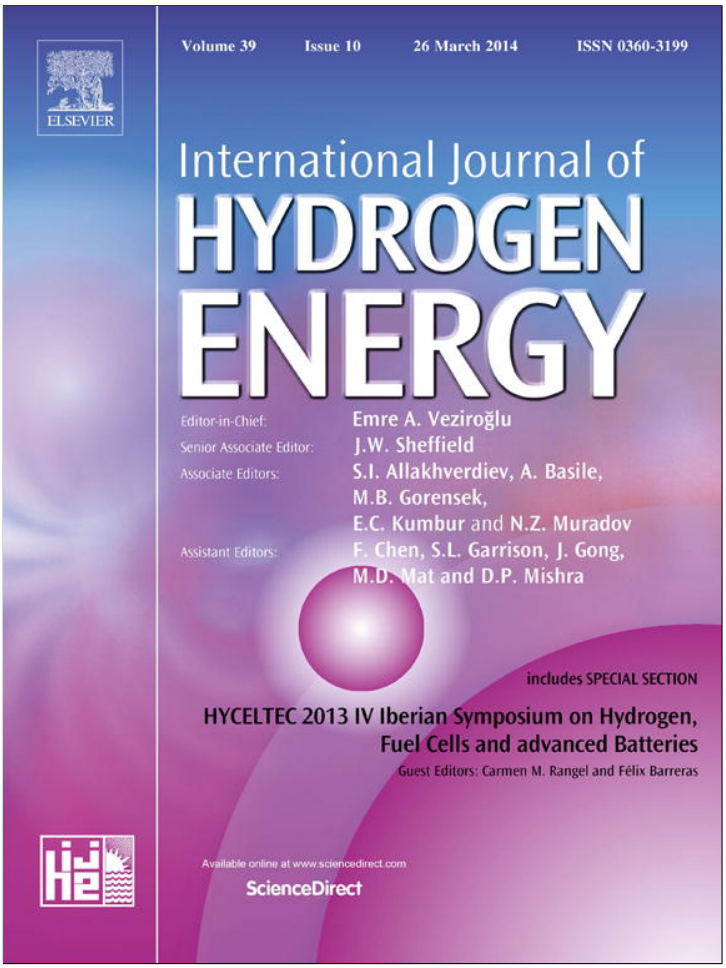

INTERNATIONAL JOURNAL OF HYDROGEN ENERGY 39 (2014) 5083–5096

Available online at www.sciencedirect.com

ScienceDirect

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

Parameter identification of an SOFC model with an efficient, adaptive differential evolution algorithm

Wenyin Gong [a], *Zhihua Cai* [a], *Jie Yang* [b,*], *Xi Li* [c], *Li Jian* [d]

[a] *School of Computer Science, China University of Geosciences, Wuhan 430074, PR China*
[b] *School of Mechanical and Electronic Information, China University of Geosciences, Wuhan 430074, PR China*
[c] *Department of Control Science and Engineering, Huazhong University of Science & Technology, Wuhan 430074, PR China*
[d] *State Key Laboratory of Materials Processing and Die & Mould Technology, Huazhong University of Science and Technology, Wuhan 430074, PR China*

ARTICLE INFO

Article history:
Received 5 August 2013
Received in revised form
28 November 2013
Accepted 12 January 2014
Available online 15 February 2014

Keywords:
Solid oxide fuel cell (SOFC)
Parameter identification
Electrochemical model
Differential evolution algorithms

ABSTRACT

An efficient, adaptive differential evolution (DE) algorithm is proposed in which DE parameter adaptation is implemented. A ranking-based vector selection and crossover rate repairing technique are also presented. The method is referred to as IJADE (Improved Jingqiao Adaptive DE). To verify the performance of IJADE, the parameters of a simple SOFC electrochemical model that is used to control the output performance of an SOFC stack are identified and optimized. The SOFC electrochemical model is built to provide the simulated data. The results indicate that the proposed method is able to efficiently identify and optimize model parameters while showing good agreement with both simulated and experimental data. Additionally, when compared to other DE variants and other evolutionary algorithms, IJADE obtained better results in terms of the quality of the final solutions, robustness, and convergence speed.

1. Introduction

Fuel cells, which have attracted considerable interest in recent years, are electrochemical devices that convert the chemical energy of fuel directly into electrical energy [1]. Depending on the nature of the electrolyte used, there are several different types of fuel cells, such as proton exchange membrane fuel cells (PEMFCs), molten carbonate fuel cells (MCFCs), solid oxide fuel cells (SOFCs), etc. However, among the different types of fuel cells, SOFCs are extremely promising due to their high efficiency, low pollutant emissions, flexible fuel capability, and high waste heat utilization [2].

Modeling SOFCs is of interest for their design and performance control. During the last few decades, many SOFC models have been developed [3–12]. Generally, these models can be classified into three categories [2]: i) white box models, which include the physical model and equivalent circuit model; ii) black box models, which are statistical, data-driven approaches based on artificial intelligence and experimental data; and iii) gray box models, which are partially physical and partially empirical. However, no matter the model type, there are important parameters that must be identified to improve the design and performance control of SOFCs [2,13–15]. For example, in any black box model, such as the genetic algorithm neural network (RBF-NN) model proposed in Ref. [16],

* *Corresponding author*. Tel.: +86 132 0717 0091.
E-mail addresses: wenyingong@yahoo.com, wygong@cug.edu.cn (W. Gong), zhcai@cug.edu.cn (Z. Cai), flyyangj@163.com (J. Yang).
0360-3199/$ – see front matter
http://dx.doi.org/10.1016/j.ijhydene.2014.01.064

Nomenclature

$\mathbf{b}_i$	binary string
Cr	crossover rate
D	number of decision variables
E	Nernst reversible voltage, V
E_0	standard potential, V; open-circuit voltage, V
F	Faraday constant (=96,485 C mol^{-1})
F_i	scaling factor for the ith target vector
I	current density, mA cm^{-2}
I_0	exchange current density, mA cm^{-2}
I_L	limit current, mA cm^{-2}
I_k	kth current density in the sample data, mA cm^{-2}
N	number of cells in a stack
Np	population size
n_e	number of moles of electrons transferred
n	number of sample data
P	partial pressure, atm
R	universal gas constant (=8.314 kJ (kmol K)$^{-1}$)
R_i	rank of each vector $\mathbf{x}_i$ in the sorted population
T	operating temperature, K
$\mathbf{x}$	target vector
$\mathbf{x}_i$	ith vector
$\mathbf{x}^p_{best}$	$p$best solution
$\mathbf{u}_i$	trial vector
V	voltage, V
$\mathbf{V}_i$	mutant vector

Greek letters

β	electron transfer coefficient
μ	mean value

Subscripts

a	anode
act	activation loss
c	cathode
cell	fuel cell
conc	concentration loss
H_2	hydrogen
H_2O	water
i	number population size
j	number of decision variables
k	kth sample value
O_2	oxygen
ohm	ohmic loss
opti	optimization value
rand	random integer
stack	solid oxide fuel cell stack
sample	sample value

Abbreviations

ABC	artificial bee colony
CLPSO	comprehensive learning particle swarm optimizer
CoDE	composite differential evolution
DE	differential evolution
DEGL	differential evolution with global and local mutation operators
EAs	evolutionary algorithms
NFE	number of function evaluations
GA	genetic algorithm
GA-RBF	genetic algorithm-radial basis function
IGA	improved genetic algorithm
IJADE	Improved Jingqiao Adaptive Differential Evolution
JADE	Jingqiao Adaptive Differential Evolution
jDE	j differential evolution
MCFC	molten carbonate fuel cell
MSE	mean squared error
RBF	radial basis function
RBF-NN	radial basis function-neural network
rcGA	real-coded genetic algorithm
PEMFC	proton exchange membrane fuel cell
SaDE	strategy adaptation-based differential evolution
SOFC	solid oxide fuel cell

three parameters, i.e., the output weights, centers, and widths of the Gaussian functions, must be chosen appropriately. An SOFC is a complex nonlinear multivariable strongly coupled system, and the parameter identification and optimization of a simple SOFC electrochemical model is very important to execute an online control strategy. In particular, the working parameters of an SOFC system must be tracked and adjusted when the operation condition suddenly changes [14,17].

Recently, evolutionary algorithm-based optimization techniques for parameter identification problems of SOFC models have received significant attention. The use of evolutionary algorithms (EAs) to identify model parameters appears to be a natural choice because EAs are extremely effective regardless of the gradient and initial condition information [18]. In Refs. [16,19], Wu et al. proposed a genetic algorithm-radial basis function (GA-RBF) neural network method for modeling and predictive control of an SOFC, where a genetic algorithm (GA) was used to optimize the parameters of the radial basis function (RBF) neural networks. Yang et al. [13] presented an improved GA (IGA) method for parameter optimization of a tubular SOFC stack. In Ref. [13], the IGA method was used to optimize the parameters of a simple electrochemical model to fit the simulated data of a dynamic SOFC model. Li et al. proposed a model predictive control strategy for the SOFC control problem based on a GA [20] where a support vector machine model was identified to approximate the behavior of the SOFC system and the GA was used to solve the constrained predictive control problem. In Ref. [21], Bozorgmehri and Hamedi proposed an artificial neural network (ANN) and a GA-based method to model and optimize cell parameters to improve the performance of singular intermediate temperature SOFCs (IT-SOFCs). The ANN is used to model the SOFC performance using experimental data, and the GA is used to optimize SOFC parameters, e.g., anode support thickness, anode support porosity, electrolyte thickness, and functional layer cathode thickness [21]. In Ref. [14], a GA was used to estimate the electrode microstructure distributions in NASA bi-electrode supported SOFCs.

In addition to solving the parameter identification problems for SOFCs, evolutionary algorithms are also used to solve these problems for other full cell models, such as PEMFC models. In Ref. [22], Outeiro et al. proposed a simulated annealing

algorithm to extract the parameter of PEMFC models. Ye et al. [23] presented a particle swarm optimization (PSO)-based parameter identification method for PEMFC models, where both simulated and experimental data are used to evaluate the performance of the PSO. In Ref. [24], the effects of the parameter range, the validation strategy, and the selection of the algorithm on the performance of GAs in solving parameter identification problems for PEMFC models are validated. Askarzadeh and Rezazadeh developed a grouping-based global harmony search algorithm for PEMFC parameter identification [25]. Chakraborty et al. conducted an empirical study of the parameter optimization of PEMFC models using differential evolution (DE) [26], where different DE variants were evaluated and their performance was compared with GAs. In Ref. [27], Yang and Wang presented a bio-inspired P systems-based optimization algorithm to solve PEMFC model parameter estimation problems. In Ref. [28], an adaptive ribonucleic acid (RNA) genetic algorithm was proposed and used to extract the parameters of PEMFC models. Inspired by the foraging behavior of bacteria and bees, a hybrid artificial bee colony technique was presented in Ref. [29] for the parameter identification problems of PEMFC models.

In the aforementioned studies, most of the models are primarily based on a GA. However, serious drawbacks (e.g. low speed, premature convergence, and degradation for highly interactive fitness functions) have been observed for GAs [30]. These drawbacks motivate us to investigate more efficient optimization techniques to solve the parameter identification problems of SOFC models. Differential evolution (DE) [31] is a simple yet powerful evolutionary algorithm that has been successfully used in a variety of domains [32]. However, the use of DE for the parameter identification problems of SOFC models has not been reported. Based on these considerations, the DE method is used in this work to identify the parameters of an SOFC model. Meanwhile, to rapidly and accurately identify the SOFC parameters, the parameter adaptation proposed in JADE (i.e., Jingqiao adaptive DE) [33] is used. Moreover, the ranking-based vector selection and the crossover rate repairing techniques are used to make the algorithm faster and more effective. The proposal is referred to as IJADE (improved JADE). The major advantages of IJADE are as follows: i) the parameter adaptation releases SOFC engineers and researchers from determining the DE optimal parameters (e.g., the crossover rate Cr and scaling factor F); ii) the ranking-based vector selection can enhance the exploitation ability of DE and can hence make the algorithm converge faster; and iii) the crossover rate repairing technique allows the algorithm to rapidly determine the optimal Cr value for a specific problem. To evaluate the performance of the proposed approach in solving the parameter identification problems of SOFC models, a simple electrochemical model is used to implement control of an SOFC stack for output performance and IJADE is used to identify its parameters to match the V–I polarization of the simulated and experimental data. The experimental results indicate that the proposed approach is capable of efficiently identifying the model parameters in all cases. The results obtained by IJADE are in good agreement with both the simulated and experimental data. In addition, through the statistical comparisons with other EAs, the results demonstrate that IJADE provides the best overall results in terms of quality of the final solutions, robustness, and convergence speed. The primary contributions of this work are the proposed IJADE method and its application in parameter identification of SOFC models. To the best of our knowledge, solving the parameter identification problems of SOFC models with the DE algorithm has not been reported.

The rest of this study is organized as follows. In Section 2, the SOFC electrochemical model and objective function used in this work are briefly introduced. Section 3 presents the proposed IJADE method in detail, followed by the experimental results and analysis in Section 4. Lastly, in Section 5, this work is concluded and several possible avenues of future work are discussed.

2. Problem statement

For the sake of completeness, in this section, the SOFC electrochemical model is briefly described first. Then, a simple electrochemical model used to implement control of an SOFC stack for output performance is introduced. Third, the objective function for parameter identification of an SOFC model is given.

2.1. SOFC electrochemical model

In this work, a physically based dynamic SOFC model proposed in Ref. [5] will be used to generate the simulated data. Therefore, the electrochemical model will be briefly introduced as presented in Ref. [5].

As given in Refs. [5,34,35], the output voltage of a cell can be written as:

$$V_{\text{cell}} = E_{\text{cell}} - V_{\text{act,cell}} - V_{\text{ohm,cell}} - V_{\text{conc,cell}} \tag{1}$$

where E_{cell} is the Nernst reversible voltage of a cell, $V_{\text{act,cell}}$ is the activation loss, $V_{\text{ohm,cell}}$ is the ohmic loss, and $V_{\text{conc,cell}}$ is the concentration loss. Then, the output voltage of the SOFC stack with N_{cell} cells can be obtained as [5]:

$$V_{\text{stack}} = N_{\text{cell}} V_{\text{cell}} = E - V_{\text{act}} - E_{\text{ohm}} - E_{\text{conc}} \tag{2}$$

2.1.1. Nernst reversible voltage

The Nernst reversible voltage E_{cell} can be calculated as:

$$E_{\text{cell}} = E_{0,\text{cell}} + \frac{RT}{4F} \ln\left[\frac{(P_{H_2})^2 P_{O_2}}{(P_{H_2O})^2}\right] \tag{3}$$

where $E_{0,\text{cell}}$ is the standard potential; $R = 8.314$ kJ (kmol K)$^{-1}$, which is the universal gas constant; $F = 96{,}486$ C mol^{-1}, which is the Faraday constant; T is the operating temperature of the fuel cell in degrees Kelvin; P_{H_2} is the partial pressure of hydrogen; P_{O_2} is the partial pressure of oxygen; and P_{H_2O} is the partial pressure of water.

2.1.2. Activation voltage loss

According to the well-known Butler–Volmer equation, the activation voltage loss is normally given by [4,36]:

$$I = I_0 \left\{ \exp\left(\beta \frac{n_e F V_{\text{act,cell}}}{RT}\right) - \exp\left(-(1-\beta)\frac{n_e F V_{\text{act,cell}}}{RT}\right) \right\} \tag{4}$$

where I_0 is the exchange current density, β is the transfer coefficient, and n_e is the number of moles of electrons

transferred. For fuel cell applications, the transfer coefficient β is typically set at 0.5. Then Equation (4) can be expressed as:

$$I = 2I_0 \sinh\left(\frac{n_e F V_{act,cell}}{2RT}\right) \tag{5}$$

Therefore,

$$V_{act,cell} = \frac{2RT}{n_e F}\sinh^{-1}\left(\frac{I}{2I_0}\right) \tag{6}$$

2.1.3. Ohmic voltage loss

Ohmic resistance occurs due to resistance to the flow of ions in the electrolyte and resistance to the flow of electrons through the electrode materials. The overall ohmic voltage loss can be written as:

$$V_{ohm,cell} = IR_{ohm,cell} \tag{7}$$

where $R_{ohm,cell}$ is the ionic resistance, which normally decreases as the temperature increases [5].

2.1.4. Concentration voltage loss

Concentration voltage loss occurs due to the mass transfer resistance to the flow of the reactants and products through the porous electrodes. In Ref. [5], the concentration voltage loss is calculated as:

$$V_{conc,cell} = \frac{RT}{4T}\left\{\ln\left[\frac{(P_{H_2})^2 P_{O_2}}{(P_{H_2O})^2}\right] - \ln\left[\frac{\left(P^*_{H_2}\right)^2 P^*_{O_2}}{\left(P^*_{H_2O}\right)^2}\right]\right\} \tag{8}$$

where $P^*_{H_2}$, $P^*_{O_2}$, and $P^*_{H_2O}$ are the effective partial pressures of hydrogen, oxygen, and water, respectively.

2.2. Simple SOFC electrochemical model

In this work, a simple electrochemical model is chosen to implement control of an SOFC stack for output performance. According to Ref. [1], if only a static response is considered, the operational voltage of a fuel cell can be modeled by:

$$V = E_0 - A\ln\left(\frac{I}{I_0}\right) - IR_{ohm} + B\ln\left(1 - \frac{L}{I_L}\right) \tag{9}$$

where E_0 is the open-circuit voltage, A is the slope of the Tafel line, R_{ohm} is the area-specific resistance in kΩ cm^2, B is a constant that depends on the fuel cell and its operating state, and I_L is the limit current density in mA cm^{-2}. Actually, in Equation (9), the second term on the right side is the Tafel equation [4]. However, according to Ref. [4], the Tafel equation is normally used under a high activation polarization. At a low activation polarization, the Tafel equation may cause large errors. Therefore, to better fit the results of actual SOFCs, the Butler–Volmer equation is applied in this work to describe the activation voltage loss. Then, Equation (9) can be modified as [4,34]:

$$V = E_0 - A\sinh^{-1}\left(\frac{I}{2I_{0,a}}\right) - A\sinh^{-1}\left(\frac{I}{2I_{0,c}}\right) - IR_{ohm} + B\ln\left(1 - \frac{I}{I_L}\right) \tag{10}$$

where $I_{0,a}$ and $I_{0,c}$ are the anode and cathode exchange current density, respectively, in mA cm^{-2}.

2.3. Parameter identification based on an objective function

Before using optimization techniques to identify the parameters of an SOFC model, there are two issues that must be addressed. First, the parameters that will be optimized must be chosen. In the above SOFC electrochemical model formulated by Equation (10), there are seven unknown parameters: E_0, A, $I_{0,a}$, $I_{0,c}$, R_{ohm}, B, and I_L. In this work, all seven parameters are treated as unknown parameters to be identified by the optimization algorithm. Thus, the decision vector $\mathbf{x}$ is formulated as:

$$\mathbf{x} = \{E_0,\ A,\ I_{0,a},\ I_{0,c},\ R_{ohm},\ B,\ I_L\} \tag{11}$$

The other important issue in evolutionary algorithms is to determine the objective function. When optimization techniques are used in parameter identification problems of SOFC models, the objective function should be defined first. In this work, the mean squared error (MSE) is used as the objective function [13]:

$$f(\mathbf{x}) = \frac{1}{n}\cdot\sum_{k=1}^{n}(\text{error}_k)^2 = \frac{1}{n}\cdot\sum_{k=1}^{n}\left(V_{sample,k} - V_{opti,k}\right)^2 = \frac{1}{n}\cdot\sum_{k=1}^{n}\left\{V_{sample,k} - N_{cell}\left[E_0 - A\sinh^{-1}\left(\frac{I_k}{2I_{0,a}}\right) - A\sinh^{-1}\left(\frac{I_k}{2I_{0,c}}\right) - I_k R_{ohm} + B\ln\left(1 - \frac{I_k}{I_L}\right)\right]\right\}^2 \tag{12}$$

subject to:

$$I_k < I_L, \quad k = 1, \cdots, n \tag{13}$$

where n is the number of sample data points, N_{cell} is the number of cells in the SOFC stack, and I_k is the kth current density in the sample data. To optimally fit the sample data, the above objective function must be minimized with respect to the set of unknown parameters $\mathbf{x}$ under the constraint in Equation (13). Additionally, according to [4], $I_{0,a} > I_{0,c}$ is used. Clearly, the smaller the objective function, the smaller is the difference between the experimentally measured and calculated voltages. Theoretically, the objective function should be zero for any experimental V–I data when the exact value has been identified for each parameter. However, a small value is actually expected due to the

presence of measurement noise and numerical calculation errors.

3. The proposed IJADE method

In this section, the proposed IJADE method will be introduced in detail. Because IJADE is an improved version of JADE [33], the JADE method is first briefly presented in Section 3.1, followed by the description of IJADE in Section 3.2.

3.1. The JADE method

To improve the performance of the original DE algorithm, Zhang and Sanderson proposed the JADE method in Ref. [33], which is briefly described as follows.

3.1.1. Population initialization

The population of JADE consists of Np vectors. Initially, the population is generated at random. For example, for the ith vector, $\mathbf{x}_i$ is initialized as follows:

$$x_{i,j} = \underline{x}_j + \mathrm{rndreal}(0,1)\cdot\left(\overline{x}_j - \underline{x}_j\right) \tag{14}$$

where $\underline{x}_j$ and $\overline{x}_j$ are the lower bound and upper bound, respectively, of x_j, i.e., $x_j \in [\underline{x}_j, \overline{x}_j]$. $i = 1,\cdots,Np, \quad j = 1,\cdots,D$. Np is the population size, D is the number of decision variables (in this work, $D = 7$), and rndreal(0, 1) is a uniformly distributed random real number in (0, 1).

3.1.2. Mutation operator

After initialization, the mutation operator is applied to generate the mutant vector $\mathbf{v}_i$ for each target vector $\mathbf{x}_i$ in the current population. In JADE [37], the authors presented the "DE/rand-to-pbest/1" mutation as:

$$\mathbf{v}_i = \mathbf{x}_{r1} + F_i\cdot(\mathbf{x}^p_{\mathrm{best}} - \mathbf{x}_{r1}) + F_i\cdot(\mathbf{x}_{r2} - \mathbf{x}_{r3}) \tag{15}$$

where F_i is the scaling factor for the ith target vector, $\mathbf{x}^p_{\mathrm{best}}$ refers to the pbest solution, which is randomly selected from the top $100\times p$ solutions with $p\in(0,1)\cdot r_1, r_2, r_3 \in \{1,\cdots,Np\}$, and $r_1 \neq r_2 \neq r_3 \neq i$.

3.1.3. Crossover operator

To diversify the current population, following mutation, JADE uses the crossover operator to produce the trial vector $\mathbf{u}_i$ between $\mathbf{x}_i$ and $\mathbf{v}_i$. The most commonly used operator is the binomial crossover performed on each component as follows:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if}\left(\mathrm{rndreal}(0,1) < \mathrm{Cr}_i \ \text{ or } \ j == j_{\mathrm{rand}}\right) \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{16}$$

where Cr_i is the crossover rate, j_{rand} is a randomly generated integer within {1, D}, and "a == b" means that a is equal to b.

3.1.4. Selection

Lastly, to maintain a constant population size in the following generations, the selection operation is used to determine whether the trial or target vector survives to the next generation. In JADE, the one-to-one tournament selection is used as follows:

$$\mathbf{x}_i = \begin{cases} \mathbf{u}_i, & \text{if } f(\mathbf{u}_i) \le f(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{otherwise} \end{cases} \tag{17}$$

where $f(\mathbf{x})$ is the objective function to be optimized.

3.1.5. Parameter adaptation

Previous studies [38,39], indicate that the parameter settings of Cr_i and F_i are crucial to the performance of DE. To remedy this drawback, in JADE [33], the parameters Cr_i and F_i are adaptively controlled according to their successful experience in the last generation. The adaptation techniques proposed in JADE are briefly introduced as follows.

For each generation, for each target vector, the crossover rate Cr_i is independently generated as follows:

$$\mathrm{Cr}_i = \mathrm{rndn}_i(\mu_{\mathrm{Cr}}, 0.1) \tag{18}$$

and truncated to the interval [0, 1], where μ_{Cr} is the mean value to generate Cr_i and is updated as follows:

$$\mu_{\mathrm{Cr}} = (1-c)\cdot\mu_{\mathrm{Cr}} + c\cdot\mathrm{mean}_{\mathrm{A}}(S_{\mathrm{Cr}}) \tag{19}$$

where c is a constant in [0, 1]; $\mathrm{mean}_{\mathrm{A}}(\cdot)$ is the typical arithmetic mean operation; and S_{Cr} is the set of all successful crossover rates Cr_i at generation g.

To maintain the population diversity, for each target vector, the mutation factor F_i is independently calculated in the following way:

$$F_i = \mathrm{rndc}_i(\mu_F, 0.1) \tag{20}$$

and then truncated to be 1.0 if $F_i > 1.0$ or regenerated if $F_i \le 0$. $\mathrm{rndc}_i(\mu_F,0.1)$ is 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 in the following manner:

$$\mu_F = (1-c)\cdot\mu_F + c\cdot\mathrm{mean}_{\mathrm{L}}(S_F) \tag{21}$$

where S_F is the set of all successful mutation factors F_i at generation g; and $\mathrm{mean}_{\mathrm{L}}(\cdot)$ is the Lehmer mean:

$$\mathrm{mean}_{\mathrm{L}}(S_F) = \frac{\sum_{i=1}^{|S_F|} F_i^2}{\sum_{i=1}^{|S_F|} F_i} \tag{22}$$

3.2. Improved JADE

To further enhance the performance of JADE to solve parameter identification problems of SOFC models, two improvements are proposed: i) the use of a ranking-based vector selection in the "DE/rand-to-pbest/1" mutation to accelerate the convergence speed, and ii) the implementation of a crossover rate repairing technique to rapidly determine the optimal Cr value. In addition, two constraint handling techniques are used to address the violated solutions.

3.2.1. Ranking-based vector selection

In Equation (15), because $\mathbf{x}_{r1}$, $\mathbf{x}_{r2}$, and $\mathbf{x}_{r3}$ are only selected randomly from the population, the result may be an algorithm that is good at exploring the search space but slow at finding solutions. Therefore, to reduce the computational efforts and accelerate the parameter identification process of SOFC models, the previously proposed ranking-based vector

selection technique [40] is used in the mutation operator in this work.

In the ranking-based vector selection technique, the population is first ranked from the best to the worst in terms of the objective function value $f(\mathbf{x})$ of each vector. Then, the rank R_i of each vector xi in the sorted population is calculated by:

$$R_i = Np - i, \quad i = 1, 2, \cdots, Np \tag{23}$$

Afterwards, the selection probability of each vector is calculated as follows:

$$pv_i = \left(\frac{R_i}{Np}\right)^2 \tag{24}$$

Lastly, vectors ($\mathbf{x}_{r1}$ and $\mathbf{x}_{r2}$) in the mutation are selected according to their selection probabilities as shown in Algorithm 1. Clearly, better solutions have a better likelihood of being chosen as $\mathbf{x}_{r1}$ and $\mathbf{x}_{r2}$ in the "DE/rand-to-pbest/1" mutation to generate the mutant vector.

Algorithm 1. Ranking-based vector selection.

Input: The target vector index i
Output: The selected vector indexes r_1, r_2, r_3
while rndreal$[0, 1) > pv_{r_1}$ ***or*** $r_1 == i$ **do**
  Randomly select $r_1 \in \{1, \mu\}$;
Randomly select $r_2 \in \{1, \mu\}$;
while rndreal$[0, 1) > pv_{r_2}$ ***or*** $r_2 == r_1$ ***or*** $r_2 == i$ **do**
  Randomly select $r_2 \in \{1, \mu\}$;
Randomly select $r_3 \in \{1, \mu\}$;
while $r_3 == r_2$ ***or*** $r_3 == r_1$ ***or*** $r_3 == i$ **do**
  Randomly select $r_3 \in \{1, \mu\}$;

3.2.2. *Crossover rate repairing technique*

As mentioned, the binomial crossover is used in JADE, as shown in Equation (16). To analyze the behavior of the binomial crossover, let bi be a binary string generated for each target vector xi as follows:

$$b_{i,j} = \begin{cases} 1, & \text{if}(\text{rndreal}(0,1) < \text{Cr}_i \text{ or } j == j_{\text{rand}}) \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

Then, the binomial crossover in Equation (16) 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 $i = 1, \cdots, Np$ and $j = 1, \cdots, D$. According to Equations (25) and (26), it can be observed that the binary string $\mathbf{b}_i$ is stochastically related to Cr_i; however, the trial vector $\mathbf{u}_i$ is directly related to its binary string $\mathbf{b}_i$ but not directly related to its crossover rate Cr_i. Based on this consideration, the crossover rate repairing technique is proposed, where the crossover rate is repaired by its corresponding binary string, i.e., using the average number of components taken from the mutant. Suppose that Cr_i' is the repaired crossover rate and is calculated as:

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

where $\mathbf{b}_i$ is the binary string calculated in Equation (25). The crossover rate is repaired after its binary string is generated by Equation (25) based on Cr_i. If the trial vector $\mathbf{u}_i$ is a successful vector, Cr_i' will be stored in S_{Cr} rather than storing Cr_i.

Table 1 – Operating conditions in the tubular SOFC model presented in Ref. [5].

Parameter	Value
Cell number	96
Anode pressure (atm)	3
Cathode pressure (atm)	3
H_2 mass flow rate (mol/s)	0.9e-3
H_2O mass flow rate (mol/s)	1.0e-4
Air mass flow rate (mol/s)	12.0e-3
Inlet fuel temperature (K)	1173
Inlet air temperature (K)	1173
Load current (A)	0–158

3.2.3. *Constraint handling techniques*

There are two types of constraints when applying IJADE for parameter identification of an SOFC model, i.e., boundary constraints and constrained functions. If the solution violates one of the constraints, the constraint handling technique must be used to address the violated constraint to make the solution feasible.

With respect to the boundary constraint, after performing the mutation operation to generate a new solution, several variables may be out of their corresponding boundaries, i.e., $x_j \notin [\underline{x}_j, \overline{x}_j]$. If this occurs, the reinitialization boundary handling technique is applied:

$$x_j = \underline{x}_j + \text{rndreal}(0,1) \cdot \left(\overline{x}_j - \underline{x}_j\right) \tag{28}$$

There are two constrained functions, $I_k < I_L$ and $I_{0,a} > I_{0,c}$, where $k = 1, \cdots, n$. If the solution violates one of the constrained functions, an extremely large objective function value is assigned to this solution, i.e.,

$$f(\mathbf{x}) = \begin{cases} \text{INF}, & \text{if } I_k \geq x(I_L) \text{ or } x(I_{0,a}) \leq x(I_{0,c}) \\ \text{calculated by Equation (12)}, & \text{otherwise} \end{cases} \tag{29}$$

where INF is an extremely large positive value (e.g., 1e100), and $x(I_L)$ is the I_L value of the solution $\mathbf{x}$.

3.2.4. The IJADE algorithm

Combining the parameter adaptation techniques presented in JADE [33] with the crossover rate repairing technique and the ranking-based "DE/rand-to-pbest/1" mutation, the proposed IJADE is developed. The pseudo-

Algorithm 2. The pseudo-code of the IJADE algorithm.

Input: Control parameters: Np and Max_NFEs
Output: The best final solution
Set $\mu_{Cr} = 0.5, \mu_F = 0.5, c = 0.1, p = 0.05$ as presented in [23];
Initialize the population $\mathbb{P}$ randomly;
Calculate the objective function value of each solution in the population;
NFEs = Np;
while *NFEs < Max_NFEs* **do**
  $S_{Cr} = \phi, S_F = \phi$;
  Sort the population from the best to the worst based on the objective function value of each solution;
  for $i = 1$ *to* Np **do** /* Calculate parameters */
    Generate Cr_i and F_i with Equations (18) and (20), respectively;
    Calculate the ranking R_i and the selection probability pr_i;
  for $i = 1$ *to* Np **do** /* Generate $\mathbf{v}_i$ and $\mathbf{u}_i$ */
    Select r_1, r_2, and r_3 as shown in Algorithm 1;
    Produce the mutant vector $\mathbf{v}_i$ with ranking-based "DE/rand-to-$p$best/1";
    Get the binary string $\mathbf{b}_i$ as shown in Equation (25);
    Calculate the repaired crossover rate Cr'_i with Equation (27);
    for $j = 1$ *to* D **do**
      $u_{i,j} = b_{i,j} \cdot v_{i,j} + (1 - b_{i,j}) \cdot x_{i,j}$;
    Apply the boundary constraint-handling to the violated solution as shown in Equation (28);
    Calculate the objective function value $f(\mathbf{u}_i)$ of $\mathbf{u}_i$ according to Equation (29);
  for $i = 1$ *to* Np **do** /* Survival selection */
    if $\mathbf{u}_i$ *is better than its parent* $\mathbf{x}_i$ **then**
      $\mathbf{x}_i = \mathbf{u}_i$;
      $Cr'_i \rightarrow S_{Cr}$;
      $F_i \rightarrow S_F$;
  Update the μ_{Cr} and μ_F with Equations (19) and (21), respectively;
  NFEs = NFEs + Np;

Table 2 – Identified parameters by the IJADE method for the simulated data at different temperatures.

Parameter	Range	1073 K	1123 K	1173 K	1223 K	1273 K
E_0 (V)	[0, 1.2]	1.1145	1.1140	1.1133	1.1123	1.1112
A (V)	[0, 1]	0.0364	0.0283	0.0250	0.0236	0.0230
$I_{0,c}$ (mA cm^{-2})	[0, 30]	6.7820	5.1917	4.3163	3.9061	3.6934
$I_{0,a}$ (mA cm^{-2})	[0, 30]	27.7513	24.7935	22.1158	19.4218	17.2071
R_{ohm} (kΩ cm^2)	[0, 1]	0.0040	0.0036	0.0031	0.0028	0.0025
B (V)	[0, 1]	0.0653	0.0694	0.0741	0.0785	0.0823
i_L (mA cm^{-2})	[0, 200]	159.8741	159.9367	160.0318	160.1001	160.1285
MSE (V^2)		1.42E-03	6.03E-04	1.87E-04	7.39E-05	5.62E-05
R^2		1.0000	1.0000	1.0000	1.0000	1.0000
Running time (s)		3.0270	3.0260	3.0860	3.1070	3.1030

Table 3 – Identified parameters by the IJADE method for the simulated data at different pressures.

Parameter	Range	1 atm	3 atm	5 atm	7 atm	9 atm
E_0 (V)	[0, 1.2]	1.0855	1.1133	1.1262	1.1347	1.1411
A (V)	[0, 1]	0.0250	0.0250	0.0250	0.0250	0.0250
$I_{0,c}$ (mA cm^{-2})	[0, 30]	4.3262	4.3163	4.3168	4.3166	4.3166
$I_{0,a}$ (mA cm^{-2})	[0, 30]	22.2237	22.1158	22.1402	22.1411	22.1432
R_{ohm} (kΩ cm^2)	[0, 1]	0.0031	0.0031	0.0031	0.0031	0.0031
B (V)	[0, 1]	0.0742	0.0741	0.0742	0.0742	0.0742
i_L (mA cm^{-2})	[0, 200]	160.0336	160.0318	160.0343	160.0344	160.0346
MSE (V^2)		1.84E-04	1.87E-04	1.88E-04	1.88E-04	1.87E-04
R^2		1.0000	1.0000	1.0000	1.0000	1.0000
Running time (s)		3.1510	3.0860	3.0730	2.9950	3.0310

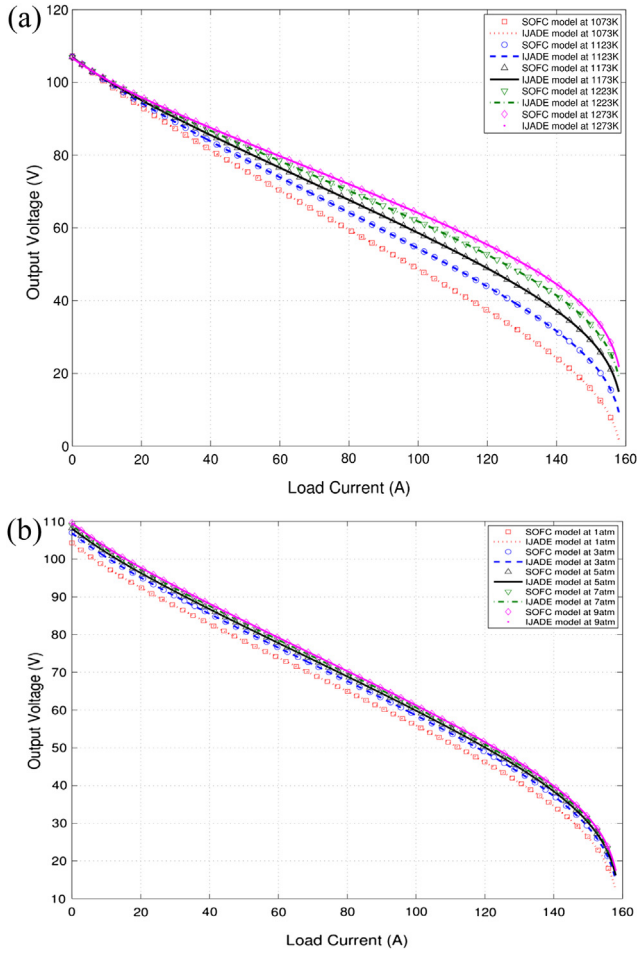

Fig. 1 – Comparisons between the simulated data and the data obtained by IJADE at different temperatures ((a)) and different pressures ((b)).

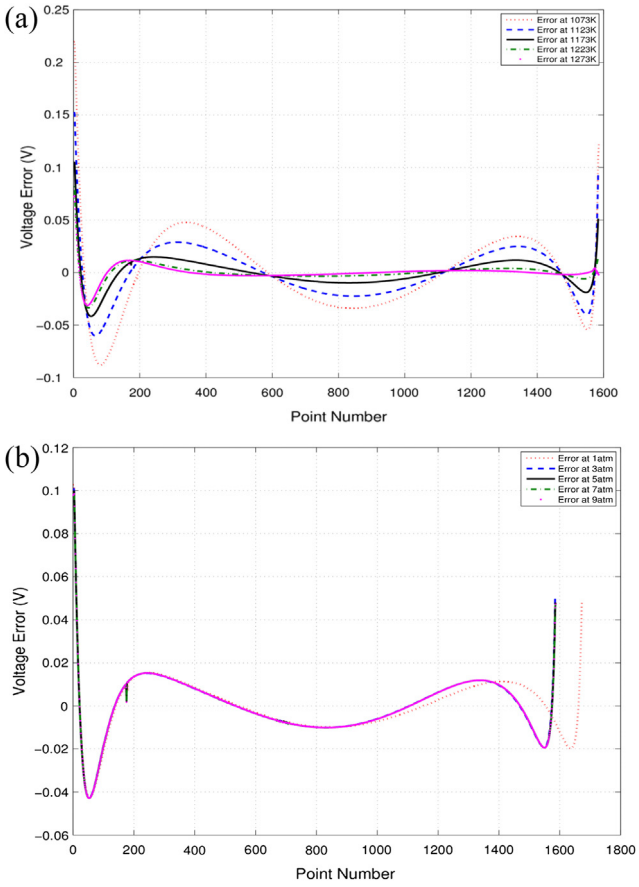

Fig. 2 – The voltage deviation between the simulated data and the data obtained by IJADE at different temperatures ((a)) and different pressures ((b)).

code of IJADE is introduced in Algorithm 2, where NFEs indicates the number of function evaluations, and Max_NFEs is the maximum number of NFEs, which is the termination criterion in this work. From Algorithm 2, it can be observed that there are only two parameters (*Np* and Max_NFEs) that must be selected by the user, whereas the other parameters (i.e., μ_{Cr}, μ_F, c, *p*) are kept as default values, as in JADE [33]. This approach means that IJADE can easily be applied in real-world applications.

4. Results and analysis

4.1. *Parameter settings*

Because the parameter adaptation of Cr and *F* is used, there are only two parameters that must be initialized for the IJADE method in this work. In all tests, the initial values were $Np = 50$ and Max_NFEs = 15,000. Recall that because different algorithms may have different NFEs at one generation, the maximum number of generations is not used as the termination criterion. The programs were executed on the following platform: CPU: Inter Core i7-3770 3.40 GHz; RAM: 8.00 GB; Operating system: Microsoft Windows 7 Home Edition; Compiler: Microsoft Visual C++ 6.0. All programs used in this work were coded in standard C++.

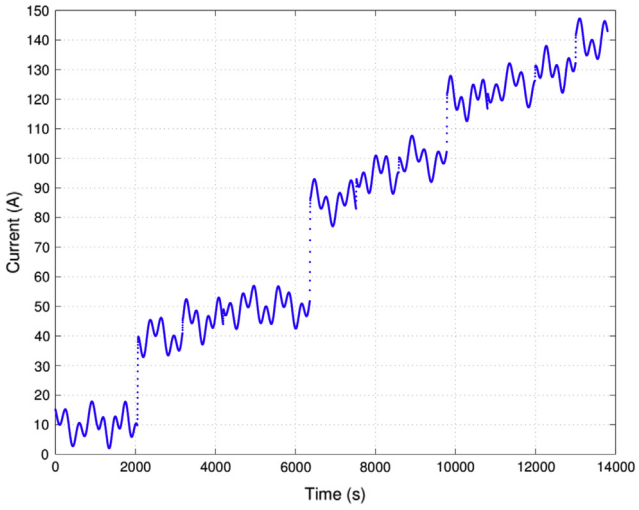

Fig. 3 – The load current with the wave signal and the step signal obtained from Ref. [13].

Table 4 – Identified parameters by the IJADE method for the simulated data with interference at 1173 K.

Parameter	E_O (V)	A (V)	$i_{O,c}$ (mA cm^{-2})	$i_{O,a}$ (mA cm^{-2})	R_{ohm} (KΩ cm^2)	B (V)	i_L (mA cm^{-2})	MSE (V^2)	R^2	Running time (s)
Range	[0, 1.2]	[0, 1]	[0, 30]	[0, 30]	[0, 1]	[0, 1]	[0, 200]			
Value	1.1126	0.0230	4.2179	0.0032	0.0702	159.2115	19.0755	0.0066132	1.0000	2.6680

4.2. *Results comparison to simulated data without interference*

To validate the performance of IJADE for parameter identification of an SOFC model, IJADE is applied to identify the unknown parameters of a simple electrochemical model, as mentioned in Section 2.2. A physically based dynamic tubular SOFC model proposed in Ref. [5] is used to generate the simulated V–I data. This model was chosen because the MATLAB/Simulink simulator of this model was provided by Wang and Nehrir in Ref. [41]. The operating conditions in the tubular SOFC model are shown in Table 1. The simulated data are generated with the simulator set to different temperatures (1073 K, 1123 K, 1173 K, 1223 K, and 1273 K) and different pressures (1 atm, 3 atm, 5 atm, 7 atm, and 9 atm). Each data set has approximately 15,800 data points, where only approximately 1580 data points were selected from the original data set with a step size of 10 to reduce the running time. Note that more data points can result in the algorithm obtaining a smaller objective function value. For the seven unknown parameters, their search ranges are reported in Tables 2 and 3. Note that to accelerate the search process, the search ranges of each parameter in the simple electrochemical model are set according to their physical meaning [13]. The parameters identified by the IJADE method at different temperatures and different pressures are reported in Tables 2 and 3, respectively. In addition, the MSE values, R^2 values, and running time in seconds are also reported in these tables. Note that R^2 is the coefficient of determination, which measures the goodness of fit between the sample data and the data predicted by IJADE. The running time is the consumed CPU time in seconds when the Max NFEs are worked out in one run. In addition, comparisons of the V–I characteristics of the simulated data and the data obtained by IJADE are plotted in Fig. 1(a) and (b), respectively, at different temperatures and different pressures. The output voltage errors for the simulated data are shown in Fig. 2.

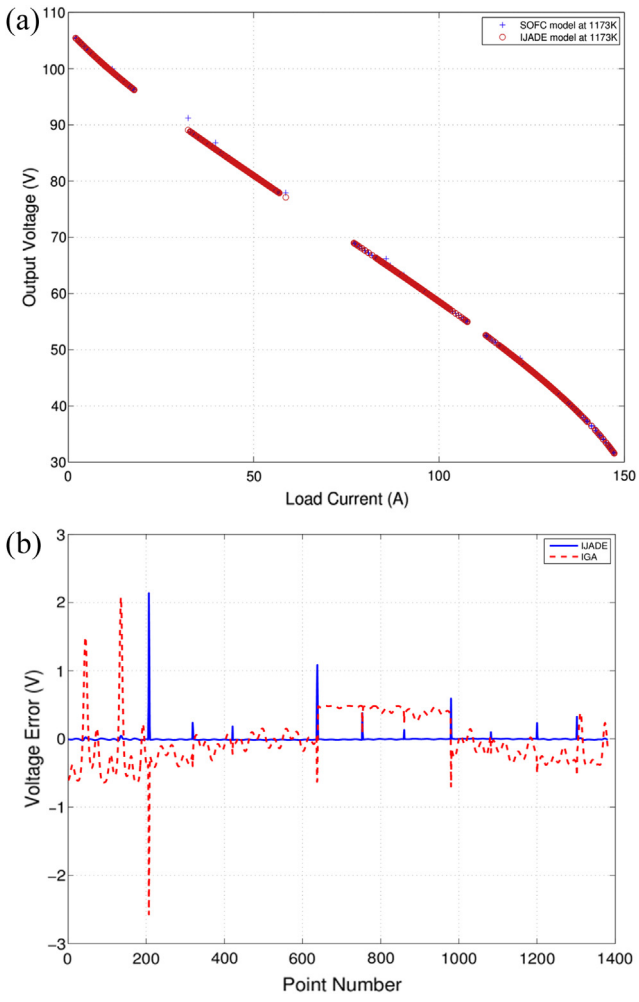

Fig. 4 – The results obtained by IJADE for the simulated data with interference. (a): V–I characteristics and (b): voltage deviation.

With respect to the results at different temperatures, as shown in Table 2, it can be observed that IJADE resulted in extremely small MSE values for all simulated data sets. The R^2 values for all data sets were 1.00, which means that the data predicted by IJADE agrees well with the simulated data. In addition, from Fig. 1(a), it is clear that the predicted data obtained by IJADE are in good agreement with the simulated data at different temperatures. Moreover, as observed in Fig. 2(a), for the majority of the data points, the output voltage errors between the data obtained by IJADE and the simulated data are within (−0.05, 0.05) at the different temperatures. A careful look at the results reported in Table 2 shows that the values of five of the parameters (E_0, A, $I_{0,a}$, $I_{0,a}$, $I_{0,c}$, and R_{ohm}) gradually decrease as the temperature increases, whereas the values of the remaining two parameters (B and i_L) increase as the temperature increases.

The results obtained by IJADE at different pressures are reported in Table 3, Figs. 1 and 2(b). Similar to the results at different temperatures, the IJADE method is still capable of providing a good fit between the simulated data and the predicted data at different pressures, according to the MSE values, R^2 values, V–I characteristics, and output voltage errors. Unlike the results at different temperatures, the identified values for A, R_{ohm}, and B obtained by IJADE are nearly the same at different pressures. The parameter E_0 increases as the pressure increases, whereas the other three parameters ($I_{0,a}$, $I_{0,c}$, and I_L) show no significant differences at different pressures.

Table 5 – Identified parameters by the IJADE method for the experimental data of ASC-SOFC at different situations, where "case 1, 2, 4" mean dry H_2 at 873 K, 923 K, and 973 K, respectively; "case 3" means synthetic reformate at 923 K; "case 5" means dry H_2 at FU 20%; "case 6, 7, 8" respectively indicate synthetic reformate at FU 20%, FU 50%, and FU 50% + humidified air.

Parameter	Range	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
E_0 (V)	[0, 1.2]	1.1700	1.1550	1.1434	1.0056	1.1555	1.0066	1.0054	1.0060
A (V)	[0, 1]	0.0523	0.0666	0.0321	0.0177	0.0352	0.0225	0.0287	0.1049
$i_{0,c}$ (mA cm^{-2})	[0, 30]	2.2393	7.5040	2.2223	29.9999	1.7684	19.3617	29.9999	29.9999
$i_{0,a}$ (mA cm^{-2})	[0, 30]	10.3260	20.5281	30.0000	30.0000	10.4986	19.3618	30.0000	30.0000
R_{ohm} (kΩ cm^2)	[0, 1]	5.22E-15	1.06E-16	0.0018	0.0017	0.0011	5.65E-16	5.89E-04	1.94E-10
B (V)	[0, 1]	0.0026	2.13E-15	0.0652	7.00E-15	7.97E-15	0.0988	0.2739	0.1107
i_L (mA cm^{-2})	[0, 200]	170.5147	173.5736	200.0000	192.1762	172.8100	120.6765	200.0000	199.9992
MSE (V^2)		4.04E-06	1.44E-06	2.38E-06	2.51E-06	1.06E-06	5.73E-07	1.50E-07	1.38E-06
R^2		0.9996	0.9997	0.9998	0.9994	0.9997	0.9992	0.9999	0.9995
Running time (s)		0.0550	0.0780	0.0780	0.0700	0.0570	0.0560	0.0540	0.0560

4.3. *Results comparison to simulated data with interference*

In the previous section, interference was not used in the load current in the simulated data. In this section, to further evaluate the effectiveness of IJADE, the method is used to identify unknown parameters of a simple electrochemical model for simulated data with interference. The simulated data are obtained from Ref. [13], and the load current with the wave signal and step signal is plotted in Fig. 3. This data set contains approximately 1400 *V–I* data points. The parameters identified by IJADE are described in Table 4. In addition, the *V–I* characteristics and output voltage errors are shown in Fig. 4. According to the results, it can be observed that the IJADE method still results in extremely good agreement between the simulated data with interference and the predicted data. The MSE and R^2 values are 0.0066132 and 1.0, respectively. Furthermore, from Fig. 4(b), it is clear that IJADE is more stable than IGA [11]. Only the step signal point shows large deviations between the simulated data and the predicted data obtained by IJADE. However, the proposed IJADE method is able to rapidly obtain extremely small deviations after the step signal, which means that IJADE can rapidly and accurately identify the parameters of an SOFC model.

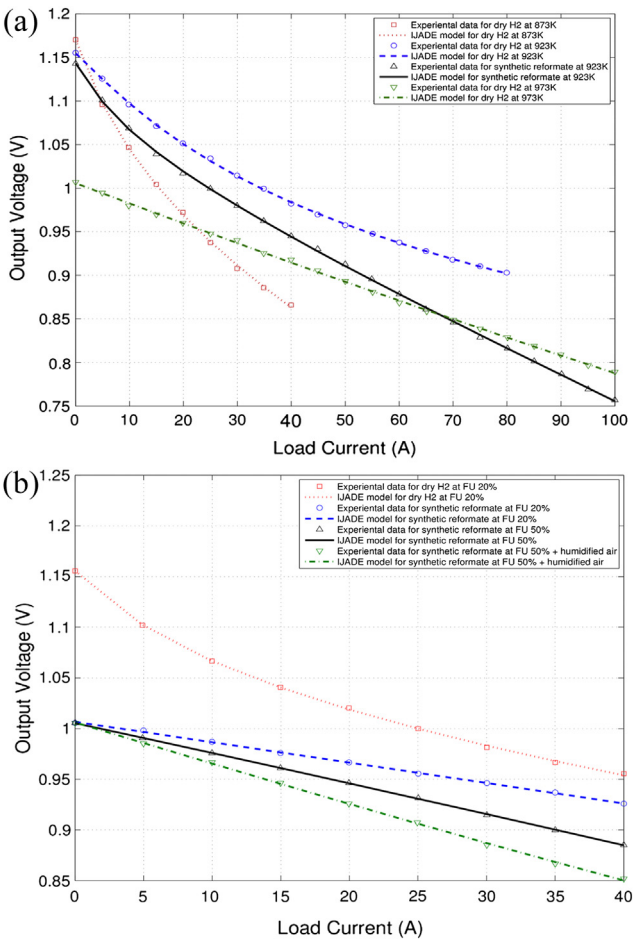

Fig. 5 – Comparisons between the experimental data of ASC-SOFC and the data obtained by IJADE at different temperature ((a)) and different fuel utilization ((b)).

4.4. *Results comparison to experimental data*

Through the results on the simulated data with and without interference in the previous sections, the effectiveness and efficiency of IJADE for parameter identification of a simple SOFC electrochemical model are validated. To further validate IJADE, in this section, two different types of experimental *V–I* data are used. The first data set is obtained from Ref. [42], which was generated by an Elcogen 10 × 10 cm^2 ASC-10B planar single cell (ASC-SOFC). The second data set was

Table 6 – Identified parameters by the IJADE method for the experimental data of HUST-SOFC at different temperature.

Parameter	Range	923 K	973 K	1023 K	1073 K
E_0 (V)	[0, 1.2]	1.1540	1.1780	1.1799	1.1643
A (V)	[0, 1]	0.0624	0.0311	0.0627	0.0421
$I_{0,c}$ (mA cm^{-2})	[0, 30]	0.4919	0.0873	0.5615	0.4736
$I_{0,a}$ (mA cm^{-2})	[0, 30]	4.7048	0.0012	30.0000	3.5078
R_{ohm} (kΩ cm^2)	[0, 1]	0.0004	3.0392	1.08E-17	4.61E-18
B (V)	[0, 1]	0.0132	0.1151	0.0061	0.0092
i_L (mA cm^{-2})	[0, 200]	78.4898	169.5794	80.6032	79.7977
MSE (V^2)		7.78E-06	1.87E-07	6.53E-06	2.50E-06
R^2		0.9997	1.0000	0.9995	0.9998
Running time (s)		0.0800	0.0770	0.0800	0.0760

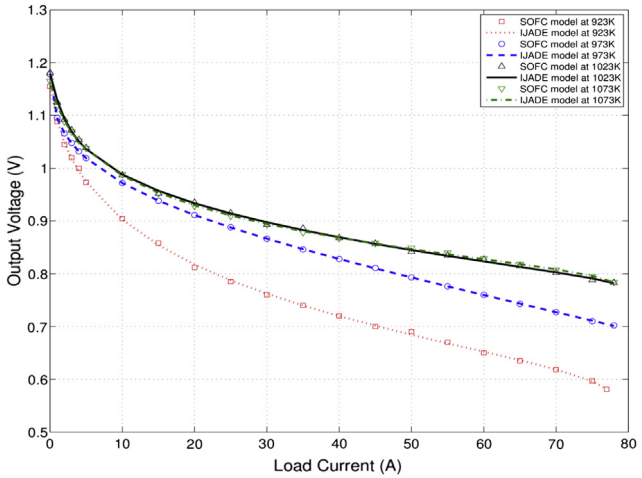

Fig. 6 – Comparisons between the experimental data of HUST-SOFC and the data obtained by IJADE at different temperatures.

provided by the SOFC Research & Development Center of Huazhong University of Science and Technology, which was generated by a planar single cell (HUST-SOFC).

4.4.1. *Experimental data of the ASC-SOFC*

The data contain eight data sets at different temperatures and different fuel uses. A more detailed description of the data can be found in Ref. [42]. The identified parameters from IJADE are tabulated in Table 5. Additionally, a comparison of the *V–I* characteristics of the experimental data and data obtained by IJADE are plotted in Fig. 5. Again, from the results, the predicted data obtained by the IJADE method are in extremely good agreement with the experimental data for all cases. The R^2 values in all cases are greater than 0.999, indicating a good fit between the experimental data and the predicted data obtained by IJADE.

4.4.2. *Experimental data of the HUST-SOFC*

For the HUST-SOFC, there are four data sets at different temperatures (923 K, 973 K, 1023 K, and 1273 K). Each data set contains 21 *V–I* data points. The identified parameters and the *V–I* characteristics obtained by IJADE are shown in Table 6 and Fig. 6, respectively. Similar to the results for the ASC-SOFC, IJADE shows extremely promising results in terms of MSE values, R^2 values, and *V–I* characteristics at different temperatures.

To summarize, according to the results obtained by IJADE in both experimental data sets, ASC-SOFC and HUST-SOFC, it can be observed that IJADE is an extremely efficient method that can identify unknown parameters of SOFC models. When the parameters of a simple electrochemical model are precisely identified by IJADE, this simple model can fit the experimental data of different SOFC models well.

4.5. *Statistical comparisons among different EAs*

In the above sections, the effectiveness and efficiency of IJADE were verified in solving parameter identification problems of a simple SOFC electrochemical model using both simulated and experimental data. However, there is another issue that must be addressed: is the IJADE method better than the original JADE method and other optimization techniques? To answer this question, in this section, IJADE is compared with other EAs in solving parameter identification problems of an SOFC model. The compared algorithms include five advanced DE variants (jDE [39], SaDE [43], CoDE [44], DEGL [45], and JADE [33]), the real-coded genetic algorithm (rcGA) [46], the artificial bee colony (ABC) [47], and the comprehensive learning particle swarm optimizer (CLPSO) [48]. These eight algorithms were selected due to their promising performance obtained from benchmark problems. To make a fair comparison among the different algorithms, all algorithms had the same population size ($Np = 50$) and the same maximal NFEs (Max_NFEs = 15,000). The other algorithm parameters of the eight algorithms were set to the same values used in their original studies. Because all of the nine algorithms are stochastic algorithms, for each algorithm, over 100 independent runs were performed for each data set to make the comparison statistically meaningful. The mean and standard deviation values of the MSE values for over 100 runs were recorded. For the sake of brevity, only the simulated data at the different temperatures of the SOFC model presented in Ref. [5] are used in this section. The results are shown in Table 7, where the overall best and second best results of all the algorithms are highlighted in gray boldface and boldface, respectively. To

Table 7 – Comparison of the MSE values of different EAs for the simulated data at different temperatures. All results are averaged over 100 independent runs. In "$A_{\pm B}$", "A" and "B" indicate the mean value and standard deviation value of MSE, respectively.

Algorithm	1073 K	1123 K	1173 K	1223 K	1273 K
rcGA	6.4035E+04$_{\pm 5.0980E+04}$ [a]	6.6346E+04$_{\pm 5.1886E+04}$ [a]	6.8025E+04$_{\pm 5.2500E+04}$ [a]	6.9473E+04$_{\pm 5.3106E+04}$ [a]	7.0487E+04$_{\pm 5.3507E+04}$ [a]
ABC	1.1040E+00$_{\pm 8.2475E-01}$ [a]	9.5623E-01$_{\pm 6.3588E-01}$ [a]	9.6213E-01$_{\pm 8.1366E-01}$ [a]	9.1598E-01$_{\pm 5.6534E-01}$ [a]	1.1565E+00$_{\pm 9.8551E-01}$ [a]
CLPSO	3.3859E-01$_{\pm 9.3841E-02}$ [a]	3.5827E-01$_{\pm 1.1246E-01}$ [a]	3.7487E-01$_{\pm 1.4812E-01}$ [a]	4.3247E-01$_{\pm 1.5824E-01}$ [a]	4.6585E-01$_{\pm 2.2511E-01}$ [a]
jDE	2.2612E-02$_{\pm 2.1915E-02}$ [a]	1.7587E-02$_{\pm 1.6684E-02}$ [a]	1.7869E-02$_{\pm 1.7869E-02}$ [a]	1.5411E-02$_{\pm 1.7592E-02}$ [a]	1.0299E-02$_{\pm 9.8444E-03}$ [a]
SaDE	3.8698E-03$_{\pm 6.3575E-03}$ [a]	**1.8810E-03$_{\pm 3.5749E-03}$** [a]	**1.8533E-03$_{\pm 4.2617E-03}$** [a]	**5.4744E-04$_{\pm 2.2481E-03}$** [a]	**2.3077E-04$_{\pm 6.5245E-04}$** [a]
CoDE	1.3967E-01$_{\pm 5.7710E-02}$ [a]	1.2525E-01$_{\pm 4.8248E-02}$ [a]	1.1466E-01$_{\pm 4.5674E-02}$ [a]	1.2015E-01$_{\pm 5.3394E-02}$ [a]	1.0160E-01$_{\pm 4.6336E-02}$ [a]
DEGL	2.7115E-02$_{\pm 3.9087E-02}$ [a]	2.8111E-02$_{\pm 5.0496E-02}$ [a]	3.1728E-02$_{\pm 4.9533E-02}$ [a]	3.6366E-02$_{\pm 7.2276E-02}$ [a]	3.6221E-02$_{\pm 7.2765E-02}$ [a]
JADE	**1.4298E-03$_{\pm 9.0393E-05}$** [a]	3.2831E-03$_{\pm 2.2484E-02}$ [a]	2.0503E-03$_{\pm 1.1348E-02}$ [a]	9.2346E-04$_{\pm 8.4449E-03}$ [a]	1.7357E-03$_{\pm 1.0067E-02}$ [a]
IJADE	**1.4159E-03$_{\pm 3.5109E-09}$**	**6.0340E-04$_{\pm 2.7094E-10}$**	**1.8655E-04$_{\pm 1.1988E-07}$**	**7.3925E-05$_{\pm 3.5917E-12}$**	**5.6214E-05$_{\pm 1.1131E-10}$**

[a] Indicates IJADE is significantly better than its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

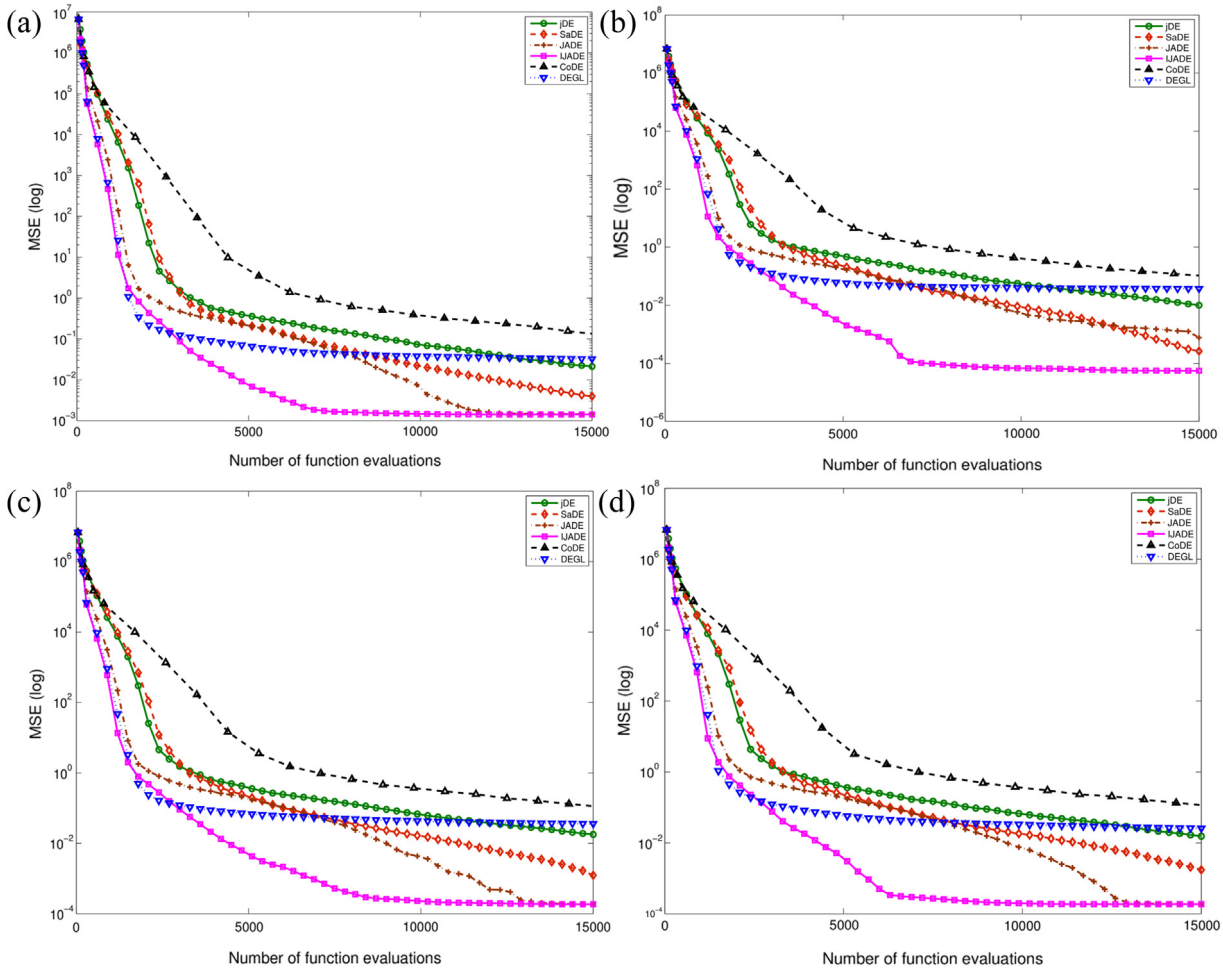

Fig. 7 – Comparison of the convergence speed among different DE variants for the simulated data at different situations. (a) at 1073 K, (b) at 1273 K, (c) at 3 atm, and (d) at 7 atm.

compare the significance between two algorithms, the paired Wilcoxon signed-rank test [49] was used. In addition, the convergence speeds of the different algorithms were also compared, as shown in Fig. 7.

Based on the results reported in Table 7, it can be clearly observed that the IJADE method consistently provided better average MSE values than all of the other eight EAs for all cases. With respect to the standard deviation values, it can be observed that IJADE obtained the smallest standard deviation values for all cases, indicating that IJADE is the most robust method among the nine algorithms. According to the Wilcoxon's test results, IJADE significantly outperformed the other eight EAs for all simulated data at different temperatures. In addition, from Fig. 7, it is clear that IJADE converged the fastest compared with the other algorithms overall. Although the DEGL algorithm is able to obtain the fastest convergence speed in the early stage, that method stagnates quickly.

From Table 7, the six DE variants are better than rcGA, ABC, and CLPSO. IJADE is the best of the nine algorithms, followed by SaDE, JADE, and DEGL, while rcGA is the worst. It is worth noting that although the MSE values of SaDE and JADE are extremely close to those of IJADE, because the simulated data were generated from the SOFC model provided in Ref. [41], no information is available regarding the accurate values of the parameters of the simple electrochemical model. Therefore, any reduction in the objective function value is significant because it results in an improvement of the knowledge of the real values of the parameters.

In general, when compared with the other DE variants and other EAs, the superiority of IJADE is indicated in terms of the mean MSE values, robustness, and convergence speed. Moreover, the two improvements (i.e., the ranking-based vector selection and the crossover rate repairing technique) of the JADE method worked extremely well when comparing the results of JADE and IJADE.

5. Conclusions

Apart from developing a new SOFC model, in this study, an efficient IJADE method is proposed to solve the parameter identification problems of SOFC models with greater speed and accuracy. The IJADE method is an improved version of

JADE with two improvements, i.e., the ranking-based vector selection and the crossover rate repairing technique. These improvements are simple yet efficient, and combining them with the parameter adaptation technique previously proposed in JADE makes IJADE a powerful alternative for complex real-world applications. The effectiveness and efficiency of IJADE were validated by identifying unknown parameters of a simple SOFC electrochemical model using both simulated and experimental data. The superiority of IJADE was also verified after comparing it with other state-of-the-art DE variants and other EAs.

Even with a simple SOFC electrochemical model, when its parameters are accurately identified by the IJADE method, it can fit both simulated and experimental data extremely well, as demonstrated by the results obtained in this work. This result motivates us to apply IJADE to solve other complex optimization problems of fuel cell models, such as parameter optimization of ANN-based models [21]. In addition, the IJADE method can be used in an RBF neural network model, as presented in Ref. [19], for predictive control of an SOFC. In future work, these expectations will be verified.

The source code of IJADE can be obtained from the first author upon request.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China under Grant no. 61203307 and 61075063, the Fundamental Research Funds for the Central Universities at China University of Geosciences (Wuhan) under Grant no. CUG130413 and CUG090109, and the Research Fund for the Doctoral Program of Higher Education under Grant no. 20110145120009.

REFERENCES

[1] Larminie J, Dicks A. Fuel cell systems explained. 2nd ed. Wiley; 2003.
[2] Wang K, Hissel D, Pra M, Steiner N, Marra D, Sorrentino M, et al. A review on solid oxide fuel cell models. Int J Hydrogen Energy 2011;36(12):7212–28.
[3] Hall D, Colclaser R. Transient modeling and simulation of a tubular solid oxide fuel cell. IEEE Transac Energy Convers 1999;14(3):749–53.
[4] Chan S, Khor K, Xia Z. A complete polarization model of a solid oxide fuel cell and its sensitivity to the change of cell component thickness. J Power Sources 2001;93(12):130–40.
[5] Wang C, Nehrir M. A physically based dynamic model for solid oxide fuel cells. IEEE Transac Energy Convers 2007;22(4):887–97.
[6] Kakac S, Pramuanjaroenkij A, Zhou XY. A review of numerical modeling of solid oxide fuel cells. Int J Hydrogen Energy 2007;32(7):761–86.
[7] Yang J, Li X, Mou HG, Li J. Predictive control of solid oxide fuel cell based on an improved Takagi–Sugeno fuzzy model. J Power Sources 2009;193(2):699–705.
[8] Chakraborty UK. Static and dynamic modeling of solid oxide fuel cell using genetic programming. Energy 2009;34(6):740–51.
[9] Yang J, Li X, Mou HG, Li J. Control-oriented thermal management of solid oxide fuel cells based on a modified Takagi–Sugeno fuzzy model. J Power Sources 2009;188(2):475–82.
[10] Milewski J, Swirski K, Santarelli M, Leone P. Advanced methods of solid oxide fuel cell modeling. Springer; 2011.
[11] Marzooghi H, Raoofat M, Dehghani M, Elahi G. Dynamic modeling of solid oxide fuel cell stack based on local linear model tree algorithm. Int J Hydrogen Energy 2012;37(5):4367–76.
[12] He Z, Li H, Birgersson E. Reduced model for the planar solid oxide fuel cell. Comput Chem Eng 2013;52:155–67.
[13] Yang J, Li X, Jiang JH, Jian L, Zhao L, Jiang JG, et al. Parameter optimization for tubular solid oxide fuel cell stack based on the dynamic model and an improved genetic algorithm. Int J Hydrogen Energy 2011;36(10):6160–74.
[14] Shi J, Xue X. Inverse estimation of electrode microstructure distributions in NASA bi-electrode supported solid oxide fuel cells. Chem Eng J 2012;181:607–13.
[15] Gogoi T, Das R. Inverse analysis of an internal reforming solid oxide fuel cell system using simplex search method. App Math Model 2013;37(10–11):6994–7015.
[16] Wu X, Zhu X, Cao G, Tu H. Modeling a SOFC stack based on GA-RBF neural networks identification. J Power Sources 2007;167(1):145–50.
[17] Chaichana K, Patcharavorachot Y, Chutichai B, Saebea D, Assabumrungrat S, Arpornwichanop A. Neural network hybrid model of a direct internal reforming solid oxide fuel cell. Int J Hydrogen Energy 2012;37(3):2498–508.
[18] Back T. Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford, UK: Oxford University Press; 1996.
[19] Wu X, Zhu X, Cao G, Tu H. Predictive control of SOFC based on a GA-RBF neural network model. J Power Sources 2008;179(1):232–9.
[20] Li Y, Shen J, Lu J. Constrained model predictive control of a solid oxide fuel cell based on genetic optimization. J Power Sources 2011;196(14):5873–80.
[21] Bozorgmehri S, Hamedi M. Modeling and optimization of anode-supported solid oxide fuel cells on cell parameters via artificial neural network and genetic algorithm. Fuel Cells 2012;12(1):11–23.
[22] Outeiro M, Chibante R, Carvalho A, de Almeida A. A parameter optimized model of a proton exchange membrane fuel cell including temperature effects. J Power Sources 2008;185(2):952–60.
[23] Ye M, Wang X, Xu Y. Parameter identification for proton exchange membrane fuel cell model using particle swarm optimization. Int J Hydrogen Energy 2009;34(2):981–9.
[24] Ohenoja M, Leiviskä K. Validation of genetic algorithm results in a fuel cell model. Int J Hydrogen Energy 2010;35(22):12618–25.
[25] Askarzadeh A, Rezazadeh A. A grouping-based global harmony search algorithm for modeling of proton exchange membrane fuel cell. Int J Hydrogen Energy 2011;36(8):5047–53.
[26] Chakraborty UK, Abbott TE, Das SK. PEM fuel cell modeling using differential evolution. Energy 2012;40(1):387–99.
[27] Yang S, Wang N. A novel P systems based optimization algorithm for parameter estimation of proton exchange membrane fuel cell model. Int J Hydrogen Energy 2012;37(10):8465–76.
[28] Zhang L, Wang N. An adaptive RNA genetic algorithm for modeling of proton exchange membrane fuel cells. Int J Hydrogen Energy 2013;38(1):219–28.

[29] Zhang W, Wang N, Yang S. Hybrid artificial bee colony algorithm for parameter estimation of proton exchange membrane fuel cell. Int J Hydrogen Energy 2013;38(14):5796–806.
[30] Leung Y-W, Wang Y. An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Transac Evol Comput 2001;5(1):41–53.
[31] Storn R, Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 1997;11(4):341–59.
[32] Das S, Suganthan PN. Differential evolution: a survey of the state-of-the-art. IEEE Transac Evol Comput 2011;15(1):4–31.
[33] Zhang J, Sanderson AC. JADE: adaptive differential evolution with optional external archive. IEEE Transac Evol Comput 2009;13(5):945–58.
[34] I. EG&G Technical Services. Fuel cell handbook. 7th ed. U.S. Department of Energy; 2004.
[35] Gebregergis A, Pillay P, Bhattacharyya D, Rengaswemy R. Solid oxide fuel cell modeling. *IEEE T Ind Electron*;56(1):139–48.
[36] Chan S, Low C, Ding O. Energy and exergy analysis of simple solid-oxide fuel-cell power systems. J Power Sources 2002;103(2):188–200.
[37] Zhang J, Sanderson AC. Adaptive differential evolution: a robust approach to multimodal problem optimization. Berlin: Springer-Verlag; 2009.
[38] Liu J, Lampinen J. A fuzzy adaptive differential evolution algorithm. Soft Comput 2005;9(6):448–62.
[39] Brest J, Greiner S, Boskovic B, Mernik M, Zumer V. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Transac Evol Comput 2006;10(6):646–57.
[40] Gong W, Cai Z. Differential evolution with ranking-based mutation operators. IEEE Transac Cybern 2013;43(6):2066–81.
[41] Wang C, Nehrir MH. Dynamic models for PEMFs and tubular SOFCs http://www.coe.montana.edu/ee/fuelcell/; 2013.
[42] Elcogen. SOFC fuel cell technical data http://www.elcogen.com/single-cell-technical-data; 2013.
[43] Qin AK, Huang VL, Suganthan PN. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Transac Evol Comput 2009;13(2):398–417.
[44] Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters. IEEE Transac Evol Comput 2011;15(1):55–66.
[45] Das S, Abraham A, Chakraborty UK, Konar A. Differential evolution using a neighborhood-based mutation operator. IEEE Transac Evol Comput 2009;13(3):526–53.
[46] Herrera F, Lozano M. Gradual distributed real-coded genetic algorithms. IEEE Transac Evol Comput 2000;4(1):43–63.
[47] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 2007;39(3):459–71.
[48] Liang JJ, Qin AK, Suganthan PN, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transac Evol Comput 2006;10(3):281–95.
[49] Shaw C, Williams K, Assassa R. Patients' views of a new nurse-led continence service. *J Clin Nurs*;9(4):574–84.