# Parameter optimization of PEMFC model with improved multi-strategy adaptive differential evolution

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

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

**Abstract**

Parameter optimization of proton exchange membrane fuel cell (PEMFC) model has received considerable attention recently. In order to estimate the unknown parameters of PEMFC model faster and obtain more accurate solutions, in this paper, an improved multi-strategy adaptive differential evolution (DE) is presented for the parameter optimization problems of PEMFC model. The approach is referred to as rank-MADE, for short. In rank-MADE, the multiple mutation strategies of DE are adaptively selected to avoid choosing a suitable strategy for a specific problem by trial-and-error method. Furthermore, the ranking-based vector selection technique is employed in different mutation strategies to accelerate the process of parameter optimization of PEMFC model. In order to verify the performance of rank-MADE, it is applied to estimate the parameters of the Ballard Mark V FC, the SR-12 Modular PEM Generator, the BCS 500-W stack, the Temasek FC, and the WNS-FC model. In addition, rank-MADE is compared with other advanced DE variants and other evolutionary algorithms (EAs). Experimental results show that rank-MADE is able to provide higher quality of solutions, faster convergence speed, and higher success rate compared with other DE variants. Additionally, the *V*-*I* characteristics obtained by rank-MADE agree well with the experimental data in all cases. Therefore, rank-MADE can be an effective alternative in the field of other complex parameter optimization problems of fuel cell models.

*Key words:* Proton exchange membrane fuel cell (PEMFC), parameter optimization, differential evolution, strategy adaptation, ranking-based vector selection.

## 1. Introduction

Due to the urgent demands of clean energy solutions of our world, recently, researchers pay more attention to develop new technologies in the field of power generation systems. As one of the most popular types of new technologies, fuel cells (FCs) are considered to be a promising candidate in the twenty-first century, because of their low aggression to the environment, low noise, good dynamic response, and high efficiency. There are several different kinds of fuel cells based on the nature of used electrolyte [1]. However, among various kinds of fuel cells, the proton exchange membrane fuel cells (PEMFCs) are widely studied and promising area for different applications [2].

For an efficient design of PEMFC-based systems, the PEMFC models are required. There have been many studies on modeling and simulation of the PEMFC [3, 4, 5, 6, 7]. Mo *et al.* [8] classified different PEMFC models into two approaches: i) mechanistic models, which aim at simulating the heat, mass transfer and electrochemical phenomena present in fuel cells; and ii) models on empirical or semi-empirical equations, which are applied to predict the effect of different input parameters on the voltage-current (*V*-*I*) characteristics of the fuel cell, without examining in depth the physical and electrochemical phenomena involved in the operation. In this paper, the model based on the second approach is used. However, no matter what type of models, the model parameters need to be optimized in order to improve the accuracy of the models and make the models indicate the actual PEMFC performance better [8].

*Corresponding author. Tel: +86-27-67883716.
*Email addresses:* wenyingong@yahoo.com; wygong@cug.edu.cn (Wenyin Gong), zhcai@cug.edu.cn (Zhihua Cai)

In order to improve the design and performance of PEMFC systems, parameter optimization of PEMFC model has attracted increasing interest in recent years. However, since the PEMFC is a complex nonlinear, multi-variable, and strongly coupled system [9], the optimization of parameters of PEMFC model is difficult to be handled by the traditional methods. Recently, the use of heuristic optimization techniques for parameter optimization of PEMFC model has received more attention [10], such as genetic algorithms [8], simulated annealing [11], particle swarm optimization [12], artificial bee swarm [13], harmony search [14], seeker optimization algorithm [15], artificial immune system [16], P systems based optimization algorithm [17], bird mating optimizer [18], and so on. Most recently, differential evolution is also used to solve the parameter optimization of PEMFC model [19]. However, in order to efficiently and fast solve the parameter optimization problems in PEMFC model, it is necessary to investigate more efficient optimization techniques to reduce the necessary computational efforts to achieve an optimal design [20].

Differential evolution (DE) is a simple yet efficient global numerical optimization algorithm [21, 22]. Owing to its simple structure, ease of use, speed, and robustness, it has be successfully applied in diverse fields, such as optimization of over-current relays [23], optimal power flow [24], modeling of oxygen mass transfer [25], design of cascade fuzzy controller [26], motion estimation [27], parameter extraction of solar cell models [28], etc. For more details, interested readers can refer to two good surveys of DE in [29] and [30], and the references therein.

Inspired by the various successful applications of DE, in this paper, the DE algorithm is employed for the parameter optimization problems of PEMFC model. In order to reduce the computational efforts and obtain higher quality of solutions, a ranking-based multi-strategy adaptive DE, referred to as rank-MADE, is proposed. More specifically, in rank-MADE, the multiple mutation strategies of DE are adaptively selected to avoid choosing a suitable strategy for a specific problem by trial-and-error method. The *Probability Matching* technique [31] is used to assign the selection probabilities of different strategies. In addition, to accelerate the process of parameter optimization of PEMFC model the ranking-based vector selection technique presented in [32] is employed in different mutation strategies. The proposed rank-MADE is used to solve the parameter optimization problems of the Ballard Mark V FC [5], the SR-12 Modular PEM Generator [5], the BCS 500-W stack [5], the Temasek FC [33], and the WNS-FC model [34]. The performance of rank-MADE is compared with other highly-competitive advanced DE variants (*i.e.*, SaDE [35], JADE [36], CoDE [37], and DEGL [38]) and other evolutionary algorithms (*i.e.*, rcGA [39], FEP [40], ABC [41], and CLPSO [42]). Experimental results show that rank-MADE is able to provide higher quality of solutions, faster convergence speed, and higher success rate compared with other DE variants. Additionally, the good agreement between the experimental data and the output data of the model obtained by rank-MADE can be observed in all cases.

The main contributions of this work are as follows: i) Inspired by the success of the ensemble learning in machine learning [43], ensemble of different advanced improvements of evolutionary algorithms (EAs) may be also able to develop more enhanced optimization techniques. Based on this consideration and in order to provide an effective alternative for optimizing the parameters of PEMFC model faster and more accurately, the rank-MADE method is proposed, where our previous proposed multi-strategy adaptation technique [44] and the ranking-based vector selection technique [32] are synergized. ii) rank-MADE is used to solve the parameter identification problems of different PEMFC models, and its performance is compared with other highly-competitive DE variants and other EAs. iii) To make the comparisons among different algorithms statistically meaningful, the Wilcoxon's test and the Friedman test are employed.

The rest of this paper is organized as follows. Section 2 briefly introduces the mathematical formulation of PEMFC model used in this work. In addition, the objective function to be optimized, the original DE algorithm, and the parameter optimization of PEMFC model with artificial intelligence (AI) methods are also described in this section. In Section 3, the proposed rank-MADE method is presented in detail. The experimental results and analysis are given in Section 4. Finally, Section 5 draws the conclusions of this work.

## 2. Preliminaries

In this section, the mathematical formulation of PEMFC model used in this work is firstly introduced, followed by the description of the objective function. Additionally, the original DE algorithm is also briefly introduced herein.

*2.1. Mathematical formulation of PEMFC model*

In this work, the mathematical model of PEMFC stacks presented in [5, 45] is adopted. For $n$ cells connected in series to form a stack, the terminal voltage of the stack can be calculated by [45],

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

where $V_{\mathrm{FC}}$ is the output voltage of a single cell, which can be formulated as [5]

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

$E_{\mathrm{Nernst}}$ is the thermodynamic potential defined by [4]

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

where $T$ is the cell temperature (K), $P_{\mathrm{H_2}}$ and $P_{\mathrm{O_2}}$ are the hydrogen and oxygen partial pressures (atm), respectively.

According to [4], the activation overpotential $V_{\mathrm{act}}$, including anode and cathode, can be expressed by the following formula

$$V_{\mathrm{act}} = -\left[\xi_1 + \xi_2 \cdot T + \xi_3 \cdot T \cdot \ln\left(C_{\mathrm{O_2}}\right) + \xi_4 \cdot T \cdot \ln\left(i_{\mathrm{cell}}\right)\right] \tag{4}$$

where $\xi_1, \xi_2, \xi_3, \xi_4$ are the parametric coefficients for each cell model, $i_{\mathrm{cell}}$ is the cell current (A), and $C_{\mathrm{O_2}}$ (mol/cm$^3$) is the concentration of oxygen in the catalytic interface of the cathode, given by [5, 8]

$$C_{\mathrm{O_2}} = \frac{P_{\mathrm{O_2}}}{5.08 \times 10^6 \cdot \exp\left(-498/T\right)} \tag{5}$$

The ohmic voltage drop $V_{\mathrm{ohm}}$ can be determined by the following expression [4]

$$V_{\mathrm{ohm}} = i_{\mathrm{cell}} \cdot (R_M + R_C) \tag{6}$$

where $R_M$ is the equivalent membrane resistance to proton conduction, and $R_C$ is the equivalent contact resistance to electron conduction ($\Omega$). $R_M$ is defined by [8]

$$R_M = \frac{\rho_M \cdot \ell}{A} \tag{7}$$

$$\rho_M = \frac{181.6 \cdot \left[1 + 0.03 \cdot \left(\frac{i_{\mathrm{cell}}}{A}\right) + 0.062 \cdot \left(\frac{T}{303}\right)^2 \cdot \left(\frac{i_{\mathrm{cell}}}{A}\right)^{2.5}\right]}{\left[\lambda - 0.634 - 3 \cdot \left(\frac{i_{\mathrm{cell}}}{A}\right)\right] \cdot \exp\left[4.18 \cdot \left(\frac{T-303}{T}\right)\right]} \tag{8}$$

where $A$ is the effective electrode area (cm$^2$), $\rho_M$ is the membrane specific resistivity for the flow of hydrated protons ($\Omega \cdot$ cm ), and $\ell$ is the thickness of the membrane (cm), which serves as the electrolyte of the cell. The parameter $\lambda$ is an adjustable parameter with a possible range of [10, 24].

The concentration overpotential $V_{\mathrm{con}}$ caused by the change in the concentration of the reactants at the surface of the electrodes as the fuel is calculated by [5]

$$V_{\mathrm{con}} = -B \cdot \ln\left(1 - \frac{J}{J_{\max}}\right) \tag{9}$$

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$^2$) including the fixed current density $J_n$ [5], and $J_{\max}$ is the maximum value of $J$.

### 2.2. Objective function

In the above model (Equations (1) - (9)), there are 11 unknown parameters, *i.e.*, $\xi_1$, $\xi_2$, $\xi_3$, $\xi_4$, $\lambda$, $R_c$, $B$, $J_n$, $J_{\max}$, $\ell$, and $A$. In this work, the first 10 parameters will be optimized, while the effective electrode area $A$ is left to be constant during the optimization process. Thus, there are totally 10 parameters to be optimized, which can be denoted as a real-valued vector:

$$\mathbf{x} = \{\xi_1, \xi_2, \xi_3, \xi_4, \lambda, R_C, B, J_n, J_{\max}, \ell\}$$

In order to identify the optimal values of the 10 unknown parameters by the optimization techniques, it needs to define a objective function to be optimized. In this work, the mean square error (MSE) between the output voltage of the *actual* PEMFC stack and the model output voltage are used as the objective function [9]:

$$\min \quad f(\mathbf{x}) = \frac{1}{N} \cdot \sum_{k=1}^{N} (V_{\mathrm{sm},k} - V_{\mathrm{s},k})^2 \tag{10}$$

where $V_{\mathrm{sm}}$ is the output voltage of the actual PEMFC stack, $V_{\mathrm{s}}$ is the model output voltage, and $N$ is the number of the experimental data point.

### 2.3. Differential evolution

The DE algorithm [21] is a simple evolutionary algorithm for global numerical optimization. It creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has better fitness value. The pseudo-code of the original DE algorithm is shown in Algorithm 1. Where $D$ is the number of decision variables; $\mu$ is the population size; $F$ is the mutation scaling factor; $Cr$ is the crossover rate; $x_{i,j}$ is the $j$-th variable of the solution $\mathbf{x}_i$; $\mathbf{u}_i$ is the offspring. rndint$(1, D)$ is a uniformly distributed random integer number between 1 and $D$. rndreal$_j[0, 1)$ is a uniformly distributed random real number in $[0, 1)$, generated anew for each value of $j$. In [22], the vectors $\mathbf{x}_i$, $\mathbf{x}_{r_1}$, $\mathbf{v}_i$, and $\mathbf{u}_i$ are named as *target vector*, *base vector*, *mutant vector*, and *trial vector*, respectively.

**Algorithm 1:** The original DE algorithm

```
Input: Cr, F, and μ
Output: The best solution x_best
Generate the initial population randomly;
Evaluate the fitness for each individual;
Set the generation counter t = 1;
while not terminated do
    for i = 1 to μ do                                          /* Generate offspring */
        Select uniform randomly r1 ≠ r2 ≠ r3 ≠ i;
        j_rand = rndint(1, D);
        for j = 1 to D do
            if rndreal_j[0, 1) < Cr or j == j_rand then
                u_{i,j} = v_{i,j} = x_{r1,j} + F · (x_{r2,j} − x_{r3,j});
            else
                u_{i,j} = x_{i,j};
    for i = 1 to μ do                                          /* Survival selection */
        Evaluate the offspring u_i;
        if u_i is better than its parent x_i then
            x_i = u_i;
    t = t + 1;
```

### 2.4. Related work on AI-based PEMFC parameter optimization

In Section 1, some AI-based optimization techniques for parameter optimization of PEMFC model are outlined. In this section, they will be introduced in brief.

In [8], Mo *et al.* presented a niche hybrid genetic algorithm (HGA) for parameter optimization of PEMFC model, where the niche techniques and Nelder-Mead's simplex method are merged into GA. The results of HGA is only compared with simple GA (SAG) in [8]. Outeiro *et al.* [11, 46] applied the simulated annealing (SA) as optimization

technique to extract the parameters of PEMFC model. In [12], a PSO-based parameter optimization technique of PEMFC model was presented according to the *V-I* characteristics. The results indicated the effectiveness of PSO when identifying the parameters of PEMFC model even in noise environments [12]. Ohenoja and Leiviskä [47] conducted comprehensively experiments to indicate how the parameter range, the validation strategy, and the selected algorithm influence on the performance of GAs in parameter optimization of PEMFC model. According to the results, the authors claimed that the parameter range needs a careful consideration before optimization trials [47]. In [48], Li *et al.* firstly presented an effective informed adaptive PSO (EIA-PSO) to balance the global and local search. Then, EIA-PSO was employed for the PEMFC model parameter optimization. Askarzadeh and Rezazadeh [49] proposed a modified PSO (MPSO) to optimize the parameters of PEMFC model, where a modified method is presented for the PSO's inertia weight in MPSO. In [13], an artificial bee swarm optimization algorithm is proposed for optimizing the parameters of a steady-state PEMFC stack model suitable for electrical engineering applications. In [14, 50, 9], the grouping-based global harmony search (HS), tournament selection based HS, and elite-based global HS were respectively presented for the PEMFC model parameter optimization. To optimize the PEMFC model parameters, Dai *et al.* [15] proposed a novel seeker optimization algorithm (SOA), which is based on the concept of simulating human searching behaviors. As a promising optimization technique, artificial immune system (AIS) has obtained increasing attention recently. In [16], the AIS-based parameter extraction of PEMFC model was present, and its results is compared with those of GAs and PSO. Yang and Wang [17] proposed a novel bio-inspired P systems-based optimization algorithm (BIPOA) to solve the PEMFC model parameter optimization problems. In [19], a DE variant, *i.e.*, DEGL [38], was employed for the parameter optimization problem of PEMFC stack. Inspired by the mechanism of biological RNA, Zhang and Wang [51] presented an adaptive RNA GA (ARNA-GA) for estimating the PEMFC model parameters. The results of ARNA-GA are indirectly compared with SGA, HGA, and RGA [51]. In [18], Askarzadeh and Rezazadeh proposed a bird mating optimizer (BMO), which is inspired by the intelligent behavior of birds during mating season. BMO is used to optimize the parameters of the Ballard Mark V FC.

According to the literature review, it is noted that the use of DE for parameter optimization of PEMFC model is scarce. Because DE has been successfully applied in diverse domains [30], it may also be useful to optimize the PEMFC model parameters. For this reason, an improved DE variant (rank-MADE) will be developed to efficiently solve the parameter optimization problems of PEMFC model in the next section.

## 3. Our Approach: rank-MADE

In this section, the proposed rank-MADE will be presented in detail as follows, including the motivations of this work, the multi-strategy adaptation, and the ranking-based vector selection.

### *3.1. Motivations*

In the DE literature, there are many advanced DE variants (such as SaDE [35], DEGL [38], JADE [36], MADE [44], CoDE [37], rank-DE [32], etc), where different improvements are proposed to improve the performance of the original DE algorithm. In machine learning, ensemble learning that uses multiple models is very useful to obtain better predictive performance, such as bagging, boosting, etc [43]. Inspired by the success of ensemble learning, ensemble of different improvements of existing advance DE variants may be useful to further enhance DE performance. In addition, when solving the parameter optimization problems of PEMFC model, the fuel cell researchers and engineers require the optimization technique that can fast and precisely optimize the parameters. Based on these considerations, the rank-MADE method is developed, which synergized the multi-strategy adaptation and ranking-based vector selection techniques previously proposed in [44] and [32]. The multi-strategy adaptation technique is able to alleviate the difficulty of the choice of the best DE strategy. Meanwhile, the ranking-based vector selection can accelerate the parameter optimization process of PEMFC model. Most importantly, the two techniques do not increase the complexity of the original DE method significantly.

### *3.2. Multi-strategy adaptation*

In the DE algorithm, there are many mutation strategies [22], however, the choice of the most suitable one for a specific problem is very difficult [35]. Take this aspect into consideration, in this work, the multi-strategy adaptation method presented in [44] is employed to avoid choosing the most suitable strategy by trial-and-error method. This method is briefly described as follows. More details can be found in [44].

**Algorithm 2:** Ranking-based vector selection for "DE/rand/1"

**Input**: The target vector index $i$
**Output**: The selected vector indexes $r_1, r_2, r_3$
**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 $r_3 \in \{1, \mu\}$;

*3.2.1. Strategy pool*

In [44], there are four strategies in the pool. However, it is clear that the "DE/rand/2/bin" got the worst performance in the whole run. Therefore, in this work, only three strategies are selected to form the strategy pool:

- "DE/rand/1/bin":

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{11}$$

- "DE/rand-to-best/2/bin":

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{best} - \mathbf{x}_{r1}) + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) + F \cdot (\mathbf{x}_{r_4} - \mathbf{x}_{r_5}) \tag{12}$$

- "DE/current-to-rand/1":

$$\mathbf{v}_i = \mathbf{x}_i + F \cdot (\mathbf{x}_{r_1} - \mathbf{x}_i) + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{13}$$

*3.2.2. Credit assignment*

Denote $w_a(t)$ as the reward that a strategy $a$ receives after its application at time $t$. In order to assign the credit for each strategy, the averaged normalization award is adopted:

$$w_a(t) = \frac{w'_a(t)}{\max\limits_{a=1,\cdots,K} w'_a(t)} \tag{14}$$

where $w'_a(t)$ is calculated as

$$w'_a(t) = \frac{\sum_{i=1}^{|S_a|} S_a}{|S_a|} \tag{15}$$

and $S_a$ is the set of all relative fitness improvement $\eta_i$ of a strategy $a(a = 1, \cdots, K)$ at generation $t$. Similar to the method proposed in [52], for the minimization problem, $\eta_i$ is calculated by

$$\eta_i = \frac{\delta}{cf_i} \cdot (pf_i - cf_i) \tag{16}$$

where $i = 1, \cdots, \mu$. $\delta$ is the fitness of the best-so-far solution in the population. $pf_i$ and $cf_i$ are the fitness of the target parent and its offspring, respectively. Note that as we consider the minimization problem in this work, $pf_i - cf_i \geq 0$. In case no improvement is achieved, a null reward is assigned. In [44], four different credit assignment methods are presented, and the averaged normalization award is able to provide highly-competitive results through benchmark functions. Based on this consideration, it is selected for the credit assignment in this work.

*3.2.3. Strategy selection*

The probability matching (PM) technique [31] is used to assign the selection probability of each strategy. Suppose we have $K > 1$ strategies ($K = 3$ in this work) in the pool $A = \{a_1, \cdots, a_K\}$ and a probability vector $\mathbf{P}(t) = \{p_1(t), \cdots, p_K(t)\}$ $(\forall t : 0 \leq p_i(t) \leq 1; \sum_{i=1}^{K} p_i(t) = 1)$. In this work, the PM technique is used to adaptively update the probability $p_a(t)$ of each strategy $a$ based on its reward. $q_a(t)$ is the known quality (or empirical estimate) of a strategy $a$, that is updated as follows [53]:

$$q_a(t+1) = q_a(t) + \alpha[w_a(t) - q_a(t)], \tag{17}$$

where $\alpha \in (0, 1]$ is the adaptation rate. The PM method updates the probability $p_a(t)$ as follows [31, 53]:

$$p_a(t+1) = p_{\min} + (1 - K \cdot p_{\min}) \frac{q_a(t+1)}{\sum_{i=1}^{K} q_i(t+1)}. \tag{18}$$

where $p_{\min} \in (0, 1)$ is the minimal probability value of each strategy, used to ensure that no operator gets lost.

### *3.3. Ranking-based vector selection*

In order to reduce the computational efforts and make the parameter optimization process of PEMFC model faster, in this work, the ranking-based vector selection technique [32] is also used in the above-mentioned three strategies.

In the ranking-based vector selection technique, the population is firstly ranked from the best to the worst. Then, the rank $R_i$ of each vector $\mathbf{x}_i$ in the sorted population is assigned as

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

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

$$pv_i = \left( \frac{R_i}{\mu} \right)^2 \tag{20}$$

Finally, the vectors in the mutation strategy are selected according to their selection probabilities as shown in Algorithm 2. Note that in Algorithm 2 only the vector selection for "DE/rand/1" is illustrated, for other mutation operators the vector selection is similar to Algorithm 2.

**Algorithm 3:** The pseudo-code of rank-MADE

```
Input: Cr, F, and μ
Output: The best solution x_best
Generate the initial population randomly;
Evaluate the fitness for each individual;
Set the generation counter t = 1;
Set K = 3, p_min = 0.05, and α = 0.3;
For each strategy a, set q_a(t) = 0 and p_a(t) = 1/K;
while not terminated do
    for i = 1 to μ do                                          /* Generate offspring */
        Select the strategy SI_i based on its selection probability;
        Select uniform randomly r_1 ≠ r_2 ≠ r_3 ≠ r_4 ≠ r_5 ≠ i;
        j_rand = rndint(1, D);
        for j = 1 to D do
            if rndreal_j[0, 1) < Cr or j == j_rand then
                if SI_i == 1 then
                    u_{i,j} is generated by "DE/rand/1" strategy;
                else if SI_i == 2 then
                    u_{i,j} is generated by "DE/rand-to-best/2" strategy;
                else
                    u_{i,j} is generated by "DE/current-to-rand/1" strategy;
            else
                u_{i,j} = x_{i,j};
    for i = 1 to μ do                                          /* Survival selection */
        Evaluate the offspring u_i;
        if u_i is better than its parent x_i then
            Calculate η_i using Equation (16);
            x_i = u_i;
        else
            Set η_i = 0;
        S_{SI_i} ← η_i;
    Calculate the reward w_a(t) for each strategy;
    Update the quality q_a(t) for each strategy;
    Update the probability p_a(t) for each strategy;
    t = t + 1;
```

Table 1: Main features of related approaches.

| Algorithm | Ranking-based mutation | Multi-strategy adaptation | Problems |
|---|---|---|---|
| SaJADE [54] | No | Yes, strategy adaptive mechanism | benchmark functions |
| PM-ASS-DE [44] | No | Yes, probability matching | benchmark functions |
| rank-DE [32] | Yes | No | benchmark functions |
| $R_{cr}$-IJADE [28] | Yes | No | solar cell models |
| rank-MADE | Yes | Yes, probability matching | PEMFC models |

### *3.4. The rank-MADE approach*

Combining the multi-strategy adaptation method with the ranking-based vector selection, in this work, the rank-MADE method is proposed. The pseudo-code of rank-MADE is described in Algorithm 3. The proposed rank-MADE has the two main advantages: i) it releases to seek the most suitable strategy for a specific problem; and ii) it can accelerate the convergence speed, and hence, it reduces the computational efforts when solving complex problems. These two advantages can make this approach be suitable to real-world applications, such as the parameter optimization problems of fuel cell models. In order to verify the expectation, in the following section, rank-MADE is used to solve the parameter optimization problems of PEMFC model.

As mentioned above, rank-MADE synergizes different components of different previous approaches, however, it is different from our previous work presented in [54, 44, 32, 28]. In order to clearly indicate the differences between rank-MADE and our previous work, the main features of these approaches are tabulated in Table 1. From Table 1, it can be observed that the differences between rank-MADE and other approaches are as follows:

- rank-MADE vs SaJADE [54]: In SaJADE [54], the strategy adaptation is controlled by a simple strategy adaptive mechanism; whereas in rank-MADE the probability matching is used to implement the strategy adaptation. In addition, the ranking-based mutation is not used in SaJADE.
- rank-MADE vs PM-ASS-DE [44]: Both approaches use the the probability matching to assign the selection probabilities of different strategy. In this view, rank-MADE is an improved version of PM-ASS-DE. However, in rank-MADE the ranking-based mutation is used to accelerate the convergence rate for parameter optimization of PEMFC models.
- rank-MADE vs rank-DE [32]: Both of them use the ranking-based mutation, while in rank-MADE the multi-strategy adaptation is used to release to seek the most suitable strategy when solving the parameter optimization problems of PEMFC models.
- rank-MADE vs $R_{cr}$-IJADE [28]: There are three main differences between the two approaches: i) rank-MADE is a multi-strategy DE variants, whereas $R_{cr}$-IJADE only has one mutation strategy; ii) in $R_{cr}$-IJADE, the crossover rate repairing technique is proposed, which is not used in rank-MADE; and iii) $R_{cr}$-IJADE is used for parameter extraction of solar cell models, while in this work rank-MADE is optimized the parameters of PEMFC models.

To sum up, although each component of rank-MADE is not new, however, the ensemble of these is a new proposal. In addition, in the application point of view, the application of rank-MADE for the parameter optimization of PEMFC model is interesting, which may be provide an efficient alternative for researchers and engineers in the field of fuel cell.

## 4. Experimental results and analysis

For validation of the proposed rank-MADE, in this work, four cases of PEMFC model are chosen from the literature: i) the Ballard Mark V FC [5], ii) the SR-12 Modular PEM Generator [5], iii) the BCS 500-W stack [5], and iv) the Temasek FC [33]. The parameter values and operation conditions of PEMFC stack are tabulated in Table 2. In this work, there are 10 unknown parameters of PEMFC that need to be optimized. The ranges of these parameters are shown in Table 3, where the ranges of the first 9 parameters are originated from [9].

Table 2: Parameter Values and Operation Conditions of PEMFC Stack.

| | $n$ | $T$ (K) | $A$ (cm$^2$) | $P_{H_2}$ (atm) | $P_{O_2}$ (atm) |
|---|---|---|---|---|---|
| Ballard Mark V FC | 1 | 343 | 50.6 | 1 | 1 |
| SR-12 PEM Generator | 48 | 323 | 62.5 | 1.47628 | 0.2095 |
| BCS 500-W | 32 | 333 | 64 | 1 | 0.2095 |
| Temasek FC | 20 | 323 | 150 | 0.5 | 0.5 |

Table 3: Ranges of the Unknown Parameters to Be Optimized.

| Parameter | $\xi_1$ | $\xi_2$ | $\xi_3$ | $\xi_4$ | $\lambda$ | $R_c$ ($\Omega$) | $B$ (V) | $J_n$ (mA/cm$^2$) | $J_{\max}$ (mA/cm$^2$) | $\ell$ ($\mu$m) |
|---|---|---|---|---|---|---|---|---|---|---|
| Lower bound | -1.1997 | 0.001 | -3.60E-05 | -2.60E-04 | 10 | 0.0001 | 0.0136 | 1 | 500 | 25 |
| Upper bound | -0.8532 | 0.005 | -9.80E-05 | -9.54E-05 | 24 | 0.0008 | 0.5 | 30 | 1500 | 178 |

Table 4: Parameter settings for all DE variants.

| Algorithm | Parameter settings |
|---|---|
| rank-MADE | $\mu = 50, Cr = 0.9, F = 0.5, p_{\min} = 0.05, \alpha = 0.3$ |
| SaDE | $\mu = 50, LP = 50$ [35] |
| JADE | $\mu = 100, p = 0.05, c = 0.1$ [36] |
| CoDE | $\mu = 30$ [37] |
| DEGL | $\mu = 10 \times D, Cr = 0.9, F = 0.8$ [38] |

### *4.1. Parameter Settings for DE Variants*

For all experiments, the parameters of all DE variants are shown in Table 4 unless a change is mentioned. To make a fair comparison, the parameters of SaDE [35], JADE [36], CoDE [37], and DEGL [38] are the same as used in their original corresponding literature. For rank-MADE, the population size is set the same as used in SaDE [35], since both of them are multi-strategy-based DE variants; while for the rest four parameters the same settings presented in [44] are used. All algorithms are coded in standard C++. The maximal number of function evaluations (Max_NFEs) are set to $10,000$. Since DE is the stochastic algorithm, in order to make the comparison meaningful, each problem is optimized over 100 independent runs.

### *4.2. Performance criteria*

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

- **MSE**: It is calculated by Equation (10) to measure the solution quality of a method obtained.
- **NFEs$_{\mathbf{VTR}}$**: Let $\epsilon = f(\mathbf{x}) - f(\mathbf{x}^*)$ be the *error* of a solution $\mathbf{x}$, where $f(\mathbf{x}^*)$ is the target fitness to be reached by a specific problem. The NFEs$_{\mathrm{VTR}}$ is used to record the number of function evaluations in each run for finding a solution satisfying $\epsilon \leq 1e-5$.
- **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 $\epsilon \leq 1e-5$.
- **Convergence graphs**: The graphs show the mean error performance of the total runs.

### *4.3. Comparison with other DE variants*

Since the proposed rank-MADE is an improved DE variant, in order to indicate the superiority of this approach against other DE variants, in this section, the above-mentioned DE variants are first evaluated in the four PEMFC models in terms of different performance criteria.

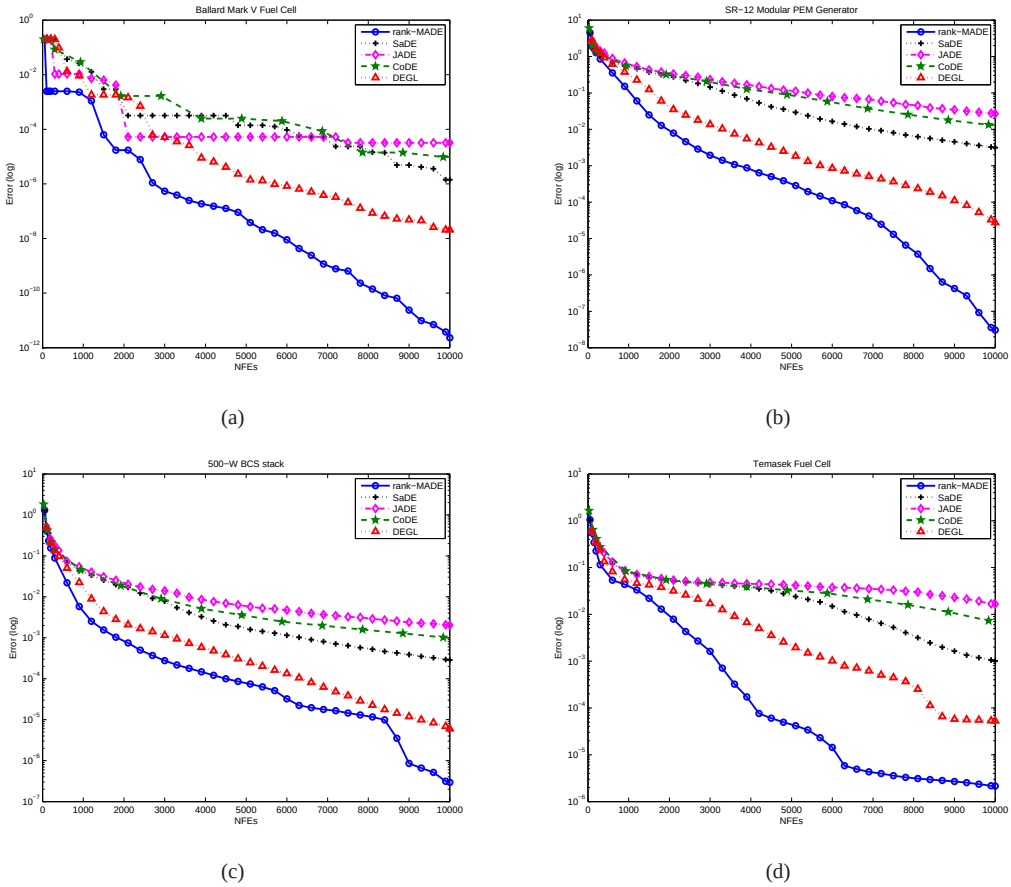



Figure 1: Convergence graphs of all algorithms on 1(a) the Ballard Mark V FC, 1(b) the SR-12 Modular PEM Generator, 1(c) the BCS 500-W stack, and 1(d) the Temasek FC.

Table 5: Comparison on the Identified Parameters and Performance Criteria of Different Algorithms for the Ballard Mark V Fuel Cell. The Target Fitness $f(\mathbf{x}^*)$ Is $4.74765534862167E-05$.

| | | SaDE | JADE | CoDE | DEGL | rank-MADE |
|---|---|---|---|---|---|---|
| Parameter | $\xi_1$ | -1.021249 ± 0.0758694 | -1.020216 ± 0.0771143 | -1.035307 ± 0.0832044 | -1.03601 ± 0.0918698 | -1.032845 ± 0.1239228 |
| | $\xi_2$ | 0.0034094 ± 2.79E-04 | 0.0034316 ± 3.12E-04 | 0.003447 ± 3.16E-04 | 0.0032844 ± 6.12E-04 | 0.0033514 ± 4.44E-04 |
| | $\xi_3$ | 6.80E-05 ± 1.44E-05 | 6.89E-05 ± 1.47E-05 | 6.72E-05 ± 1.58E-05 | 6.62E-05 ± 1.67E-05 | 6.46E-05 ± 2.25E-05 |
| | $\xi_4$ | -1.37E-04 ± 1.72E-05 | 0.1892152 ± 1.8936216 | -1.40E-04 ± 1.39E-05 | -1.46E-04 ± 3.95E-05 | -1.11E-04 ± 1.56E-06 |
| | $\lambda$ | 17.815496 ± 2.2987532 | 17.619827 ± 3.2692295 | 18.288234 ± 2.1973919 | 19.3148 ± 2.9747414 | 22.357894 ± 1.1541713 |
| | $R_c$ | 0.0005745 ± 1.46E-04 | 0.0007202 ± 0.0025525 | 0.000505 ± 1.82E-04 | 0.0006057 ± 2.02E-04 | 0.0007966 ± 2.18E-05 |
| | $B$ | 0.0180224 ± 0.0053908 | 0.2119834 ± 1.909938 | 0.0189587 ± 0.0044109 | 0.1310208 ± 0.1350317 | 0.0165263 ± 6.87E-04 |
| | $J_n$ | 23.665732 ± 3.7645158 | 37.419977 ± 147.78442 | 23.639985 ± 4.4226959 | 22.859263 ± 7.4288673 | 29.892491 ± 0.4868794 |
| | $J_{\max}$ | 1492.4046 ± 4.6439227 | 1477.4357 ± 135.36285 | 1493.5394 ± 4.2682876 | 1256.94 ± 278.80288 | 1499.9137 ± 0.4798894 |
| | $\ell$ | 120.65452 ± 28.459245 | 112.04822 ± 29.381111 | 124.62694 ± 26.008335 | 132.16139 ± 41.217744 | 175.52731 ± 11.130183 |
| Criterion | MSE | **6.39E-05 ± 2.85E-05**$^\dagger$ | 1.04E-04 ± 9.56E-05$^\dagger$ | 6.50E-05 ± 1.66E-05$^\dagger$ | 1.44E-03 ± 1.01E-02$^\dagger$ | **4.75E-05 ± 3.41E-07** |
| | $\text{NFEs}_{\text{VTR}}$ | 8792.31 ± 883.01 | 9520.00 ± 554.08 | 9183.66 ± 786.47 | **4819.61 ± 894.21** | **3412.50 ± 1228.36** |
| | $S_r$ | **0.65** | 0.05 | 0.41 | 0.51 | **1.00** |

Hereinafter, "†" indicates that rank-MADE is significant better than its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

Table 6: Comparison on the Identified Parameters and Performance Criteria of Different Algorithms for the SR-12 Modular PEM Generator. The Target Fitness $f(\mathbf{x}^*)$ Is 0.156166272722352.

| | | SaDE | JADE | CoDE | DEGL | rank-MADE |
|---|---|---|---|---|---|---|
| Parameter | $\xi_1$ | -1.02561 ± 0.1030541 | -1.034749 ± 0.0721448 | -1.0229 ± 0.0801908 | -1.008984 ± 0.10693 | -0.998792 ± 0.1204101 |
| | $\xi_2$ | 0.0032459 ± 4.26E-04 | 0.0032937 ± 2.84E-04 | 0.0032539 ± 3.27E-04 | 0.0031744 ± 4.57E-04 | 0.0032155 ± 5.17E-04 |
| | $\xi_3$ | 6.73E-05 ± 1.67E-05 | 6.83E-05 ± 1.17E-05 | 6.82E-05 ± 1.61E-05 | 6.62E-05 ± 1.97E-05 | 7.09E-05 ± 2.06E-05 |
| | $\xi_4$ | -9.55E-05 ± 1.76E-07 | -9.74E-05 ± 1.75E-06 | -9.64E-05 ± 8.39E-07 | -9.54E-05 ± 1.97E-09 | -9.54E-05 ± 4.51E-12 |
| | $\lambda$ | 19.584996 ± 2.9809272 | 18.462946 ± 2.550822 | 19.085869 ± 2.9271004 | 23.921234 ± 0.3353184 | 23.999947 ± 3.30E-04 |
| | $R_c$ | 0.0001379 ± 5.79E-05 | 0.0003449 ± 1.19E-04 | 0.0002556 ± 1.22E-04 | 0.0001002 ± 3.21E-07 | 0.0001 ± 6.95E-10 |
| | $B$ | 0.1791865 ± 0.0063177 | 0.1580305 ± 0.0152112 | 0.1684005 ± 0.0094675 | 0.1861327 ± 2.79E-04 | 0.1861435 ± 7.97E-06 |
| | $J_n$ | 17.952174 ± 8.359753 | 15.803156 ± 6.1561746 | 17.404213 ± 6.5811275 | 16.267198 ± 8.9160532 | 15.339315 ± 9.9366125 |
| | $J_{\max}$ | 710.39021 ± 10.001811 | 696.36233 ± 17.83463 | 704.28625 ± 10.025273 | 713.20633 ± 8.8591155 | 712.24664 ± 9.9388476 |
| | $\ell$ | 30.328113 ± 10.095862 | 59.35475 ± 24.708502 | 44.937445 ± 15.285959 | 25.022364 ± 0.0309311 | 25.000054 ± 3.04E-04 |
| Criterion | MSE | 0.159124 ± 2.49E-03$^\dagger$ | 0.1821294 ± 1.10E-02$^\dagger$ | 0.1686544 ± 5.07E-03$^\dagger$ | **0.1561927 ± 6.49E-05**$^\dagger$ | **0.1561663 ± 1.28E-07** |
| | $\text{NFEs}_{\text{VTR}}$ | NA ± NA | NA ± NA | NA ± NA | **9384.62 ± 455.20** | **5719.50 ± 1127.16** |
| | $S_r$ | 0.00 | 0.00 | 0.00 | **0.52** | **1.00** |

### *4.3.1. Ballard Mark V FC*

For the Ballard Mark V FC, the results (mean ± standard deviation) of all DE variants are reported in Table 5. All results are averaged over 100 runs. The overall best and the second best results among the five DEs are highlighted in **grey boldface** and **boldface**, respectively. In order to statistically compare the significance in terms of the MSE values between two algorithms, the paired Wilcoxon signed-rank test at $\alpha = 0.05$ is adopted.

According to the results shown in Table 5, it is clear to see that rank-MADE obtains the overall best results among all DE variants.

- With respect to the MSE values, rank-MADE is able to get the significant better results compared with other DE variants based on the Wilcoxon's test. In addition, rank-MADE provides the smallest standard deviation of MSE, which means that this approach is the most robust one among the five compared DEs.
- For the $\text{NFEs}_{\text{VTR}}$, rank-MADE requires the smallest $\text{NFEs}_{\text{VTR}}$ values to reach $\epsilon \leq 1e-5$. This phenomenon indicates that rank-MADE can reduce the computational efforts to reach the required accuracy compared with other DEs.
- In terms of the success rate, Table 5 clearly shows that rank-MADE can successfully solve the parameter optimization problem of the Ballard Mark V FC in all runs. However, for other four DE variants, there are some runs that failed to reach the required accuracy ($1e-5$).

### *4.3.2. SR-12 modular PEM generator*

In this case, the results of all DEs are shown in Table 6. Similar to the results shown in the Ballard Mark V FC, rank-MADE obtains the best results among all DEs. DEGL gets the second best results in terms of all performance

Table 7: Comparison on the Identified Parameters and Performance Criteria of Different Algorithms for the 500-W BCS Stack. The Target Fitness $f(\mathbf{x}^*)$ Is 0.080620362614142.

| | | SaDE | JADE | CoDE | DEGL | rank-MADE |
|---|---|---|---|---|---|---|
| Parameter | $\xi_1$ | -1.028371 ± 0.1047142 | -1.040735 ± 0.0738692 | -1.034078 ± 0.0895158 | -1.033311 ± 0.1113914 | -1.026953 ± 0.1380159 |
| | $\xi_2$ | 0.0033601 ± 4.35E-04 | 0.0033744 ± 3.15E-04 | 0.0033413 ± 3.84E-04 | 0.003394 ± 4.58E-04 | 0.0032749 ± 4.86E-04 |
| | $\xi_2$ | 6.92E-05 ± 1.91E-05 | 6.78E-05 ± 1.35E-05 | 6.69E-05 ± 1.52E-05 | 7.04E-05 ± 1.88E-05 | 6.40E-05 ± 2.06E-05 |
| | $\xi_4$ | -2.60E-04 ± 6.11E-08 | -2.59E-04 ± 6.47E-07 | -2.60E-04 ± 3.89E-07 | -2.60E-04 ± 7.83E-10 | -2.60E-04 ± 1.86E-10 |
| | $\lambda$ | 21.415265 ± 1.7089912 | 18.291407 ± 2.4222491 | 20.404572 ± 1.9522797 | 22.042059 ± 0.3309158 | 22.022697 ± 0.2378777 |
| | $R_c$ | 0.0007645 ± 5.26E-05 | 0.0005368 ± 1.46E-04 | 0.0006849 ± 9.29E-05 | 0.0007997 ± 3.80E-07 | 0.0008 ± 1.02E-07 |
| | $B$ | 0.023261 ± 0.0132924 | 0.0599936 ± 0.0281342 | 0.0410432 ± 0.0217215 | 0.0140747 ± 0.0018136 | 0.0138872 ± 0.0017135 |
| | $J_n$ | 15.632572 ± 6.8891146 | 15.250586 ± 6.6975231 | 15.107213 ± 7.1807205 | 11.548915 ± 7.8967328 | 1.1688426 ± 0.92269 |
| | $J_{\max}$ | 1400.4734 ± 74.057285 | 1297.6723 ± 108.21698 | 1358.8213 ± 106.11027 | 1498.1705 ± 2.6615516 | 1499.8577 ± 0.7903497 |
| | $\ell$ | 161.69931 ± 21.961528 | 107.34752 ± 33.000504 | 135.58707 ± 28.43836 | 177.71679 ± 1.4661152 | 177.9474 ± 0.4192133 |
| Criterion | MSE | 0.0808961 ± 2.26E-04$^\dagger$ | 0.0826358 ± 6.39E-04$^\dagger$ | 0.0815721 ± 3.48E-04$^\dagger$ | **0.0806272 ± 1.17E-05**$^\dagger$ | **0.0806217 ± 8.05E-06** |
| | $\text{NFEs}_{\text{VTR}}$ | NA ± NA | NA ± NA | NA ± NA | **8162.64 ± 791.71** | **5136.73 ± 1331.10** |
| | $S_r$ | 0.00 | 0.00 | 0.00 | **0.91** | **0.98** |

Table 8: Comparison on the Identified Parameters and Performance Criteria of Different Algorithms for the Tesasek Fuel Cell. The Target Fitness $f(\mathbf{x}^*)$ Is 0.005350750634420.

| | | SaDE | JADE | CoDE | DEGL | rank-MADE |
|---|---|---|---|---|---|---|
| Parameter | $\xi_1$ | -1.031596 ± 0.0915411 | -1.030217 ± 0.08226 | -1.036283 ± 0.0870782 | -1.031217 ± 0.1111281 | -1.053657 ± 0.125614 |
| | $\xi_2$ | 0.0033444 ± 3.80E-04 | 0.0033394 ± 3.19E-04 | 0.003366 ± 3.75E-04 | 0.0033372 ± 4.64E-04 | 0.0034069 ± 5.45E-04 |
| | $\xi_3$ | 6.88E-05 ± 1.69E-05 | 6.71E-05 ± 1.40E-05 | 6.82E-05 ± 1.54E-05 | 6.87E-05 ± 1.88E-05 | 6.87E-05 ± 2.30E-05 |
| | $\xi_4$ | -9.59E-05 ± 7.21E-07 | -9.89E-05 ± 3.36E-06 | -9.86E-05 ± 2.17E-06 | -9.54E-05 ± 7.27E-10 | -9.54E-05 ± 9.92E-11 |
| | $\lambda$ | 18.124982 ± 2.8688591 | 18.166463 ± 3.0411002 | 18.72864 ± 3.0917908 | 20.030178 ± 3.649708 | 22.05994 ± 3.105797 |
| | $R_c$ | 2.02E-04 ± 7.14E-05 | 2.37E-04 ± 8.78E-05 | 2.61E-04 ± 1.05E-04 | 2.09E-04 ± 5.93E-05 | 2.47E-04 ± 5.07E-05 |
| | $B$ | 0.0337897 ± 0.0127113 | 0.0465626 ± 0.0263156 | 0.0305712 ± 0.0088909 | 0.0366364 ± 0.0023466 | 0.0372538 ± 5.09E-04 |
| | $J_n$ | 16.529477 ± 6.6541628 | 16.090957 ± 6.1018559 | 16.179831 ± 7.5686319 | 15.632442 ± 8.401901 | 16.725764 ± 11.738041 |
| | $J_{\max}$ | 623.84636 ± 56.936935 | 716.09655 ± 177.36801 | 625.04348 ± 32.192166 | 620.28937 ± 9.1103672 | 622.01113 ± 11.698825 |
| | $\ell$ | 49.298956 ± 16.672435 | 62.903887 ± 21.277323 | 59.997894 ± 22.309745 | 36.199719 ± 14.474108 | 30.255754 ± 11.486564 |
| Criterion | MSE | 0.0063898 ± 4.16E-03$^\dagger$ | 0.0220811 ± 8.90E-03$^\dagger$ | 0.0119779 ± 5.58E-03$^\dagger$ | **0.0053793 ± 2.38E-04**$^\dagger$ | **0.005353 ±3.46E-06** |
| | $\text{NFEs}_{\text{VTR}}$ | NA ± NA | NA ± NA | NA ± NA | **6873.33 ± 1237.69** | **4225.77 ±1344.13** |
| | $S_r$ | 0.00 | 0.00 | 0.00 | **0.90** | **0.97** |

criteria. rank-MADE is able to get the highest quality of solutions and the smallest $\text{NFEs}_{\text{VTR}}$ value. In addition, rank-MADE can also successfully reach the required accuracy of solutions in all runs ($S_r = 1.0$). DEGL only provides $S_r = 0.52$. While for other three DE methods (SaDE, JADE, and CoDE), there is no successful run in all 100 runs.

#### *4.3.3. BCS 500-W stack*

As the results shown in Table 7, in the case of BCS 500-W stack, the proposed rank-MADE gets the best performance in terms of all performance criteria, followed by DEGL. In 2 out of 100 runs, rank-MADE fails to reach $\epsilon \leq 1e-5$, but it gets success in the rest 98 runs. DEGL gets the success rate with $S_r = 0.91$. There is no success run for SaDE, JADE, and CoDE.

#### *4.3.4. Temasek FC*

The results are reported in Table 8 in this case. Like the previous cases, rank-MADE still provides the best results with respect the quality of solutions, the success rate, and the $\text{NFEs}_{\text{VTR}}$ in this case. DEGL gets the second best results. There is also no success run for SaDE, JADE, and CoDE in the case of the Temasek FC.

#### *4.3.5. On the convergence speed*

In order to compare the convergence speed of all DE variants, the convergence graphs in all above cases are plotted in Figure 1. As can be seen in Figure 1, it is clear to observe that the proposed rank-MADE consistently gets the fastest convergence speed in all cases, followed by DEGL, SaDE, CoDE, and JADE.

### *4.4. Statistical comparison among other EAs*

In the previous section, rank-MADE is compared with other DE variants through the experimental data of four different PEMFC models. To further understand the performance of rank-MADE, in this section, rank-MADE is compared with other evolutionary algorithms (EAs): *i.e.*, real-coded genetic algorithm (rcGA) [39], fast evolutionary

Table 9: Comparison on the MSE values among different EAs for different PEMFC models. All results are averaged over 100 independent runs.

| Algorithm | Ballard Mark V | SR-12 | BCS 500W | Temasek FC | WNS-FC (313 K) | WNS-FC (333 K) | WNS-FC (353 K) |
|---|---|---|---|---|---|---|---|
| MADE | **4.79E-05 ± 9.0E-07** | 0.15633 ± 5.0E-04 | 0.08065 ± 9.4E-05 | **0.00537 ± 1.1E-04** | 0.01378 ± 2.5E-04 | 0.01719 ± 1.4E-04 | **0.02135 ± 4.2E-05** |
| SaDE | 6.39E-05 ± 2.8E-05 | 0.15912 ± 2.5E-03 | 0.08090 ± 2.3E-04 | 0.00639 ± 4.2E-03 | 0.01451 ± 6.9E-04 | 0.01745 ± 1.8E-04 | 0.02164 ± 2.1E-04 |
| JADE | 1.04E-04 ± 9.6E-05 | 0.18213 ± 1.1E-02 | 0.08264 ± 6.4E-04 | 0.02208 ± 8.9E-03 | 0.02806 ± 1.1E-02 | 0.03153 ± 1.0E-02 | 0.04490 ± 1.4E-02 |
| CoDE | 6.50E-05 ± 1.7E-05 | 0.16865 ± 5.1E-03 | 0.08157 ± 3.5E-04 | 0.01198 ± 5.6E-03 | 0.01750 ± 1.6E-03 | 0.01932 ± 1.1E-03 | 0.02437 ± 2.0E-03 |
| DEGL | 1.44E-03 ± 1.0E-02 | **0.15619 ± 6.5E-05** | **0.08063 ± 1.2E-05** | 0.00538 ± 2.4E-04 | **0.01369 ± 1.9E-04** | **0.01713 ± 1.3E-04** | 0.02136 ± 3.1E-05 |
| rcGA | 6.39E+00 ± 2.0E+01 | 0.87105 ± 2.4E-01 | 0.13500 ± 3.3E-02 | 0.21249 ± 1.3E-01 | 0.46276 ± 1.3E-01 | 0.55347 ± 1.5E-01 | 0.67596 ± 1.5E-01 |
| FEP | 2.08E-04 ± 9.5E-05 | 0.64211 ± 1.5E-01 | 0.10391 ± 1.2E-02 | 0.07086 ± 1.5E-02 | 0.34010 ± 5.9E-02 | 0.39310 ± 6.7E-02 | 0.51039 ± 8.4E-02 |
| ABC | 8.77E-05 ± 3.8E-05 | 0.27410 ± 8.3E-02 | 0.09662 ± 1.3E-02 | 0.02724 ± 1.3E-02 | 0.10029 ± 5.4E-02 | 0.11328 ± 6.7E-02 | 0.14109 ± 8.1E-02 |
| CLPSO | 2.06E-03 ± 1.5E-02 | 0.18341 ± 2.8E-02 | 0.08393 ± 4.9E-03 | 0.01227 ± 9.5E-03 | 0.05582 ± 5.5E-02 | 0.07900 ± 7.2E-02 | 0.09909 ± 9.5E-02 |
| rank-MADE | **4.75E-05 ± 3.4E-07** | **0.15617 ± 1.3E-07** | **0.08062 ± 8.0E-06** | **0.00535 ± 3.5E-06** | **0.01355 ± 1.5E-04** | **0.01703 ± 7.2E-05** | **0.02131 ± 2.5E-05** |

Table 10: Statistical results by the Wilcoxon's test between rank-MADE with its competitor. "Yes" means that rank-MADE significantly outperforms its competitor in terms of the MSE values by the Wilcoxon's test at $\alpha = 0.05$.

| Algorithm | Ballard Mark V | | SR-12 | | BCS 500W | | Temasek FC | | WNS-FC (313 K) | | WNS-FC (333 K) | | WNS-FC (353 K) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *p*-value | sig. | *p*-value | sig. | *p*-value | sig. | *p*-value | sig. | *p*-value | sig. | *p*-value | sig. | *p*-value | sig. |
| MADE | 5.63E-12 | Yes | 0.00E+00 | Yes | 6.66E-16 | Yes | 1.07E-08 | Yes | 8.08E-12 | Yes | 3.38E-14 | Yes | 1.24E-11 | Yes |
| SaDE | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes |
| JADE | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes |
| CoDE | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes |
| DEGL | 4.10E-13 | Yes | 0.00E+00 | Yes | 8.22E-15 | Yes | 1.13E-06 | Yes | 1.30E-09 | Yes | 5.19E-11 | Yes | 2.51E-14 | Yes |
| rcGA | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes |
| FEP | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes |
| ABC | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes |
| CLPSO | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes | 0.00E+00 | Yes |

programming (FEP) [40], artificial bee colony (ABC) [41], and comprehensive learning particle swarm optimizer (CLPSO) [42]. The four algorithms are chosen due to their good performance obtained in the benchmark problems. In addition, besides the experimental data of the four PEMFC models mentioned above, the PEMFC model (WNS-FC, for short) proposed in [34] are also used to generate the *V*-*I* data at different temperature (313 K, 333 K, and 353 K). The Matlab/Simulink generator of this model is provided in [55]. For the sake of completeness, the above-mentioned DE variants are also compared in this experiment. In addition, the original version of rank-MADE, *i.e.*, MADE[1], is also selected for comparison. The parameter settings of all of the DE variants are shown in Table 4. The parameters of the four EAs (*i.e.*, rcGA, FEP, ABC, and CLPSO) are set the same as used in their corresponding literature. For each data set of different PEMFC models, the Max_NFEs $= 10,000$ are used. In addition, each algorithm is executed over 100 independent runs for each data set. For the sake of space limitation, only the MSE performance is used in this section.

The mean and standard deviation values of MSE for all problems are reported in Table 9. All results are averaged over 100 runs. The overall best and the second best results among the ten compared EAs are highlighted in **grey boldface** and **boldface**, respectively. Moreover, in order to make the comparison meaningful, the Wilcoxon's test is employed to compare the differences between rank-MADE and its competitor. The statistical results by the Wilcoxon's test are described in Table 10, where the *p*-value and the significance are reported. If $p < 0.05$, it indicates that rank-MADE significantly outperforms its competitor in terms of the MSE values. Additionally, according to the Friedman test, the final rankings of all EAs for all problems are given in Table 11. Note that the Friedman test, which is used to obtain the rankings of different algorithms for all problems, is calculated by the KEEL software [56].

According to the results shown in Table 9, it is clear that the proposed rank-MADE consistently obtains the best mean MSE values in all cases. In addition, in all of the 7 problems rank-MADE provides the smallest standard deviation values of MSE, which means that rank-MADE is the most robust method among the 10 compared EAs. In 4 out of 7 cases, DEGL gets the second best MSE values, while in 3 cases MADE is the second best one. Compared the results between MADE and rank-MADE, it can be seen that rank-MADE improves the performance of MADE consistently in all cases, which verifies the benefit of integration of the ranking-based vector selection technique.

Considering the statistical results, Table 10 clearly indicates that rank-MADE gets significantly better results than its competitors in all problems by the Wilcoxon's test at $\alpha = 0.05$. Especially, comparing rank-MADE with SaDE,

[1]The only difference between rank-MADE and MADE is the ranking-based vector selection proposed in Section 3.3. In MADE, the ranking-based vector selection is not used. MADE have the same parameter settings as used in rank-MADE.

Table 11: Average rankings of the algorithms by the Friedman's test in terms of the mean MSE values in all of the PEMFC models.

| Algorithm | Ranking |
|---|---|
| MADE | 2.5714 |
| SaDE | 3.8571 |
| JADE | 6.1429 |
| CoDE | 4.8571 |
| DEGL | 3.1429 |
| rcGA | 10.000 |
| FEP | 8.7143 |
| ABC | 7.5714 |
| CLPSO | 7.1429 |
| rank-MADE | 1.0000 |

JADE, CoDE, rcGA, ABC, FEP, and CLPSO, the $p$-values are 0.0 for all problems, which mean that in all 100 runs rank-MADE is able to obtain better MSE values than the 7 methods. In addition, according to the Friedman test, from Table 11, it can be observed that rank-MADE gets the first overall ranking in all problems, followed by MADE, DEGL, SaDE, CoDE, JADE, CLPSO, ABC, FEP, and rcGA.

In general, according to the results shown in Tables 9, 10, and 11, it can be concluded that the proposed rank-MADE is very promising when solving the parameter optimization problems of PEMFC models. It is able to consistently provide the best results in all problems compared with other DE variants and other EAs.

### *4.5. On the V-I characteristics*

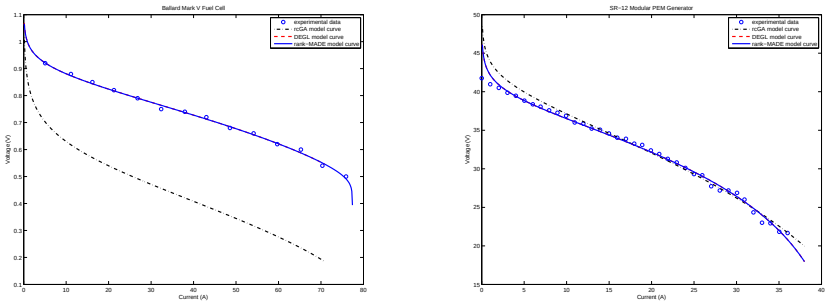
In order to further investigate the capability of the experimental data and the model output voltage obtained by rank-MADE, DEGL, and rcGA[2], their optimal parameter values are fed back to the PEMFC mathematical model to achieve *V-I* characteristics. The comparisons between the experimental data and the simulated data are plotted in Figure 2. As shown in Figure 2, the shapes of the fitted curves of rank-MADE and DEGL are very close to each other, and the *V-I* characteristics obtained by both of them are highly coincide with the experimental data in all four cases. On the contrary, the *V-I* characteristics obtained by rcGA are in bad agreement with the experimental data in all problems due to its poor optimized parameters. It it worth emphasizing that although the fitted curves of rank-MADE and DEGL are very close to each other, rank-MADE is able to provide better mean and standard deviation values of MSE than those of DEGL in all problems as reported in Table 9.

### *4.6. Discussions*

With the purpose of optimizing the unknown parameters of PEMFC model faster and more accurate, in this work, a ranking-based multi-strategy adaptive DE (rank-MADE) method is presented. Then, this approach is employed for the parameter optimization problems of the Ballard Mark V FC, the SR-12 Modular PEM Generator, the BCS 500-W stack, the Temasek FC, and WNS-FC at different temperature. The performance of the approach is compared with five highly-competitive DE variants and four state-of-the-art EAs. From the experimental results it can be obtained that:

- In terms of the solution quality, the convergence speed, and the success rate, rank-MADE is the best one among all compared algorithms. It is capable of providing higher quality of solutions, getting higher success rate, and converging faster in all cases. rank-MADE is also the most robust one according to the standard deviation values of MSE.
- By returning the optimized parameters by rank-MADE to the PEMFC mathematical model, the *V-I* characteristics obtained by this approach are in very good agreement with the experimental data in the four cases. This means that rank-MADE is effective to solve the parameter optimization problems of PEMFC model.

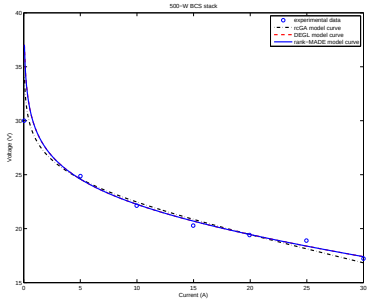
[2]Note that for the sake of clarity, we only show the *V-I* characteristics of rank-MADE, DEGL, and rcGA in Figure 2. DEGL is selected due to its second best results obtained in four cases; while rcGA is chosen because of its worst performance obtained in all problems.



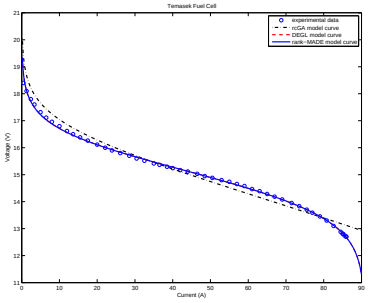

(a)



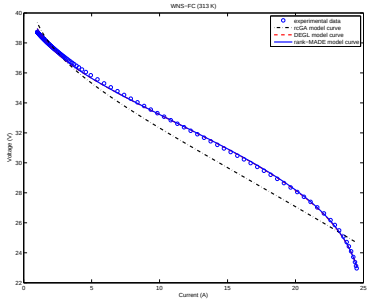

(b)



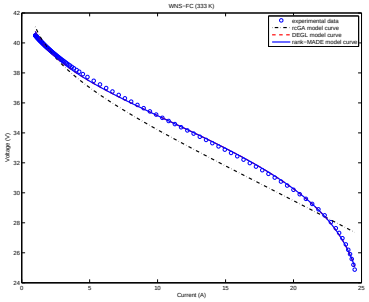

(c)



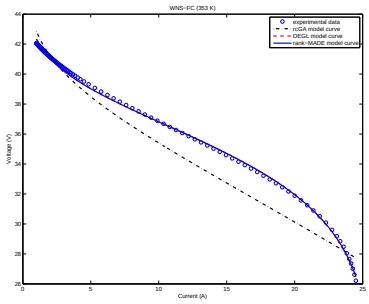

(d)




(e)




(f)




(g)

Figure 2: Comparisons between the experimental data and the model curve obtained from the identified parameters by rank-MADE, DEGL, and rcGA on 1(a) the Ballard Mark V FC, 1(b) the SR-12 Modular PEM Generator, 1(c) the BCS 500-W stack, 1(d) the Temasek FC, 2(e) the WNS-FC at 313 K, 2(f) the WNS-FC at 333 K, and 2(g) the WNS-FC at 353 K.

- Among the 10 methods, rank-MADE ranks the first, followed by MADE, DEGL, SaDE, CoDE, JADE, CLPSO, ABC, FEP, and rcGA. Interestingly, rank-MADE, DEGL, SaDE, and CoDE are all multi-strategy based DE variants. This might motivate the researchers to study new multi-strategy adaptive DE to solve the complex real-work application problems.

- By carefully looking at the results shown in Table 9, the mean MSE values of rank-MADE, MADE, and DEGL are very close to each other. However, since the *V-I* data adopted in this work are stemmed from [5, 33] and generated from the model proposed in [34], no information is available about the accurate values of the parameters; therefore, any reduction in the objective function value is significant because it results in improvement in the knowledge about the real values of the parameters.

## 5. Conclusions

In order to obtain higher quality of solutions and reduce the computational efforts, in this paper, two improvements in the DE literature are synergized and the ranking-based multi-strategy adaptive DE method is proposed for the parameter optimization problems of PEMFC model. In rank-MADE, the multi-strategy adaptive selection is employed to release the engineer to choose the most suitable strategy for a specific problem at hand. In addition, the ranking-based vector selection technique is applied to each mutation strategy to make the algorithm converge faster. Experiments on different PEMFC models and comparisons with other advanced DE variants and other EAs verify the expectation that this approach obtains more accurate solutions and converges faster. In addition, it gets the higher success rate in all cases of experimental data. Moreover, the *V-I* characteristics obtained by rank-MADE are in very good agreement with the experimental data.

Due to the superior performance obtained by rank-MADE for parameter optimization of PEMFC model,one possible future work is the application of GPGPU-aided rank-MADE [57] to other complex parameter optimization problems of fuel cell models, such as the solid oxide fuel cell [58]. In addition, since rank-MADE is not a special technique for fuel cells and it does not increase the complexity of the original DE algorithm, it can also be useful to other real-world problems like other DEs [30], such as the modeling of oxygen mass transfer [25], engineering design [59], and so on. This expectation will be verified in the future work.

The source code of rank-MADE can be obtained from the first author upon request.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive suggestions. 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] J. Larminie, A. Dicks, Fuel Cell Systems Explained (Second Edition), Wiley, 2003.
[2] S. Peighambardoust, S. Rowshanzamir, M. Amjadi, Review of the proton exchange membranes for fuel cell applications, International Journal of Hydrogen Energy 35 (17) (2010) 9349 – 9384.
[3] 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.
[4] 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.
[5] 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.
[6] A. Bıyıkoğlu, Review of proton exchange membrane fuel cell models, International Journal of Hydrogen Energy 30 (11) (2005) 1181 – 1212.
[7] D. Paclisan, W. Charon, Real time modelling of the dynamic mechanical behaviour of PEMFC thanks to neural networks, Engineering Applications of Artificial Intelligence 26 (2) (2013) 706 – 713.
[8] 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.

[9] 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.
[10] S. M. C. Ang, E. S. Fraga, N. P. Brandon, N. J. Samsatli, D. J. Brett, Fuel cell systems optimisation - methods and strategies, International Journal of Hydrogen Energy 36 (22) (2011) 14678 – 14703.
[11] 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.
[12] M. Ye, X. Wang, Y. Xu, Parameter identification for proton exchange membrane fuel cell model using particle swarm optimization, International Journal of Hydrogen Energy 34 (2) (2009) 981 – 989.
[13] A. Askarzadeh, A. Rezazadeh, A new artificial bee swarm algorithm for optimization of proton exchange membrane fuel cell model parameters, Journal of Zhejiang University SCIENCE C 12 (2011) 638–646.
[14] 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.
[15] 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.
[16] 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.
[17] 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.
[18] A. Askarzadeh, A. Rezazadeh, A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: Bird mating optimizer, International Journal of Energy Research (2012) n/a–n/aIn press.
[19] U. K. Chakraborty, T. E. Abbott, S. K. Das, PEM fuel cell modeling using differential evolution, Energy 40 (1) (2012) 387 – 399.
[20] 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.
[21] 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.
[22] K. Price, R. Storn, J. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer-Verlag, Berlin, 2005.
[23] R. Thangaraj, M. Pant, K. Deep, Optimal coordination of over-current relays using modified differential evolution algorithms, Engineering Applications of Artificial Intelligence 23 (5) (2010) 820 – 829.
[24] K. Vaisakh, L. Srinivas, Evolving ant direction differential evolution for OPF with non-smooth cost functions, Engineering Applications of Artificial Intelligence 24 (3) (2011) 426 – 436.
[25] E.-N. Dragoi, S. Curteanu, F. Leon, A.-I. Galaction, D. Cascaval, Modeling of oxygen mass transfer in the presence of oxygen-vectors using neural networks developed by differential evolution algorithm, Engineering Applications of Artificial Intelligence 24 (7) (2011) 1214 – 1226.
[26] S.-K. Oh, W.-D. Kim, W. Pedrycz, Design of optimized cascade fuzzy controller based on differential evolution: Simulation studies and practical insights, Engineering Applications of Artificial Intelligence 25 (3) (2012) 520 – 532.
[27] E. Cuevas, D. Zaldívar, M. Pérez-Cisneros, D. Oliva, Block-matching algorithm based on differential evolution for motion estimation, Engineering Applications of Artificial Intelligence 26 (1) (2013) 488 – 498.
[28] W. Gong, Z. Cai, Parameter extraction of solar cell models using repaired adaptive differential evolution, Solar Energy 94 (2013) 209 – 220.
[29] F. Neri, V. Tirronen, Recent advances in differential evolution: A survey and experimental analysis, Artificial Intelligence Review 33 (1-2) (2010) 61 – 106.
[30] 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.
[31] D. E. Goldberg, Probability matching, the magnitude of reinforcement, and classifier system bidding, Mach. Learn. 5 (4) (1990) 407–425.
[32] W. Gong, Z. Cai, Differential evolution with ranking-based mutation operators, IEEE Transactions on Cybernetics (2013) 1 – 16In press.
[33] J. Jia, Y. Wang, Q. Li, Y. Cham, M. Han, Modeling and dynamic characteristic simulation of a proton exchange membrane fuel cell, IEEE Transactions on Energy Conversion 24 (1) (2009) 283 – 291.
[34] C. Wang, M. Nehrir, S. Shaw, Dynamic models and model validation for PEM fuel cells using electrical circuits, IEEE Transactions on Energy Conversion 20 (2) (2005) 442 – 451.
[35] 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.
[36] J. Zhang, A. C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Trans. on Evol. Comput. 13 (5) (2009) 945–958.
[37] 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.
[38] 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.
[39] F. Herrera, M. Lozano, Gradual distributed real-coded genetic algorithms, IEEE Trans. on Evol. Comput. 4 (1) (2000) 43–63.
[40] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. on Evol. Comput. 3 (2) (1999) 82–102.
[41] 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.
[42] 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.
[43] D. Opitz, R. Maclin, Popular ensemble methods: An empirical study, Journal of Artificial Intelligence Research 11 (1999) 169–198.
[44] W. Gong, A. Fialho, Z. Cai, Adaptive strategy selection in differential evolution, in: J. Branke (Ed.), Genetic and Evolutionary Computation Conference (GECCO 2010), ACM Press, 2010, pp. 409–416.
[45] 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.

[46] 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.
[47] M. Ohenoja, K. Leiviskä, Validation of genetic algorithm results in a fuel cell model, International Journal of Hydrogen Energy 35 (22) (2010) 12618 – 12625.
[48] 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.
[49] 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.
[50] 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.
[51] 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.
[52] Y.-S. Ong, A. J. Keane, Meta-Lamarckian learning in memetic algorithms, IEEE Trans. on Evol. Comput. 8 (2) (2004) 99–110.
[53] D. Thierens, An adaptive pursuit strategy for allocating operator probabilities, in: Proc. Genetic Evol. Comput. Conf., 2005, pp. 1539–1546.
[54] W. Gong, Z. Cai, C. X. Ling, H. Li, Enhanced differential evolution with adaptive strategies for numerical optimization, IEEE Transactions on Systems, Man, and Cybernetics: Part B – Cybernetics 41 (2) (2011) 397–413.
[55] C. Wang, M. H. Nehrir, Dynamic models for PEMFs and tubular SOFCs (2013).
URL `http://www.coe.montana.edu/ee/fuelcell/`
[56] J. Alcalá-Fdez, L. Sánchez, S. García, KEEL: A software tool to assess evolutionary algorithms to data mining problems (2012).
URL `http://www.keel.es/`
[57] D. Chen, D. Li, M. Xiong, H. Bao, X. Li, GPGPU-aided ensemble empirical-mode decomposition for EEG analysis during anesthesia, IEEE Transactions on Information Technology in Biomedicine 14 (6) (2010) 1417–1427.
[58] D. Bhattacharyya, R. Rengaswamy, A review of solid oxide fuel cell (SOFC) dynamic models, Industrial & Engineering Chemistry Research 48 (13) (2009) 6068–6086.
[59] I. Mazhoud, K. Hadj-Hamou, J. Bigeon, P. Joyeux, Particle swarm optimization for solving engineering problems: A new constraint-handling mechanism, Engineering Applications of Artificial Intelligence 26 (4) (2013) 1263 – 1273.