comput. 15 (2014b) 149–168.
[73] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: a Practical Approach to Global Optimization (Natural Computing Series), Springer-Verlag New York, Inc, 2005.
[74] W. Gong, Z. Cai, Differential evolution with ranking-based mutation operators, IEEE Trans. Cybernet. 43 (2013a) 2066–2081.
[75] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, A novel population initialization method for accelerating evolutionary algorithms, Comput. Math. Appl. 53 (2007) 1605–1614.
[76] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, Opposition-based differential evolution, IEEE Trans. Evol. Comput. 12 (2008) 64–79.
[77] W. Gong, Z. Cai, D. Liang, Engineering optimization by means of an improved constrained differential evolution, Comput. Meth. Appl. Mech. Eng. 268 (2014a) 884–904.
[78] W. Gong, Z. Cai, D. Liang, Adaptive ranking mutation operator based differential evolution for constrained optimization, IEEE Trans. Cybernet. 45 (2015) 716–727.
[79] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.
[80] T. Easwarakhanthan, J. Bottin, I. Bouhouch, C. Boutrit, Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers, Int. J. Sol. Energy 4 (1986) 1–12.
[81] K. Bouzidi, M. Chegaar, N. Nehaoua, New method to extract the parameters of solar cells from their illuminated IV curve, in: 4th International Conference on Computer Intelgrated Manufacturing, 2007, pp. 1–4.
[82] M. Chegaar, N. Nehaoua, A. Bouhemadou, Organic and inorganic solar cells parameters evaluation from single I–V plot, Energy Convers. Manag. 49 (2008) 1376–1379.