Accelerating parameter identification of proton exchange membrane fuel cell model with ranking-based differential evolution☆

Wenyin Gong*[a], Zhihua Cai[a]

[a]*School of Computer Science,*
China University of Geosciences, Wuhan 430074, P.R. China

Abstract

Parameter identification of proton exchange membrane (PEM) fuel cell model is a very active area of research. Generally, it can be treated as a numerical optimization problem with complex nonlinear and multi-variable features. Differential evolution (DE), which has been successfully used in various fields, is a simple yet efficient evolutionary algorithm for global numerical optimization. In this paper, with the objective of accelerating the process of parameter identification of PEM fuel cell models and reducing the necessary computational efforts, we firstly present a generic and simple ranking-based mutation operator for the DE algorithm. Then, the ranking-based mutation operator is incorporated into five highly-competitive DE variants to solve the PEM fuel cell model parameter identification problems. The main contributions of this work are the proposed ranking-based DE variants and their application to the parameter identification problems of PEM fuel cell models. Experiments have been conducted by using both the simulated voltage-current data and the data obtained from the literature to validate the performance of our approach. The results indicate that the ranking-based DE methods provide better results with respect to the solution quality, the convergence rate, and the success rate compared with their corresponding original DE methods. In addition, the voltage-current characteristics obtained by our approach are in good agreement with the original voltage-current curves in all cases.

Key words: Proton exchange membrane fuel cell, parameter identification, optimization, differential evolution, ranking-based mutation operator

1. Introduction

Due to the high energy efficiency, superior durability, low emission, high scalability, good transient responses of the fuel cell (FC) technology, it has received a heightened research focus in recent years [1, 2]. Among various types of fuel cells, the proton exchange membrane fuel cells (PEMFCs) have obtained an increasing interest for both mobile and stationary applications because of their high efficiency, low noise, no waste, low operating temperature, low pressure, etc [3]. Also, due to their advantages, they can be used to build hybrid energy generation systems, such as wind/hydrogen hybrid systems to provide consistent sustainable energy supply [4].

Within different fields of research in PEMFC, the modeling of PEMFC has attracted considerable attention among researchers of different backgrounds, and different models of PEMFC are available in the literature [5, 6, 7, 8]. Mo *et al.* [9] classified different PEMFC models into two approaches: i) mechanistic models and ii) models based on empirical or semi-empirical equations. However, no matter what type of models, the parameters of models need to be identified in order to improve the accuracy of the models and make the models indicate the actual PEMFC performance better [9, 10]. For example, the parameter settings of the hydrogen flow rate, air flow rate, inlet hydrogen pressure, membrane dehydration, catalyst layer flooding, mass transport, and fluid flow regimes affect the performance

☆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. CUG090109, and the Research Fund for the Doctoral Program of Higher Education under Grant No. 20110145120009.

*Corresponding author. Tel: +86-27-67883716.

Email addresses: wenyingong@yahoo.com; wygong@cug.edu.cn (Wenyin Gong), zhcai@cug.edu.cn (Zhihua Cai)

of PEMFC models significantly [11, 12]. Identifying the parameters of PEMFC models can be treated as numerical optimization problems. However, since the PEMFC system is a complex nonlinear and multi-variable system, the parameter identification of PEMFC models is hard to be tractable by conventional methods. Therefore, it is essential to identify the parameters of PEMFC models using advanced optimization techniques.

In recent years, the use of heuristic optimization techniques for parameter identification of PEMFC models have received increasing interest, such as genetic algorithms (GAs) [9, 13, 14], simulated annealing [15, 16], particle swarm optimization (PSO) [17, 18], harmony search [19, 20, 3], seeker optimization algorithm [21], artificial immune system [22], P systems based optimization algorithm [23]. Most recently, differential evolution is also used to solve the parameter identification of PEMFC models [24]. However, in order to efficiently and fast solve the parameter identification problems in PEMFC models, it is necessary to investigate more efficient optimization techniques to reduce the necessary computational efforts to achieve an optimal design [25].

Differential evolution (DE), proposed by Storn and Price [26], is a simple, efficient, and versatile numerical optimization algorithm. The advantages are its simple structure, ease of use, speed, and robustness. Due to these advantages, DE has been successfully applied in diverse fields, such as engineering design, digital filter design [27, 28], optimal power flow [29], simulation of solar-thermal refrigeration systems [30], hydrothermal generation scheduling [31, 32], and so on. With the objective of accelerating the process of parameter identification of PEMFC models and reducing the necessary computational efforts, in this work, a generic and simple ranking-based mutation operator is presented for the DE algorithm. The ranking-based mutation operator does not increase the complexity of the original DE algorithm significantly, and it can be combined with most of advanced DE variants. Based on this consideration, it is incorporated into five highly-competitive DE variants, *i.e.*, jDE [33], SaDE [34], JADE [35], CoDE [36], and DEGL [37]. The five ranking-based DE variants together with the five original DE variants are validated by using the simulated voltage-current data of PEMFC model and the data obtained from [9]. Numerical results indicate that the ranking-based DE methods provide better results with respect to the solution quality, the convergence rate, and the success rate compared with their corresponding original DE methods. In addition, the voltage-current characteristics obtained by our approach are in good agreement with the original voltage-current curves in all cases. Thus, the ranking-based DE approaches can be an efficient alternative for other complex parameter identification problems of FC models.

The rest of this paper is organized as follows. Section 2 briefly describes the PEMFC stack model used in this work and the objective function to be optimized. Next, in Section 3 we introduces the original DE algorithm in brief. In Section 4 our proposed ranking-based mutation operator is presented in detail, followed by the experiments and discussions in Section 5. Finally, Section 6 draws the conclusions from this work.

2. Problem Formulation

In this section, we first briefly introduce the PEMFC stack model used in this work. Then, the objective function to be optimized is specified.

2.1. PEMFC stack model

In this work, the PEMFC stack model presented in [9] is used. For n cells connected in series to form a stack, the terminal voltage of the stack can be calculated by [38],

$$V_{\mathrm{s}} = n \cdot V_{\mathrm{FC}} \tag{1}$$

where V_{FC} is the output voltage of a single cell, which can be formulated as [7]

$$V_{\mathrm{FC}} = E_{\mathrm{Nernst}} - V_{\mathrm{act}} - V_{\mathrm{ohm}} - V_{\mathrm{con}} \tag{2}$$

E_{Nernst} is the thermodynamic potential defined by

$$\begin{aligned} E_{\mathrm{Nernst}} =& 1.229 - 0.85 \times 10^{-3} \cdot (T - 298.15) \\ &+ 4.3085 \times 10^{-5} \cdot T \cdot \ln\left(P^{*}_{\mathrm{H_2}} \sqrt{P^{*}_{\mathrm{O_2}}}\right) \end{aligned} \tag{3}$$

where T is the cell temperature (K), $P^*_{H_2}$ and $P^*_{O_2}$ are the hydrogen and oxygen partial pressures (atm), respectively. They are given by [5]

$$P^*_{H_2} = 0.5 \cdot RH_a \cdot P^{sat}_{H_2O} \cdot \left(\frac{1}{\frac{RH_a \cdot P^{sat}_{H_2O}}{P_a} \exp\left(\frac{1.635(i_{cell}/A)}{T^{1.334}}\right)} - 1 \right) \tag{4}$$

$$P^*_{O_2} = RH_c \cdot P^{sat}_{H_2O} \cdot \left(\frac{1}{\frac{RH_c \cdot P^{sat}_{H_2O}}{P_c} \exp\left(\frac{4.192(i_{cell}/A)}{T^{1.334}}\right)} - 1 \right) \tag{5}$$

where RH_a and RH_c are the relative humidity of vapor in the anode and cathode, P_a and P_c are the anode and cathode inlet pressures (atm), respectively. A is the effective electrode area (cm^2) and i_{cell} is the cell current (A). $P^{sat}_{H_2O}$ is the saturation pressure of water vapor (atm), which is defined as a function of the temperature T as follows [9, 24]

$$\begin{aligned} \log_{10}\left(P^{sat}_{H_2O}\right) =& 2.95 \times 10^{-2} \cdot (T - 273.15) \\ & - 9.19 \times 10^{-5} \cdot (T - 273.15)^2 \\ & + 1.44 \times 10^{-7} \cdot (T - 273.15)^3 - 2.18 \end{aligned} \tag{6}$$

According to [6], the activation overpotential V_{act}, including anode and cathode, can be expressed by the following formula

$$V_{act} = -\left[\xi_1 + \xi_2 \cdot T + \xi_3 \cdot T \cdot \ln\left(C^*_{O_2}\right) + \xi_4 \cdot T \cdot \ln\left(i_{cell}\right)\right] \tag{7}$$

where $\xi_1, \xi_2, \xi_3, \xi_4$ are the parametric coefficients for each cell model, and $C^*_{O_2}$ (mol/cm^3) is the concentration of oxygen in the catalytic interface of the cathode, given by [7, 9]

$$C^*_{O_2} = \frac{P^*_{O_2}}{5.08 \times 10^6 \cdot \exp\left(-498/T\right)} \tag{8}$$

The ohmic voltage drop V_{ohm} can be determined by the following expression [6]

$$V_{ohm} = i_{cell} \cdot (R_M + R_C) \tag{9}$$

where R_M is the equivalent membrane resistance to proton conduction, and R_C is the equivalent contact resistance to electron conduction. R_M is defined by [9]

$$R_M = \frac{\rho_M \cdot \ell}{A} \tag{10}$$

$$\rho_M = \frac{181.6 \cdot \left[1 + 0.03 \cdot \left(\frac{i_{cell}}{A}\right) + 0.062 \cdot \left(\frac{T}{303}\right) \cdot \left(\frac{i_{cell}}{A}\right)^{2.5}\right]}{\left[\lambda - 0.634 - 3 \cdot \left(\frac{i_{cell}}{A}\right)\right] \cdot \exp\left[4.18 \cdot \left(\frac{T-303}{T}\right)\right]} \tag{11}$$

where ρ_M is the membrane specific resistivity for the flow of hydrated protons ($\Omega \cdot$ cm), and ℓ is the thickness of the membrane (cm), which serves as the electrolyte of the cell. The parameter λ is an adjustable parameter with a possible range of $[10, 24]$.

The concentration overpotential V_{con} caused by the change in the concentration of the reactants at the surface of the electrodes as the fuel is calculated by [7]

$$V_{con} = -B \cdot \ln\left(1 - \frac{J}{J_{max}}\right) \tag{12}$$

where B (V) is a parametric coefficient, which depends on the cell and its operation state. J is the actual current density of the cell (A/cm^2), and J_{max} is the maximum value of J.

The purpose of parameter identification is to extract the unknown parameters of the PEMFC stack model so that a model can better fit a given PEMFC stack model. Similar to the work presented in [9, 13, 23], in this work, seven parameters, *i.e.*, $\xi_1, \xi_2, \xi_3, \xi_4, \lambda, R_C$, and B, will be identified by the DE algorithm. Other parameters of the PEMFC stack and the operation conditions are shown in Table 1.

Table 1: PEMFC stack parameters and operation conditions.

Parameter	Value
Number of cells in series n	24
Cell's effective active area A	27 cm^2
Nafion 115:5 mil ℓ	127 μm
Maximum current density $J_{\max}$	860 mA/cm^2
Relative humidity in anode RH_a	1
Relative humidity in cathode RH_c	1
Inlet Anode pressure P_a	3 bar
Inlet Cathode pressure P_c	5 bar
Stack temperature T	353.15 K

Table 2: Ranges of model parameter [13] and parameters used to generate the V_s [9, 24].

Parameter	Lower bound L_j	Upper bound U_j	actual value
ξ_1	-1.19969	-0.8532	-0.944957
ξ_2	0.001	0.005	0.00301801
ξ_3	3.6×10^{-5}	9.8×10^{-5}	7.401×10^{-5}
ξ_4	-2.60×10^{-4}	-9.54×10^{-5}	-1.88×10^{-4}
λ	10	24	23
R_c (Ω)	0.0001	0.0008	0.0001
B (V)	0.0136	0.5	0.02914489

2.2. Objective function

In order to identify the optimal values of the seven unknown parameters mentioned above by the optimization techniques, it needs to define a objective function to be optimized. In this work, the sum of the squared error (SSE) between the output voltage of the *actual* PEMFC stack and the model output voltage are used as the objective function [9]:

$$\min \quad f(\mathbf{x}) = \sum_{k=1}^{N} \left(V_{\mathrm{sm},k} - V_{\mathrm{so},k}\right)^2 \tag{13}$$

where $\mathbf{x} = \{\xi_1, \xi_2, \xi_3, \xi_4, \lambda, R_C, B\}$ is the vector of the unknown parameters, V_{sm} is the output voltage of the actual PEMFC stack, V_{so} is the model output voltage, and N is the number of the experimental data point.

3. Differential Evolution

The DE algorithm is initially proposed for the numerical optimization problems. The main procedure of DE are described as follows.

3.1. Initialization

The DE population consists of NP vectors. Initially, the population is generated at random. For example, for the i-th vector $\mathbf{x}_i$ it is initialized as follows:

$$x_{i,j} = L_j + \mathrm{rndreal}(0,1) \cdot \left(U_j - L_j\right) \tag{14}$$

where L_j and U_j are respectively the lower bound and upper bound of x_j, *i.e.*, $x_j \in [L_j, U_j]$. $i = 1, \cdots, NP$, $j = 1, \cdots, D$, and rndreal$(0,1)$ is a uniformly distributed random real number in $(0,1)$. In this work, for parameter identification of PEMFC model, $D = 7$, and the parameter ranges of the seven unknown parameters are shown in Table 2, which originally presented in [13] and also used in [24].

3.2. Mutation

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. There are many mutation strategies available in the literature [27, 35], the classical one is "DE/rand/1":

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{15}$$

where F is the mutation scaling factor, $r_1, r_2, r_3 \in \{1, \cdots, NP\}$ are mutually different integers randomly generated, and $r_1 \neq r_2 \neq r_3 \neq i$. In this work, we will try to improve the selection of vectors in the mutation operator to accelerate the process of parameter identification of PEMFC model.

3.3. Crossover

In order to diversify the current population, following mutation, DE employs 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* or *uniform* crossover performed on each component as follows:

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

where CR is the crossover rate and j_{rand} is a randomly generated integer within $\{1, D\}$. It is worth noting that there are other crossover operators in DE, such as the *exponential* crossover [27]. However, in this paper, we only focus on the binomial crossover mentioned above due to its promising performance obtained.

3.4. Selection

Finally, to keep the population size constant in the following generations, the selection operation is employed to determine whether the trial or the target vector survives to the next generation. In DE, the *one-to-one tournament selection* is used as follows:

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

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

4. Ranking-based mutation operator

In order to accelerate the parameter identification process of PEMFC model and reduce the necessary computational efforts (measured by the number of function evaluations to find an acceptable solution) to achieve an optimal design, in this work, we present the ranking-based mutation operator for the DE algorithm [39]. The key points of our approach are described in detail as follows.

4.1. Rankings Assignment

Firstly, the population is sorted in ascent order (*i.e.*, from the best to the worst) based on the fitness $f(\mathbf{x})$ of each vector. Then, the ranking R_i of the i-th vector is assigned as follows:

$$R_i = NP - i, \quad i = 1, 2, \cdots, NP \tag{18}$$

where NP is the population size. According to Equation (18), the best vector in the current population will obtain the highest ranking.

4.2. Selection Probability

After assigning the ranking for each vector, the selection probability p_i of the i-th vector $\mathbf{x}_i$ is calculated as

$$p_i = \left(\frac{R_i}{NP}\right)^2 \tag{19}$$

4.3. Vector Selection

After calculating the selection probability of each vector in Equation (19), the other issue is that in the mutation operator which vectors should be selected according to the selection probabilities. In this work, we select the *base* vector and the *terminal* point of the difference vector based on their selection probabilities, while other vectors in the mutation operator are selected randomly as the original DE algorithm. For example, for the "DE/rand/1" mutation the vectors are selected as shown in Algorithm 1. Note that the notation "$a == b$" indicates a is equal to b. From Algorithm 1 we can see that the vectors with higher rankings (or selection probabilities) are more likely to be chosen as the base vector or the terminal point in the mutation operator. Note that in Algorithm 1 we only illustrate the vector selection for "DE/rand/1", for other mutation operators the vector selection is similar to Algorithm 1.

Algorithm 1 Ranking-based vector selection for "DE/rand/1"

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

According to Algorithm 1, we can see that the only difference between the original DE mutation and the ranking-based mutation is that in the original DE mutation (see Equation (15)) r_1, r_2, r_3 are only selected randomly, while in our proposed ranking-based DE mutation r_1, r_2 are chosen according to their rankings. In summary, the ranking-based mutation operator has the following advantages:

1) It is very simple, generic, and easy to be implemented. In this way, it can be incorporated into most of DE variants. In this work, the ranking-based mutation operator is combined with five highly-competitive DE variants, *i.e.*, jDE [33], SaDE [34], JADE [35], CoDE [36], and DEGL [37] to solve the parameter identification problems of PEMFC models.

2) It does not significantly increase the complexity of the original DE variants. This makes our approach be suitable for real-world applications, such as the optimization problems in fuel cell models.

5. Experiments and discussions

In this section, the performance of our approach for parameter identification of PEMFC models is validated through both the simulated voltage-current data and the data obtained from [9]. Totally, 10 DE variants (jDE [33], SaDE [34], JADE [35], CoDE [36], DEGL [37], and their corresponding ranking-based variants) are executed. All algorithms are coded in standard C++[1]. The parameter settings for these DE variants are shown in Table 3. The maximal number of function evaluations (Max_NFEs) are set to $10,000$[2]. Because the DE algorithms belong to the stochastic algorithm, to make the comparison among different algorithms statistically meaningful, each problem is optimized over 100 independent runs. The programs are executed on the following platform: CPU: Inter Core i7-3770 3.40GHz; RAM: 8.00 GB; Operating system: Microsoft Windows 7 Home Edition; Compiler: Microsoft Visual C++ 6.0.

[1]The source codes of these methods can be obtained from the first author upon request.

[2]Note that we do not used the maximal generations as the termination condition, since for different algorithms the consumed number of function evaluations may be not the same in one generation.

Table 3: Parameter settings for all DE variants.

Algorithm	Parameter settings
jDE, rank-jDE	$NP = 100, \tau_1 = 0.1, \tau_2 = 0.1$ [33]
SaDE, rank-SaDE	$NP = 50, LP = 50$ [34]
JADE, rank-JADE	$NP = 100, p = 0.05, c = 0.1$ [35]
CoDE, rank-CoDE	$NP = 30$ [36]
DEGL, rank-DEGL	$NP = 10 \times D, CR = 0.9, F = 0.8$ [37]

Table 4: Numerical results (mean ± standard deviation) on simulated data with noise free. The target fitness $f(\mathbf{x}^*)$ is 0.

Algorithm	SSE		NFEs	time in seconds	SR	AR
jDE	0.0117176 ± 0.0074171		8233.33 ± 1225.49	**0.104** ± 0.007	0.45	1.47
rank-jDE	**5.06E-04** ± **7.49E-04**	+	**5592.00** ± **984.08**	0.108 ± 0.007	1.00	
SaDE	1.91E-07 ± 1.38E-06		3623.50 ± 387.00	**0.108** ± 0.006	1.00	1.51
rank-SaDE	**2.55E-12** ± **8.98E-12**	+	**2397.00** ± **328.36**	0.117 ± 0.008	1.00	
JADE	0.0016386 ± 0.0011805		6717.00 ± 1266.75	**0.108** ± 0.006	1.00	1.10
rank-JADE	**7.44E-04** ± **8.34E-04**	+	**6096.00** ± **1082.45**	0.114 ± 0.008	1.00	
CoDE	4.65E-05 ± 5.04E-05		4684.80 ± 766.04	**0.101** ± 0.008	1.00	1.57
rank-CoDE	**6.19E-08** ± **1.01E-07**	+	**2980.20** ± **415.61**	0.107 ± 0.005	1.00	
DEGL	1.81E-08 ± 1.29E-07		1492.40 ± 214.75	**0.112** ± 0.006	1.00	1.08
rank-DEGL	**5.06E-12** ± **1.81E-11**	+	**1388.10** ± **167.09**	0.122 ± 0.006	1.00	

Hereinafter, "+" indicates that ranking-based DE is significant better than its corresponding non-ranking-based DE according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

Similar to [24], in this work, the PEMFC stack model presented in Section 2.1 and its parameters and operation conditions shown in Table 1 are used. In addition, the ranges of the seven parameters to be identified are tabulated in Table 2. The actual values of parameters used to generate the simulate data of V_s are also shown in Table 2 [9, 24]. Both the simulated data (including noise free, low noise, and high noise) and the data obtained from [9] are used to evaluate the capability of our approach to identify the parameters of PEMFC model. The simulated data is used so as to measure the accuracy of the identified parameters by the optimization technique. The data obtained from the PEMFC literature is chosen to evaluate the practicability of the optimization technique for parameter identification when prior knowledge is not available for the PEMFC models.

5.1. Performance criteria

In order to compare the performance of different algorithms, in this work, we adopt the following performance criteria:

- **SSE** [24]: It is calculated by Equation (13) to measure the solution quality of a method obtained.
- **NFEs** [24]: It is used to record the number of function evaluations in each run for finding a solution satisfying $f(\mathbf{x}) - f(\mathbf{x}^*) \leq 1e-2$, where $f(\mathbf{x}^*)$ is the target fitness to be reached of a specific problem.
- **Success rate** (SR): It is equal to the number of success runs over total runs. A success run means that within Max_NFEs the algorithm finds a solution $\mathbf{x}$ satisfying $f(\mathbf{x}) - f(\mathbf{x}^*) \leq 1e-2$.
- **Acceleration rate** (AR) [40]: This criterion is used to compare the convergence speed between two algorithms. It is defined as follows:

$$AR = \frac{\text{ANFEs}_\text{A}}{\text{ANFEs}_\text{B}} \tag{20}$$

  where ANFEs_A is the average NFEs of algorithm A. $AR > 1$ indicates algorithm B converges faster than algorithm A.
- **The CPU time in seconds**: It records the running time (in seconds) of a method when the Max_NFEs is reached.
- **Convergence graphs**: The graphs show the averaged SSE performance of the total runs.

Table 5: Numerical results on simulated data with low noise. The target fitness $f(\mathbf{x}^*)$ is 0.29607692.

Algorithm	SSE		NFEs	time in seconds	SR	AR
jDE	0.3188408 ± 0.0096381		9175.00 ± **923.89**	**0.101** ± 0.008	0.08	1.31
rank-jDE	**0.2983047** ± **0.0016221**	+	**6979.00** ± 1086.31	0.109 ± 0.006	1.00	
SaDE	0.296128 ± 4.68E-05		4362.50 ± 516.90	**0.108** ± 0.005	1.00	1.49
rank-SaDE	**0.2960771** ± **2.14E-07**	+	**2928.00** ± **330.47**	0.116 ± 0.008	1.00	
JADE	0.3012747 ± 0.0030127		8317.20 ± 1048.67	**0.108** ± 0.006	0.93	1.12
rank-JADE	**0.2987938** ± **0.0016073**	+	**7445.00** ± **902.56**	0.114 ± 0.007	1.00	
CoDE	0.2969736 ± 4.45E-04		5719.80 ± 734.80	**0.100** ± 0.008	1.00	1.60
rank-CoDE	**0.2960983** ± **1.14E-05**	+	**3573.30** ± **453.47**	0.107 ± 0.006	1.00	
DEGL	0.2960806 ± 7.28E-06		1866.20 ± 303.45	**0.111** ± 0.005	1.00	1.13
rank-DEGL	**0.296078** ± **4.20E-06**	+	**1652.70** ± **258.66**	0.122 ± 0.006	1.00	

Table 6: Numerical results on simulated data with high noise. The target fitness $f(\mathbf{x}^*)$ is 1.19101079.

Algorithm	SSE		NFEs	time in seconds	SR	AR
jDE	1.2243306 ± 0.0147351		9850.00 ± **212.13**	**0.102** ± 0.008	0.02	1.31
rank-jDE	**1.1939761** ± **0.0019916**	+	**7539.39** ± 1032.22	0.108 ± 0.005	0.99	
SaDE	1.1910959 ± 7.72E-05		4702.00 ± 463.79	**0.108** ± 0.006	1.00	1.52
rank-SaDE	**1.1910112** ± **1.33E-06**	+	**3086.00** ± **280.70**	0.117 ± 0.008	1.00	
JADE	1.1986425 ± 0.0043916		8689.61 ± **724.49**	**0.108** ± 0.005	0.77	1.05
rank-JADE	**1.1956602** ± **0.0024067**	+	**8237.76** ± 884.40	0.114 ± 0.007	0.98	
CoDE	1.1923034 ± 7.88E-04		6440.70 ± 820.37	**0.101** ± 0.008	1.00	1.66
rank-CoDE	**1.191045** ± **2.24E-05**	+	**3882.00** ± **495.60**	0.107 ± 0.005	1.00	
DEGL	1.1910227 ± 2.25E-05		1960.70 ± 341.12	**0.111** ± 0.004	1.00	1.16
rank-DEGL	**1.191013** ± **8.05E-06**	+	**1687.70** ± **203.39**	0.119 ± 0.008	1.00	

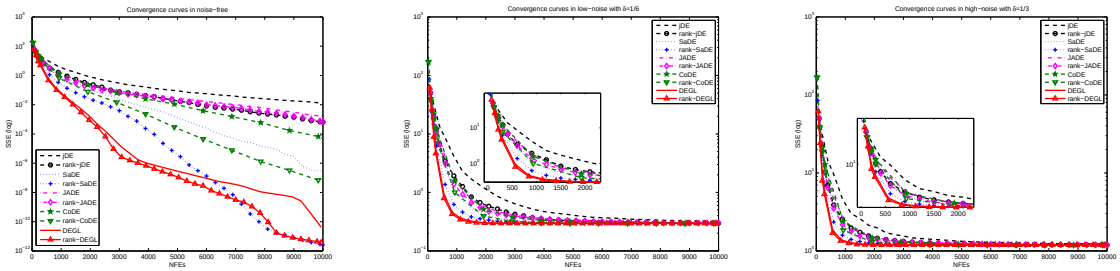
5.2. Parameter identification with simulated data

In this section, the identifying power of our approach is accessed by the simulated data. The actual values and ranges of the seven parameters are presented in Table 2 as reported in [24]. These parameters are used to generate the noise free data V_s via Equation (1). Then, similar to [24], the noise data are obtained as follows:

$$V_{so} = V_s + \mathcal{N}(0, \delta) \tag{21}$$

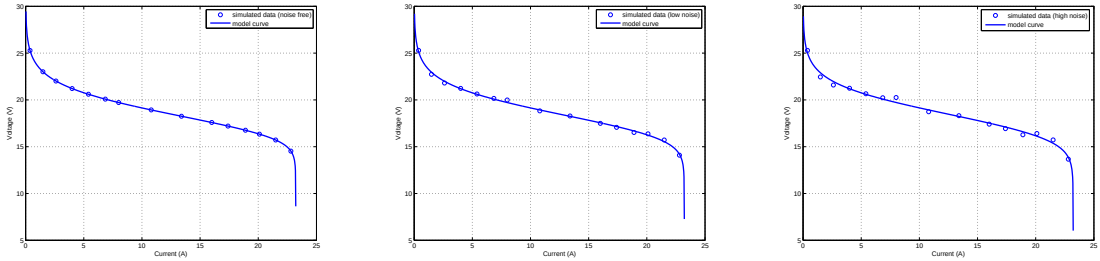
where $\mathcal{N}(0, \delta)$ indicates a Gaussian noise with mean zero and standard deviation δ. In this work, the simulated data with noise free, low noise ($\delta = 1/6$), and high noise ($\delta = 1/3$) are used. In Equation (2), the value of E_{Nernst} is set to be 1.197374 V.

The numerical results for all DE variants are reported in Tables 4, 5, and 6 for the simulated data with noise free, low noise, and high noise, respectively. All results are averaged over 100 independent runs. In these tables, the *AR* values are calculated by Equation (20), where "A" means the non-ranking-based DE, and "B" means the ranking-based DE. $AR > 1$ indicates that the ranking-based DE converges faster than its corresponding non-ranking-based DE. The overall best results within all DE variants are highlighted in **grey boldface**. The **boldface** means that ranking-based DE is better than its corresponding non-ranking DE. In addition, the convergence graphs are plotted in



(a) (b) (c)

Figure 1: Convergence graphs of all DE variants on simulated data. 1(a) for noise free; 1(b) for low noise; and 1(c) for high noise.



(a) (b) (c)

Figure 2: Comparisons between the simulated data and the model curves obtained from the identified parameters by the rank-DEGL method. 2(a) for noise free; 2(b) for low noise; and 2(c) for high noise.

Table 7: Identified parameters by the rank-DEGL method.

Parameter	Actual value	noise free	low noise	high noise	data obtained from [9]
ξ_1	-0.944957	-1.055896 ± 0.1041354	-1.03897 ± 0.1305666	-1.013888 ± 0.1251215	-1.028024 ± 0.132893
ξ_2	0.00301801	0.0031815 ± 3.54E-04	0.0028223 ± 3.83E-04	0.0027366 ± 3.54E-04	0.0032816 ± 4.10E-04
ξ_3	7.401E-05	6.20E-05 ± 1.56E-05	3.78E-05 ± 5.83E-06	3.73E-05 ± 4.63E-06	9.31E-05 ± 1.28E-05
ξ_4	-1.88E-04	-1.88E-04 ± 1.47E-10	-1.82E-04 ± 6.42E-09	-1.75E-04 ± 1.16E-08	-1.33E-04 ± 3.28E-09
λ	23	23.000272 ± 7.49E-04	23.999999 ± 3.76E-06	23.999999 ± 5.35E-06	13.176884 ± 0.0011191
R_c (Ω)	0.0001	1.00E-04 ± 3.33E-07	1.00E-04 ± 6.77E-10	1.00E-04 ± 3.43E-10	8.00E-04 ± 2.02E-09
B (V)	0.02914489	0.0291449 ± 6.93E-08	0.0342694 ± 1.49E-06	0.038843 ± 2.89E-06	0.0145172 ± 2.38E-06
SSE		5.06E-12 ± 1.81E-11	0.2960783 ± 4.20E-06	1.1910132 ± 8.05E-06	0.1931168 ± 4.14E-07

Figure 1. The comparisons between the simulated data and the model curve obtained from the identified parameters by the rank-DEGL method are given in Figure 2. Note that in Tables 4, 5, and 6, "+" means that the ranking-based DE is significant better than its corresponding non-ranking-based DE in terms of the SSE value according to the Wilcoxon signed-rank test at $\alpha = 0.05$. The Wilcoxon's test [41] is a non-parametric statistical test to evaluate the differences between two algorithms.

According to the results shown in Tables 4, 5, and 6, it is clear that our proposed ranking-based DEs consistently provide better results than their corresponding non-ranking-based DEs with respect to the SSE and NFEs values in all cases. It means that the ranking-based mutation operator is able to enhance the performance of the DE variants. The ranking-based DEs not only obtain more accurate solutions, but also they require less NEFs to reach the target fitness. Considering the standard deviation of SSE and NFEs, we see that all ranking-based DEs obtain smaller standard deviation values than their corresponding non-ranking-based DEs, which mean that the ranking-based mutation operator is able to enhance the robustness of the original DE methods. In addition, the *AR* values and the convergence graphs shown in Figure 1 indicate that the ranking-based DEs converge faster compared with their corresponding non-ranking-based DEs. For example, in Table 4, rank-jDE is on average 47% faster than jDE, since $AR = 1.47$. Also, rank-SaDE, rank-JADE, rank-CoDE, and rank-DEGL converge 51%, 10%, 57%, and 8% faster than SaDE, JADE, CoDE, and DEGL, respectively. In terms of the success rate, from Tables 4, 5, and 6, we can see that the ranking-based DEs get higher, or equal to, SR value compared with their corresponding non-ranking-based DEs in all cases. Considering the CPU time consumed by each DE method, we can observe that the ranking-based DEs only require a bit higher time than their corresponding non-ranking-based DEs.

From Tables 4, 5, and 6, it is clear to see that rank-DEGL obtains the best results among all 10 DE variants in terms of the SSE and NFEs criteria. Therefore, in order to verify the performance of parameter identification of rank-DEGL, the identified parameters in the simulated data with noise free, low noise, and high noise are reported in Table 7. All results are statistically calculated over 100 times. The optimal parameters are returned to the mathematical model, and the *V-I* characteristics are plotted in Figure 2. As shown in the figures, the *V-I* curves obtained by rank-DEGL are highly coincide with the simulated data even in the presence of noise. Moreover, in the noise free case, Table 7 shows that the mean values of the identified parameters by rank-DEGL are very close to their corresponding actual values in 5 out of 7 parameters (*i.e.*, ξ_2, ξ_4, λ, R_c, and B). Only in two parameters (ξ_1 and ξ_3), their mean values are

Table 8: Numerical results on the data obtained from [9]. The target fitness $f(\mathbf{x}^*)$ is 0.19311665.

Algorithm	SSE		NFEs	time in seconds	*SR*	*AR*
jDE	0.2192032 ± 0.0175238		9205.00 ± 1222.80	**0.102** ± 0.008	0.20	1.46
rank-jDE	**0.1942431 ± 0.0012399**	+	**6292.00 ± 1016.50**	0.109 ± 0.004	1.00	
SaDE	0.1931228 ± 3.40E-05		4220.00 ± 768.18	**0.107** ± 0.006	1.00	1.57
rank-SaDE	**0.1931167 ± 5.27E-08**	+	**2684.50 ± 380.97**	0.115 ± 0.008	1.00	
JADE	0.195819 ± 0.0024981		7624.74 ± **1057.42**	**0.107** ± 0.006	0.97	1.13
rank-JADE	**0.1946224 ± 0.0012402**	+	**6742.00** ± 1201.78	0.114 ± 0.008	1.00	
CoDE	0.1933435 ± 1.85E-04		5293.20 ± 778.79	**0.101** ± 0.008	1.00	1.63
rank-CoDE	**0.1931182 ± 1.23E-06**	+	**3245.70 ± 464.20**	0.107 ± 0.006	1.00	
DEGL	0.1931171 ± 5.92E-07		1645.70 ± 303.98	**0.111** ± 0.005	1.00	1.16
rank-DEGL	**0.193117 ± 4.14E-07**	+	**1423.80 ± 157.38**	0.117 ± 0.008	1.00	

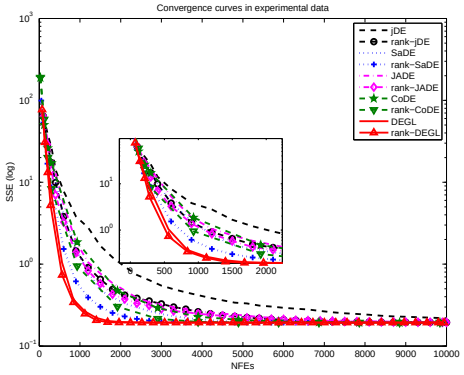


Figure 3: Convergence graphs of all DE variants on the data obtained from [9].

not close to their corresponding actual values. The reason is that there are wider ranges for these two parameters when the optimized SSE value obtained [13]. Also, in this case, the mean and standard deviation values of SSE obtained by rank-DEGL are respectively $5.06E-12$ and $1.81E-11$, which indicate that rank-DEGL is able to get the near-optimal SSE value over all 100 runs.

5.3. Parameter identification with the data obtained from [9]

As shown in Section 5.2, our proposed ranking-based DE got very promising results for parameter identification of PEMFC model with simulated data. In this section, the aforementioned DE variants are further used to identify the parameters of PEMFC model with the *V-I* data obtained from the PEMFC literature. The data originates from the graphical diagrams in [9]. Originally, in [9] there are four data sets. In this work, we only choose one out of four data set. In this data set, the PEMFC stack parameter values and operating conditions are shown in Table 1. The 10 DE variants are applied to optimize the seven parameters of PEMFC model with parameter ranges shown in Table 2. Each algorithm is performed over 100 independent runs. The results are shown in Table 8, and the identified parameters by rank-DEGL is reported in Table 7. The convergence graphs and the *V-I* characteristics of rank-DEGL are respectively plotted in Figures 3 and 4.

From Table 8, we can see that in this case our proposed ranking-based DEs also consistently obtain better performance than their corresponding non-ranking-based DEs. They can provide more accurate solutions and converge faster. Moreover, as shown in Figure 4, the voltage-current characteristics obtained by the rank-DEGL method are in very good agreement with the data obtained from [9]. With respect to the *V-I* data obtained from the literature, even without the prior knowledge, the rank-DEGL method is still practicable for parameter identification of PEMFC models.

5.4. Comparison with other evolutionary algorithms

In the previous subsections, the performance of ranking-based DE variants has been verified to identify the parameters of PEMFC model, and rank-DEGL obtains the overall best results. In order to make our technique be more

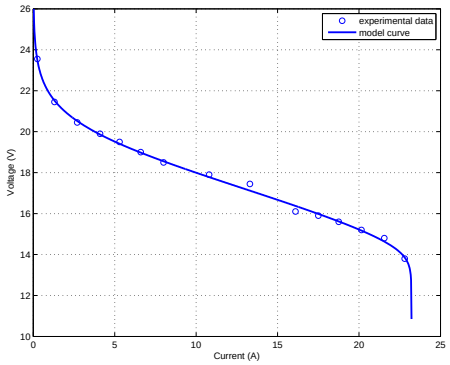


Figure 4: Comparisons between the data obtained from [9] and the model curve obtained from the identified parameters by the rank-DEGL method.

Table 9: Comparison on the SSE values among different evolutionary algorithms.

Item	rcGA	FEP	ABC	CLPSO	rank-DEGL
noise-free	1.1885488 ± 0.910582‡	0.8799576 ± 0.6026365‡	0.0361158 ± 0.0358517‡	**0.0115714 ± 0.0166569**‡	**5.06E-12 ± 1.81E-11**
low-noise	1.2601279 ± 0.7159975‡	1.1679728 ± 0.5156267‡	0.3442355 ± 0.0512642‡	**0.3172881 ± 0.0234083**‡	**0.296078 ± 4.20E-06**
high-noise	2.101808 ± 0.7200706‡	2.1187793 ± 0.5195852‡	1.2323664 ± 0.0381407‡	**1.2157016 ± 0.0246971**‡	**1.191013 ± 8.05E-06**
data obtained from [9]	1.9418167 ± 1.9568087‡	1.0820150 ± 0.5892504‡	0.2606866 ± 0.0569356‡	**0.2051803 ± 0.0213906**‡	**0.193117 ± 4.14E-07**

‡ indicates that rank-DEGL is significant better than its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

Table 10: Compared the SSE value of rank-DEGL with the reported results for the data obtained from [9].

Algorithm	ξ_1	ξ_2	ξ_3	ξ_4	λ	R_c (Ω)	B (V)	SSE
rank-DEGL	-1.0192	3.3186E-03	9.7999E-05	-1.3285E-04	13.1772	8.0000E-04	0.0145	**0.1931**
HGA [9]	-0.9450	3.0180E-03	7.4010E-05	-1.8800E-04	23.0000	1.0000E-04	0.0291	4.8469
SGA [9]	-0.9473	3.0641E-03	7.7134E-05	-1.9390E-04	19.7650	2.7197E-04	0.0240	5.6530
AIS [22]	-0.9518	3.0823E-03	7.7430E-05	-1.8800E-04	22.9121	1.0179E-04	0.0330	2.6895
MPSO [18]	-0.9480	3.0857E-03	7.7990E-05	-1.8800E-04	20.7624	2.8666E-04	0.0297	3.3881
BIPOA [23]	-0.8016	2.6673E-03	8.1288E-05	-1.2713E-04	13.5158	8.0000E-04	0.0324	**1.9350**
ARNA-GA [14]	-0.9470	3.0586E-03	7.6059E-05	-1.8800E-04	23.0000	1.1026E-04	0.0329	2.9518

evident, the performance of rank-DEGL is further compared with other evolutionary algorithms (EAs). Four representative EAs are selected: i) real-coded genetic algorithm (rcGA) [42], fast evolutionary programming (FEP) [43], artificial bee colony (ABC) [44], and comprehensive learning PSO (CLPSO) [45]. The parameter settings of these four algorithms are set as recommended in their original literature. The SSE values of different algorithms are reported in Table 9. All results are averaged over 100 runs. The overall best and the second best results within all compared EAs are highlighted in **grey boldface** and **boldface**, respectively. In addition, the Wilcoxon's test is also adopted to test the significance between rank-DEGL and other EAs. "‡" indicates that rank-DEGL is significant better than its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

From Table 9, it is clear to observe that rank-DEGL gets the overall best results compared with other four EAs in all cases. Moreover, the SSE values of rank-DEGL are more accurate and significantly better than other EAs. Additionally, rank-DEGL provides the smallest standard deviation values of the SSE values over 100 runs in the four cases, which means that rank-DEGL is the most robust one among the five EAs.

5.5. Compared with reported results

In Section 5.3, the *V-I* data is obtained from [9]. In the PEMFC literature, there are other studies that used the data reported in [9]. Therefore, in order to further indicate the superior performance of rank-DEGL, its identified parameters and SSE value are compared with the reported results of HGA [9], SGA [9], AIS [22], MPSO [18], BIPOA [23], and ARNA-GA [14]. Note that since the data used in this work are digitized from the *V-I* curve in [9], there may be some deviations between the data used in the above literature. To make a fair comparison, the parameters

reported in the above literature are fed back to the PEMFC model with the *V*-*I* data used in this work to calculate the SSE values. The results are shown in Table 10. The best and the second best SSE values are highlighted in **grey boldface** and **boldface**, respectively. From the results, it is clearly to observe that the proposed rank-DEGL method is able to obtain the best SSE value, and the second one is the BIPOA. However, the SSE value of rank-DEGL is one order of magnitude less than that of BIPOA. Therefore, we can conclude that rank-DEGL is also very competitive compared with the reported results of the methods published in the PEMFC literature.

6. Conclusions

In this paper, we present the ranking-based mutation operator for the DE algorithm. Our proposed ranking-based mutation is very simple and generic, and it does not significantly increase complexity of the original DE algorithm. The ranking-based mutation is incorporated into five representative DE variants to solve the parameter identification of PEMFC model. The main contributions of this work are the proposed ranking-based DE variants and their application to the parameter identification problems of PEMFC models. Experiments have been conducted on both simulated data and the data obtained from the literature to verify the performance of our approach. According to the numerical results, we can conclude that

- Our proposed ranking-based mutation operator accelerates the process of parameter identification of PEMFC model, and hence, it can reduce the computational efforts to achieve an optimal design.
- The ranking-based mutation operator can consistently enhance the performance of most of DE variants when solving the parameter identification of PEMFC model in terms of the solution quality, the convergence rate, and the success rate. More specifically, ranking-based DEs can obtain smaller SSE values, smaller standard deviation values of SSE, less NFEs, higher success rate, and faster convergence speed when comparing with the corresponding non-ranking-based DEs.
- In overall, the rank-DEGL method obtains the best results among all compared DE variants. In addition, it obtains significantly better results compared with other EAs.
- The voltage-current characteristics obtained by the rank-DEGL method are in very good agreement with both the simulated data and the data originated from the literature.

Due to the superior performance obtained by the rank-DEGL method, it can be an efficient alternative to other complex parameter identification problems of fuel cell models. Therefore, we recommend using rank-DEGL for other complex optimization problems in the field energy. In our future work, we will try to verify this expectation.

References

[1] J. Larminie, A. Dicks, Fuel Cell Systems Explained (Second Edition), Wiley, 2003.

[2] M. J. Khan, M. T. Iqbal, Modelling and analysis of electro-chemical, thermal, and reactant flow dynamics for a PEM fuel cell system, Fuel Cells 5 (4) (2005) 463–475.

[3] A. Askarzadeh, A. Rezazadeh, An innovative global harmony search algorithm for parameter identification of a PEM fuel cell model, IEEE Transactions on Industrial Electronics 59 (9) (2012) 3473 – 3480.

[4] J. Carton, A. Olabi, Wind/hydrogen hybrid systems: Opportunity for Irelands wind resource to provide consistent sustainable energy supply, Energy 35 (12) (2010) 4536 – 4544.

[5] J. C. Amphlett, R. M. Baumert, R. F. Mann, B. A. Peppley, P. R. Roberge, T. J. Harris, Performance modeling of the ballard mark IV solid polymer electrolyte fuel cell, Journal of The Electrochemical Society 142 (1) (1995) 1 – 15.

[6] R. F. Mann, J. C. Amphlett, M. A. Hooper, H. M. Jensen, B. A. Peppley, P. R. Roberge, Development and application of a generalised steady-state electrochemical model for a PEM fuel cell, Journal of Power Sources 86 (1 - 2) (2000) 173 – 180.

[7] J. Corrêa, F. Farret, L. Canha, M. Simoes, An electrochemical-based fuel-cell model suitable for electrical engineering automation approach, IEEE Transactions on Industrial Electronics 51 (5) (2004) 1103 – 1112.

[8] C. Siegel, Review of computational heat and mass transfer modeling in polymer-electrolyte-membrane (PEM) fuel cells, Energy 33 (9) (2008) 1331 – 1352.

[9] Z.-J. Mo, X.-J. Zhu, L.-Y. Wei, G.-Y. Cao, Parameter optimization for a PEMFC model with a hybrid genetic algorithm, International Journal of Energy Research 30 (8) (2006) 585–597.

[10] J. Thepkaew, A. Therdthianwong, S. Therdthianwong, Key parameters of active layers affecting proton exchange membrane (PEM) fuel cell performance, Energy 33 (12) (2008) 1794 – 1800.

[11] J. Carton, A. Olabi, Design of experiment study of the parameters that affect performance of three flow plate configurations of a proton exchange membrane fuel cell, Energy 35 (7) (2010) 2796 – 2806.
[12] J. Carton, V. Lawlor, A. Olabi, C. Hochenauer, G. Zauner, Water droplet accumulation and motion in PEM (proton exchange membrane) fuel cell mini-channels, Energy 39 (1) (2012) 63 – 73.
[13] M. Ohenoja, K. Leiviskä, Validation of genetic algorithm results in a fuel cell model, International Journal of Hydrogen Energy 35 (22) (2010) 12618 – 12625.
[14] L. Zhang, N. Wang, An adaptive RNA genetic algorithm for modeling of proton exchange membrane fuel cells, International Journal of Hydrogen Energy 38 (1) (2013) 219 – 228.
[15] M. Outeiro, R. Chibante, A. Carvalho, A. de Almeida, A parameter optimized model of a proton exchange membrane fuel cell including temperature effects, Journal of Power Sources 185 (2) (2008) 952 – 960.
[16] M. Outeiro, R. Chibante, A. Carvalho, A. de Almeida, A new parameter extraction method for accurate modeling of PEM fuel cells, International Journal of Energy Research 33 (11) (2009) 978–988.
[17] Q. Li, W. Chen, Y. Wang, S. Liu, J. Jia, Parameter identification for PEM fuel-cell mechanism model based on effective informed adaptive particle swarm optimization, IEEE Transactions on Industrial Electronics 58 (6) (2011) 2410 – 2419.
[18] A. Askarzadeh, A. Rezazadeh, Optimization of PEMFC model parameters with a modified particle swarm optimization, International Journal of Energy Research 35 (14) (2011) 1258–1265.
[19] A. Askarzadeh, A. Rezazadeh, A grouping-based global harmony search algorithm for modeling of proton exchange membrane fuel cell, International Journal of Hydrogen Energy 36 (8) (2011) 5047 – 5053.
[20] M. Karimi, A. Askarzadeh, A. Rezazadeh, Using tournament selection approach to improve harmony search algorithm for modeling of proton exchange membrane fuel cell, Int. J. Electrochem. Sci. 7 (2012) 6426 – 6435.
[21] C. Dai, W. Chen, Z. Cheng, Q. Li, Z. Jiang, J. Jia, Seeker optimization algorithm for global optimization: A case study on optimal modelling of proton exchange membrane fuel cell (PEMFC), International Journal of Electrical Power & Energy Systems 33 (3) (2011) 369 – 376.
[22] A. Askarzadeh, A. Rezazadeh, Artificial immune system-based parameter extraction of proton exchange membrane fuel cell, International Journal of Electrical Power & Energy Systems 33 (4) (2011) 933 – 938.
[23] S. Yang, N. Wang, A novel P systems based optimization algorithm for parameter estimation of proton exchange membrane fuel cell model, International Journal of Hydrogen Energy 37 (10) (2012) 8465 – 8476.
[24] U. K. Chakraborty, T. E. Abbott, S. K. Das, PEM fuel cell modeling using differential evolution, Energy 40 (1) (2012) 387 – 399.
[25] M. Secanell, J. Wishart, P. Dobson, Computational design and optimization of fuel cells and fuel cell systems: A review, Journal of Power Sources 196 (8) (2011) 3690 – 3704.
[26] R. Storn, K. Price, Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces, J. of Global Optim. 11 (4) (1997) 341–359.
[27] K. Price, R. Storn, J. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer-Verlag, Berlin, 2005.
[28] S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Trans. on Evol. Comput. 15 (1) (2011) 4–31.
[29] K. Vaisakh, L. Srinivas, A genetic evolving ant direction DE for OPF with non-smooth cost functions and statistical analysis, Energy 35 (8) (2010) 3155 – 3171.
[30] K. Fong, C. Lee, C. Chow, S. Yuen, Simulation-optimization of solar-thermal refrigeration systems for office use in subtropical Hong Kong, Energy 36 (11) (2011) 6298 – 6307.
[31] Y. Wang, J. Zhou, L. Mo, R. Zhang, Y. Zhang, Short-term hydrothermal generation scheduling using differential real-coded quantum-inspired evolutionary algorithm, Energy 44 (1) (2012) 657 – 671.
[32] H. Zhang, J. Zhou, N. Fang, R. Zhang, Y. Zhang, Daily hydrothermal scheduling with economic emission using simulated annealing technique based multi-objective cultural differential evolution approach, Energy 50 (1) (2013) 24 – 37.
[33] J. Brest, S. Greiner, B. Bošković, M. Mernik, V. Žumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Trans. on Evol. Comput. 10 (6) (2006) 646–657.
[34] A. K. Qin, V. L. Huang, P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. on Evol. Comput. 13 (2) (2009) 398–417.
[35] J. Zhang, A. C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Trans. on Evol. Comput. 13 (5) (2009) 945–958.
[36] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. on Evol. Comput. 15 (1) (2011) 55–66.
[37] S. Das, A. Abraham, U. K. Chakraborty, A. Konar, Differential evolution using a neighborhood-based mutation operator, IEEE Trans. on Evol. Comput. 13 (3) (2009) 526–553.
[38] J. Corrêa, F. Farret, V. Popov, M. Simoes, Sensitivity analysis of the modeling parameters used in simulation of proton exchange membrane fuel cells, IEEE Transactions on Energy Conversion 20 (1) (2005) 211 – 218.
[39] W. Gong, Z. Cai, Differential evolution with ranking-based mutation operators, IEEE Transactions on Cybernetics (2013) 1 – 16In press.
[40] S. Rahnamayan, H. Tizhoosh, M. Salama, Opposition-based differential evolution, IEEE Trans. on Evol. Comput. 12 (1) (2008) 64–79.
[41] C. Shaw, K. Williams, R. Assassa, Patients' views of a new nurse-led continence service, Journal of Clinical Nursing 9 (4) (2003) 574–584.
[42] F. Herrera, M. Lozano, J. Verdegay, Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis, Artificial Intelligence Review 12 (4) (1998) 265–319.
[43] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. on Evol. Comput. 3 (2) (1999) 82–102.
[44] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, J. of Global Optimization 39 (3) (2007) 459–471.
[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. on Evol. Comput. 10 (3) (2006) 281–295.