Parameter extraction of different fuel cell models with transferred adaptive differential evolution ☆

Wenyin Gong*[a,b], Xuesong Yan[a,b], Xiaobo Liu[c], Zhihua Cai[a,b]

[a]*Hubei Intelligent Geo-information Processing Key Laboratory (China University of Geosciences),*
Wuhan 430074, P.R. China
[b]*School of Computer Science,*
China University of Geosciences, Wuhan 430074, P.R. China
[c]*School of Automation,*
China University of Geosciences, Wuhan 430074, P.R. China

Abstract

To improve the design and control of fuel cell (FC) models, it is important to extract their unknown parameters. Generally, the parameter extraction problems of FC models can be transformed as nonlinear and multi-variable optimization problems. To extract the parameters of different FC models exactly and fast, in this paper, we propose a transferred adaptive differential evolution (DE) framework, in which the successful parameters of the adaptive DE solving previous problems are properly transferred to solve new optimization problems in the similar problem-domains. Based on this framework, an improved adaptive DE method (TRADE, in short) is presented as an illustration. To verify the performance of our proposal, TRADE is used to extract the unknown parameters of two types of fuel cell models, *i.e.*, proton exchange membrane fuel cell (PEMFC) and solid oxide fuel cell (SOFC). The results of TRADE are also compared with those of other state-of-the-art evolutionary algorithms (EAs). Even though the modification is very simple, the results indicate that TRADE can extract the parameters of both PEMFC and SOFC models exactly and fast. Moreover, the *V-I* characteristics obtained by TRADE agree well with the simulated and experimental data in all cases for both types of fuel cell models. Also, it improves the performance of the original adaptive DE significantly in terms of both the quality of final solutions and the convergence speed in all cases. Additionally, TRADE is able to provide better results compared with other EAs.

Key words: Fuel cell, parameter extraction, differential evolution

1. Introduction

In recent years, the fuel cell (FC) technology has received considerable attention because of the low emission to environment, superior durability, good transient responses, high energy efficiency, high scalability, and so on [39]. There are several different types of fuel cells based on the nature of the electrolyte used, such as proton exchange membrane fuel cell (PEMFC), molten carbonate fuel cell (MCFC), solid oxide fuel cell (SOFC), and so on. Among them, PEMFC and SOFC have been widely studied and used in promising area for different applications[12, 61]. More details can be found in the following representative review papers [54, 41, 70, 40].

Although many PEMFC and SOFC models have been developed, their model parameters must be accurately extracted and optimized all the time to obtain high-performance system control. Generally, parameter extraction of PEMFC and SOFC models can be converted into numerical optimization problems. However, due to the nonlinear and multi-variable features, it may cause difficulties to traditional optimization techniques. Recently, the heuristic

☆This work was partly supported by the National Natural Science Foundation of China under Grant Nos. 61203307 and 61375066, the Natural Science Foundation of Hubei Province under Grant No. 2013CFA004, the Fundamental Research Funds for the Central Universities at China University of Geosciences (Wuhan) under Grant No. CUG130413.

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

optimization techniques have been used to solve the problems [2], for example such as genetic algorithms [49], simulated annealing [51], particle swarm optimization [78], harmony search [5], seeker optimization algorithm [17], artificial immune system [4], P systems based optimization algorithm [76], differential evolution [13] for PEMFC, and genetic algorithm [75], differential evolution [29] for SOFC, etc. However, in order to exactly and fast solve the parameter extraction problems of PEMFC and SOFC models, it is necessary to investigate more efficient optimization techniques to reduce the necessary computational efforts to achieve an optimal design [62].

In the similar problem-domains, different optimization problems always have similar features. For example, the parameter extraction problems of PEMFC and SOFC models can be converted into the optimization problems that have similar objective functions [49, 75]. Therefore, when the optimization techniques are used to find the optimal solutions for similar problems, the previous problem-solving experiences maybe benefit to solve new problems.

Differential evolution (DE), proposed by Storn and Price [65], is a simple and efficient evolutionary algorithm (EA) for global numerical optimization. It has been obtained many successful applications in diverse fields [19], such as engineering design, digital filter design [58], optimal power flow [66], simulation of solar-thermal refrigeration systems [23], hydrothermal generation scheduling [72, 79], etc. In DE, the parameter settings of the crossover rate CR and the scaling factor F are crucial to DE's performance [24]. To improve its performance, different parameter adaptation techniques have been proposed in the DE literature [45, 11, 59, 80]. However, in the current work, there is an apparent impediment, *i.e.*, the successful parameters solving previous problems do not be reused and transferred to solve new problems. For example, in JADE [80], the parameters of μ_{CR} and μ_F are usually initialized to be 0.5 when solving new optimization problems, but without considering their previous problem-solving experiences.

In machine learning, transfer learning [53] is a new learning framework that can transfer knowledge across different problems to improve the performance of learning. Inspired by the success of transfer learning, in this paper, a transferred adaptive DE framework is proposed, where the successful parameters of the adaptive DE solving previous problems are properly reused and transferred to solve new optimization problems in the similar problem-domains. As an example, based on the proposed framework, the parameters of μ_{CR} and μ_F in JADE [80] are initialized based on their previous problem-solving experiences when solving new encountered problems. The improved JADE method is referred to as TRADE, in short. TRADE is used to extract the unknown parameters of two different types of FC models, *i.e.*, PEMFC model and SOFC model [39]. The reasons of selecting these two models are two-fold: i) The parameter extraction problems of PEMFC and SOFC models are very important to improve the performance of these two FC models, and hence, they have obtained considerable attention recently [51, 8, 75]. ii) Both of these problems can be formulated as the optimization problems, and they have similar objective functions. To the best of our knowledge, it is the first attempt to combine DE with transfer learning to solve the parameter extraction problems of both PEMFC and SOFC models simultaneously.

The main contributions of this paper are three-fold:

i) A framework of transferred adaptive differential evolution is proposed, which is simple and generic.

ii) Based on the framework, an improved JADE method, referred to as TRADE, is presented to extract the parameters of two different FC models.

iii) TRADE is highly efficient and effective to extract the parameters of both PEMFC and SOFC models, which can be an efficient alternative to other complex optimization problems of FC models.

The rest of this paper is organized as follows. Section 2 briefly introduces the parameter adaptation technique of JADE and the problem formulations of PEMFC and SOFC models used in this work. In Section 3, the related work is presented. The proposed transferred adaptive DE framework and the TRADE method are elucidated in Section 4. Followed by Section 5, the results are described and analyzed in this section. Finally, Section 6 concludes the work and points out some possible future work.

2. Preliminaries

In this section, the problem formulations of PEMFC and SOFC models and the parameter adaptation technique of JADE proposed in [80] are briefly introduced.

2.1. *Problem Formulations*

A fuel cell (FC) is an electrochemical device that converts chemical energy directly into electrical energy [39]. According to the nature of used electrolyte, there are several different types of FCs. However, among various types of FCs, PEMFC and SOFC have obtained considerable attention in recent years. In this section, the two FC models are briefly introduced as follows.

2.1.1. *Mathematical Formulation of PEMFC Stack Model*

In this work, the PEMFC stack model presented in [49] is used. For n_{cell} cells connected in series to form a stack, the terminal voltage of the stack can be calculated by [16],

$$V_{\text{PEMFC}} = n_{\text{cell}} \cdot (E_{\text{Nernst}} - V_{\text{act}} - V_{\text{ohm}} - V_{\text{con}}) \tag{1}$$

where E_{Nernst} is the thermodynamic potential defined by

$$E_{\text{Nernst}} = 1.229 - 0.846 \times 10^{-3} \cdot (T - 298.15) + 4.3085 \times 10^{-5} \cdot T \cdot \ln\left(P^*_{\text{H}_2}\sqrt{P^*_{\text{O}_2}}\right)$$

The activation overpotential V_{act}, including anode and cathode, can be expressed by the following formula [48]

$$V_{\text{act}} = -\left[\xi_1 + \xi_2 \cdot T + \xi_3 \cdot T \cdot \ln\left(\frac{P^*_{\text{O}_2}}{5.08 \times 10^6 \cdot \exp(-498/T)}\right) + \xi_4 \cdot T \cdot \ln(i_{\text{cell}})\right]$$

The ohmic voltage drop V_{ohm} can be determined by the following expression [48]

$$V_{\text{ohm}} = (i_{\text{cell}} + i_n) \cdot \left(\frac{\ell}{A} \cdot \frac{181.6 \cdot \left[1 + 0.03 \cdot \left(\frac{i_{\text{cell}}+i_n}{A}\right) + 0.062 \cdot \left(\frac{T}{303}\right) \cdot \left(\frac{i_{\text{cell}}+i_n}{A}\right)^{2.5}\right]}{\left[\lambda - 0.634 - 3 \cdot \left(\frac{i_{\text{cell}}+i_n}{A}\right)\right] \cdot \exp\left[4.18 \cdot \left(\frac{T-303}{T}\right)\right]} + R_C\right)$$

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 [15]

$$V_{\text{con}} = -B \cdot \ln\left(1 - \frac{i_{\text{cell}} + i_n}{i_{\max}}\right)$$

In the above equations, the temperature T, the hydrogen partial pressure $P^*_{\text{H}_2}$, and the oxygen partial pressure $P^*_{\text{O}_2}$ are measurable and dependent on the operating conditions of the system. i_{cell} is the cell current. The other 11 parameters $\xi_1, \xi_2, \xi_3, \xi_4, \lambda, R_C, B, i_n, i_{\max}, \ell,$ and A are unknown that need to be extracted to improve the design of PEMFC model. When the optimization techniques are used to the parameter extraction problems of PEMFC model, the objective function should be defined at first. In the literature, the mean squared error (MSE) between the output voltage of the *actual* PEMFC stack and the model output voltage are used as the objective function [49]:

$$\min \quad f_{\text{PEMFC}}(\mathbf{x}) = \frac{1}{N} \cdot \sum_{k=1}^{N} (V_{\text{PEMFC,sa},k} - V_{\text{PEMFC,so},k})^2 \tag{2}$$

where $\mathbf{x} = \{\xi_1, \xi_2, \xi_3, \xi_4, \lambda, R_C, B, i_n, i_{\max}, \ell, A\}$, $V_{\text{PEMFC,sa}}$ is the output voltage of the actual PEMFC stack, $V_{\text{PEMFC,so}}$ is the model output voltage calculated by Equation (1), and N is the number of the data points. The search ranges of these unknown parameters are reported in Table 1. Note that, in Table 1, $J_n = i_n \times A$ is the current density of the cell, and $J_{\max} = i_{\max} \times A$.

Table 1: Search Ranges of the Unknown Parameters to Be Optimized for the PEMFC Model.

Parameter	ξ_1	ξ_2	ξ_3	ξ_4	λ	R_c (Ω)	B (V)	J_n (mA/cm^2)	$J_{\max}$ (mA/cm^2)	ℓ (μm)	A (cm^2)
Lower bound	-1.1997	0.001	-3.60E-05	-2.60E-04	10	0.0001	0.0136	1	500	20	20
Upper bound	-0.8532	0.005	-9.80E-05	-9.54E-05	24	0.0008	0.5	30	1500	200	100

2.1.2. Mathematical Formulation of SOFC Stack Model

To implement the control of an SOFC stack for the output performance, a simple electrochemical model is presented in [39, 14, 29], which is modeled by:

$$V_{\mathrm{SOFC}} = N_{\mathrm{cell}} \cdot \left[E_0 - A_{\mathrm{SOFC}} \sinh^{-1}\left(\frac{I}{2I_{0,\mathrm{a}}}\right) - A_{\mathrm{SOFC}} \sinh^{-1}\left(\frac{I}{2I_{0,\mathrm{c}}}\right) - IR_{\mathrm{ohm}} + B_{\mathrm{SOFC}} \ln\left(1 - \frac{I}{I_L}\right)\right] \tag{3}$$

where N_{cell} is the number of cells in the SOFC stack, E_0 is the open-circuit voltage, A_{SOFC} is the slope of Tafel line, R_{ohm} is the area-specific resistance in kΩ, B_{SOFC} is a constant that depends on the fuel cell and its operating state, I is the current in mA, and I_L is the limit current in mA.

In the above SOFC electrochemical model formulated by Equation (3), there are seven unknown parameters, *i.e.*, E_0, A_{SOFC}, $I_{0,a}$, $I_{0,c}$, R_{ohm}, B_{SOFC}, and I_L, which should be extracted by the optimization algorithm. When the optimization techniques are used to the parameter extraction problems of SOFC model, the mean squared error (MSE) is used as the objective function [75]:

$$\min \quad f_{\mathrm{SOFC}}(\mathbf{x}) = \frac{1}{N} \cdot \sum_{k=1}^{N} (V_{\mathrm{SOFC,sa},k} - V_{\mathrm{SOFC,so},k})^2 \tag{4}$$

subject to

$$I_k < I_L, \quad k = 1, \cdots, N$$

where $\mathbf{x} = \{E_0, A_{\mathrm{SOFC}}, I_{0,\mathrm{a}}, I_{0,\mathrm{c}}, R_{\mathrm{ohm}}, B_{\mathrm{SOFC}}, I_L\}$, N is the number of the sample data, and I_k is the k^{th} current in the sample data. $V_{\mathrm{SOFC,sa}}$ is the output voltage of the actual SOFC stack and $V_{\mathrm{SOFC,so}}$ is the model output voltage calculated by Equation (3). For the seven unknown parameters, the search ranges are tabulated in Table 2.

Table 2: Search Ranges of the Unknown Parameters to Be Optimized for the SOFC Model.

Parameter	E_0 (V)	A_{SOFC} (V)	$I_{0,\mathrm{a}}$ (mA)	$I_{0,\mathrm{c}}$ (mA)	R (kΩ)	B_{SOFC} (V)	I_L (mA)
Lower bound	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Upper bound	1.2	1.0	30.0	30.0	1.0	1.0	200.0

2.2. Parameter Adaptation in JADE

Differential evolution (DE), which was firstly proposed by Storn and Price in 1995 [64, 65], is one of the most powerful evolutionary algorithms for global numerical optimization. The advantages of DE are its ease of use, simple structure, speed, efficacy, and robustness. In the last few years, DE has obtained many successful applications in diverse domains, such as engineering optimal design, digital filter design, image processing, data mining, multisensor fusion, and so on [58, 19]. However, the performance of DE is sensitive to its parameter settings [24]. Based on this consideration, some researchers studied the parameter adaptation techniques to improve DE's performance, such as jDE [11], SaDE [59], JADE [80], SaJADE [28], and so on. Because the parameter adaptation technique in JADE is used in this work, it is briefly described as follows.

In JADE [80], the parameters CR and F are adaptively controlled by μ_{CR} and μ_F, respectively. Initially, $\mu_{CR} = 0.5$ and $\mu_F = 0.5$ are used for new problems when there is no prior knowledge. Subsequently, they are updated based on their previous successful parameters during the evolution process.

At each generation, for each target vector, its crossover rate CR_i is independently generated as follows:

$$CR_i = \mathrm{rndn}_i(\mu_{CR}, 0.1) \tag{5}$$

and truncated to the interval $[0, 1]$, where μ_{CR} is the mean value of the normal distribution used to generate CR_i, with standard deviation 0.1. The μ_{CR} is updated as follows:

$$\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \mathrm{mean}_A(S_{CR}) \tag{6}$$

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

In order to maintain some level of population diversity, for each target vector, the mutation factor F_i is independently calculated as:

$$F_i = \text{rndc}_i(\mu_F, 0.1) \tag{7}$$

and then truncated to be 1.0 if $F_i > 1.0$ or regenerated if $F_i \leq 0$. $\text{rndc}_i(\mu_F, 0.1)$ is a random number generated according to the Cauchy distribution with location parameter μ_F and scale parameter 0.1. The location parameter μ_F is updated in the following manner:

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F) \tag{8}$$

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

$$\text{mean}_L(S_F) = \frac{\sum_{i=1}^{|S_F|} F_i^2}{\sum_{i=1}^{|S_F|} F_i} \tag{9}$$

3. Related Work

3.1. EA-based Parameter Extraction of FC Models

As shown in Equations (1) and (3), there are some unknown parameters in both PEMFC and SOFC models. For the control of high performance, these important unknown parameters need to be extracted and optimized. However, there are two challenges to extract the parameters [51, 4, 75]: i) there is no sufficient exact procedure for parameter extraction; and ii) it is a complex nonlinear and multi-variable problem. These challenges make the traditional optimization techniques difficult to solve the parameter extraction problems of FC models effectively. Therefore, the use of evolutionary algorithms for these problems has obtained considerable attention recently.

3.1.1. Parameter Extraction of PEMFC Model

In [49], 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. Outeiro *et al.* [51, 52] applied the simulated annealing (SA) as optimization technique to extract the parameters of PEMFC model. In [78], a particle swarm optimizer (PSO)-based parameter optimization technique of PEMFC model was presented according to the V-I characteristics. Ohenoja and Leiviskä [50] 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. In [42], 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 [7] 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 [6], 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 [5, 38, 8], 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.* [17] proposed a novel seeker optimization algorithm (SOA), which is based on the concept of simulating human searching behaviors. In [4], the artificial immune system (AIS)-based parameter extraction of PEMFC model was present, and its results is compared with those of GAs and PSO. Yang and Wang [76] proposed a novel bio-inspired P systems-based optimization algorithm (BIPOA) to solve the PEMFC model parameter optimization problems. In [13], a DE variant, *i.e.*, DEGL [18], was employed for the parameter optimization problem of PEMFC stack. Inspired by the mechanism of biological RNA, Zhang and Wang [81] presented an adaptive RNA GA (ARNA-GA) for estimating the PEMFC model parameters. In [9], Askarzadeh and Rezazadeh proposed a bird mating optimizer (BMO), which is inspired by the intelligent behavior of birds during mating season. In [25], the ranking-based DE [26] is employed for the parameter extraction of PEMFC model. Ensemble of different improvements of DE, Gong and Cai [27] developed the rank-MADE, where the multi-strategy adaptation and ranking-based vector selection techniques are synergized, to extract the parameters of PEMFC model. In [3], a backtracking search algorithm combined with Burger's chaotic map was proposed to efficiently estimate the unknown parameters of the PEMFC electrochemical-based model.

3.1.2. Parameter Extraction of SOFC Model

In [73, 74], Wu *et al.* proposed a GA-RBF neural network method for modeling and predictive control of SOFC, where the genetic algorithm (GA) is used to optimize the parameters of RBF neural networks. Yang *et al.* [75] presented an improved GA (IGA) method for parameter optimization of tubular SOFC stack. In [75], the IGA method is used to optimize the parameters of a simple electrochemical model to fit the simulated data of the dynamic SOFC model. Li *et al.* proposed a model predictive control strategy based on GA for the SOFC control problem [43], where a support vector machine model is identified to approximate the behavior of the SOFC system and GA is used to solve the constrained predictive control problem. In [10], Bozorgmehri and Hamedi proposed an artificial neural network (ANN) and a GA-based method to model and optimize cell parameters to improve the performance of singular, intermediate-temperature SOFCs (IT-SOFCs). The ANN is used to model the SOFC performance through using experimental data, and GA is employed to optimize the SOFC parameters, *i.e.*, anode support thickness, anode support porosity, electrolyte thickness, and functional layer cathode thickness [10]. In [63], GA is utilized to estimate the electrode microstructure distributions in NASA Bi-electrode supported SOFCs. Gong *et al.* [29] proposed an adaptive DE, in which the ranking-based vector selection and crossover rate repairing technique are combined to optimize the parameters of SOFC model.

3.2. EAs with Knowledge Reusage

As stated in [47], “Problems seldom exist in isolation”. If previous problem-solving knowledge can be reused to solve new similar problems, the performance of the optimization techniques may be enhanced. Recently, some researchers tried to improve the performance of EAs by reusing the knowledge. It can be briefly classified into two categories: i) EAs based on case-based reasoning (CBR), and ii) EAs based on transfer learning.

3.2.1. CBR-based EAs

In [46], Louis and Li presented a learning system that combines a genetic algorithm (GA) solver with a case-base of past problem-solving attempts to increase performance with experience when solving similar traveling salesman problems (TSPs). Borrowing ideas from CBR, Louis and McDonnel [47] proposed the case-injected genetic algorithms (CIGAR). In CIGAR, some appropriate intermediate solutions to similar previously solved problems are periodically injected into GA’s population. CIGAR is used to solve three combinational problems, *i.e.*, combinational circuit design, asset allocation, and job shop scheduling problem (JSSP) [47]. Pérez *et al.* [57] presented a CBR scheme where the design patterns extracted from a GA are reused to reduce convergence times when optimizing combinational logic circuits at the gate level. In [35, 36], based on CBR an improved Bayesian optimization algorithm (BOA) is proposed for biasing BOA, where the previous solutions and the Bayesian network obtained by BOA are stored in the case, and then reused for new problem-solving. The proposed method is mainly used for solving the knapsack problem, TSP, and minimum spanning tree problem [35, 36].

3.2.2. Transfer Learning-based EAs

Besides the CBR-based EAs, very recently, transfer learning is combined with EAs to reuse previous problem-solving knowledge to improve EAs’ performance. Feng *et al.* developed a memetic computational paradigm for search [21, 22] that models how human solves problems. In the proposed approach [21, 22], the knowledge learned from previous problem-solving experiences can be transferred to enhance future evolutionary searches. Two challenging NP-hard routing problems, *i.e.*, capacitated vehicle routing (CVR) and capacitated arc routing (CAR), are used to verify the performance the proposal [21, 22]. In [31, 56], to improve the performance of BOA, the probabilistic models previously obtained by BOA are reused and transferred across problems to speed up the solution of similar problems in the future. Pelikan and Hauschild [55] presented a framework to improve efficiency of model-directed optimization techniques by combining a distance metric with information mined from previous optimization runs on similar problems. In [31, 56, 55], different combinational optimization problems are used to evaluate the performance of the proposed methods. Santana *et al.* [60] proposed a framework for transfer learning between related optimization problems by means of structural transfer. Then, the proposed method is used for the multi-marker tagging single-nucleotide polymorphism selection problem [60]. Hacibeyoglu *et al.* [30] combined transfer learning with artificial bee colony (ABC) algorithm for the numerical optimization problems. In [30], the obtained solutions of previously solved problems are transferred as some solutions in the initial population for solving similar shifted problems. In [33], Iqbal *et al.* presented a genetic programming based learning classifier system, in which the useful building blocks are extracted from smaller problems and then they are reused to learn more complex, large-scale problems in the domain. The proposed method is evaluated on four different Boolean problem domains, *i.e.*, multiplexer, majority-on, carry, and even-parity problems [33].

4. Our Proposal

In this section, we firstly introduce the motivations of this work, followed by the framework of the transfer learning based DE. Based on the proposed framework, an improved JADE method (*i.e.*, transferred adaptive DE, TRADE, in short) is proposed to solve the parameter extraction problems of PEMFC and SOFC models.

4.1. Motivations

As reviewed in Section 3.1, in the literature, some researchers tried to reuse previous problem-solving knowledge to improve the performance of EAs when solving new similar problems. However, most of the work only considered to solve the combinational problems. Although Hacibeyoglu *et al.* [30] proposed the transfer learning based ABC algorithm for the numerical optimization problems, the previously obtained solutions only transferred to solve the same shifted functions. Therefore, the method proposed in [30] may not be used across different problems, especially when the decision variables of the problems have different physical meanings or different dimensions, such as the unknown parameters in PEMFC and SOFC models.

As described in Equations (2) and (4), the parameter extraction problems of PEMFC and SOFC models by the optimization techniques share the similar objective functions. In addition, for the same FC model, if the conditions of the model (such as temperature, hydrogen and oxygen partial pressures, etc.) are different, the extracted parameters are also different. However, for these problems they have the same objective functions. Thus, the algorithmic parameters and operators of the optimization techniques may be reused and transferred to solve the parameter extraction problems of PEMFC and SOFC models.

Since DE is among the most powerful optimization algorithms for numerical problems, based on the above considerations, we will try to combine transfer learning with DE to extract the unknown parameters of PEMFC and SOFC models., where the algorithmic parameters obtained from previous problem-solving experiences will be reused and transferred across different problems.

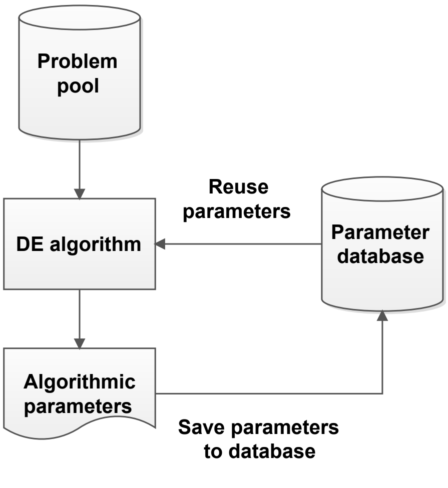


Figure 1: The framework of adaptive DE based on transfer learning

4.2. The Framework

The framework of the proposed adaptive differential evolution based on transfer learning is shown in Figure 1. Firstly, when a new problem in the problem pool will be optimized by the DE algorithm, DE inquires the parameter database. If the database is empty, the default algorithmic parameters (such as CR and F) given by the user will be set in DE to optimize the problem; otherwise, the *Reuse*() procedure will be activated to reuse and transfer the parameters saved in the database as the initial parameters of DE. Then, DE optimizes the problem and updates its parameters according to a specific adaptation technique. When the stopping criterion is met, the *Save*() procedure is activated to save the final parameters into the database.

As shown in Figure 1, the proposed framework has the following advantages:

- *Simplicity*: Compared with the original DE algorithm, only two additional procedures, *i.e.*, *Reuse*() and *Save*(), are used in the framework. These two procedures can be implemented by very simple techniques as shown in Section 4.3.
- *Generality*: The proposed framework is generic, it might be used for different adaptive DE variants, such as jDE [11], SaDE [59], JADE [80], and so on.
- *Interactivity*: In the framework, the expert knowledge can be added into the parameter database. In this way, if we have prior knowledge about the problems, we can provide more accurate parameters for the problems and save them into the database to accelerate the optimization process.

4.3. TRADE: An Improved JADE Approach

Based on the proposed framework as shown in Figure 1, as an illustration, we propose an improved JADE method, referred to as TRADE. In TRADE, the parameter database contains the μ_{CR} and μ_F values obtained from previously solved problems. If the database is empty, both μ_{CR} and μ_F are initialized to be 0.5; otherwise, the *Reuse*() procedure will reuse and transfer the parameters in the database to be the initial μ_{CR} and μ_F.

4.3.1. Parameter Saving: Save()

In TRADE, μ_{CR} and μ_F are saved to the parameter database. In this work, they are appended to the database when the following one of two conditions is satisfied: i) If the best solution in the current population is unchanged in the last continuous 500 generations; or ii) if the stopping criterion of the algorithm is met.

4.3.2. Parameter Reusage: Reuse()

Suppose that there are n items of (μ_{CR}, μ_F) in the parameter database, *i.e.*, $|\mu_{CR}| = |\mu_F| = n$. In order to reuse these knowledge and transfer them to be the initial μ_{CR_0} and μ_{F_0} in TRADE to solve new encountered problems, in this work, we adopt the following simple techniques.

The initial μ_{CR_0} is calculated by

$$\mu_{CR_0} = \text{rndn}(\overline{\mu_{CR}}, \sigma_{\mu_{CR}}) \tag{10}$$

and truncated to $[0, 1]$, where $\overline{\mu_{CR}}$ is the mean value of all μ_{CR} values in the database, $\sigma_{\mu_{CR}}$ is their standard deviation, and $\text{rndn}(\overline{\mu_{CR}}, \sigma_{\mu_{CR}})$ is a random number generator according to the normal distribution.

The initial μ_{F_0} is also calculated in the similar way:

$$\mu_{F_0} = \text{rndn}(\overline{\mu_F}, \sigma_{\mu_F}) \tag{11}$$

and truncated to $[0, 1]$, where $\overline{\mu_F}$ is the mean value of all μ_F values in the database and σ_{μ_F} is their standard deviation.

In TRADE, when a new problem is encountered, the initial μ_{CR_0} and μ_{F_0} are obtained according to Equations (10) and (11), respectively. In this manner, the previous problem-solving knowledge can be reused to bias the values of CR and F in the beginning of the evolution process. Note that if $n = 1$ in the database, in this case, we set $\mu_{CR_0} = \mu_{CR}$ and $\mu_{F_0} = \mu_F$, where μ_{CR} and μ_F are the values saved in the database.

4.3.3. The TRADE Method

The pseudo-code of TRADE is presented in Algorithm 1, where Np is the population size, NFEs is the number of function evaluations, Max_NFEs is the maximal NFEs, c and p are two parameters in JADE [80]. Compared with JADE proposed in [80], the only differences in TRADE are the *Reuse*() and *Save*() procedures as shown in Algorithm 1. Therefore, the proposed TRADE method does not increase the overall complexity of JADE significantly. Although the modifications are minor between TRADE and JADE, TRADE can significantly improves the performance of JADE when solving the parameter extraction problems of PEMFC and SOFC models as revealed in Section 5.

Algorithm 1: The pseudo-code of the TRADE algorithm

```
Input: Control parameters: $Np$ and Max_NFEs
Output: The best final solution
Set $c = 0.1$, $p = 0.05$ as presented in [80];
Initialize the population randomly;
Calculate the objective function value of each solution in the population;
NFEs = $Np$;
if The parameter database is empty then
    Set $\mu_{CR} = 0.5$ and $\mu_F = 0.5$;
else
    Activate the Reuse() procedure as shown in Section 4.3.2 to set the initial $\mu_{CR}$ and $\mu_F$;
while NFEs < Max_NEFs do
    $S_{CR} = \phi, S_F = \phi$;
    Sort the population from the best to the worst based on the objective function value of each solution;
    for $i = 1$ to $Np$ do                                   /* Calculate parameters */
        Generate $CR_i$ and $F_i$ with Equations (5) and (7), respectively;
    for $i = 1$ to $Np$ do                                   /* Generate the trial vector $\mathbf{u}_i$ */
        Produce the trial vector $\mathbf{u}_i$ with "DE/current-to-pbest/1/bin" strategy;
        Apply the boundary constraint-handling to the violated solution;
        Calculate the objective function value of the trial vector $\mathbf{u}_i$;
    for $i = 1$ to $Np$ do                                   /* Survival selection */
        if $\mathbf{u}_i$ is better than its parent $\mathbf{x}_i$ then
            $\mathbf{x}_i = \mathbf{u}_i$;
            $CR_i \rightarrow S_{CR}$;
            $F_i \rightarrow S_F$;
    Update the $\mu_{CR}$ and $\mu_F$ with Equations (6) and (8), respectively;
    NFEs = NFEs + $Np$;
Activate the Save() procedure as shown in Section 4.3.1 to save $\mu_{Cr}$ and $\mu_F$ into the database;
```

5. Results and Analysis

In this section, to evaluate the performance of TRADE, it is applied to extract the unknown parameters of PEMFC and SOFC models as presented in Section 2.1. Totally, eighteen *V-I* datasets are used, containing 7 PEMFC datasets and 11 SOFC datasets, which will be described in Section 5.2. The results of TRADE are compared with those of 9 state-of-the-art EAs, including the real-coded genetic algorithm (rcGA) [32], fast evolutionary programming (FEP) [77], artificial bee colony (ABC) [37], comprehensive learning PSO (CLPSO) [44], jDE [11], SaDE [59], DEGL [18], CoDE [71], and JADE [80].

5.1. Parameter Settings

In this work, the parameter settings of the above-mentioned 10 algorithms are shown in Table 3, unless a changed is mentioned.

All algorithms are coded in standard C++. The maximal number of function evaluations (Max_NFEs) are set to $30,000$ and $150,000$ for PEMFC and SOFC models, respectively. Since all of the ten algorithms are the stochastic algorithms, in order to make the comparison meaningful, each problem is optimized over 100 independent runs[1].

[1] Note that, in TRADE, for each problem only the parameters μ_{CR} and μ_F obtained in the first run are saved in the database.

Table 3: Parameter settings for the ten compared algorithms.

Algorithm	Parameter settings
rcGA [32]	$Np = 100, p_c = 0.9, p_m = 0.15$
FEP [77]	$Np = 100, q = 10$
ABC [37]	$Np = 50, limit = 500$
CLPSO [44]	$Np = 10, m = 7, c = 1.49445$
jDE [11]	$Np = 50, \tau_1 = 0.1, \tau_2 = 0.1$
SaDE [59]	$Np = 50, LP = 50$
DEGL [18]	$Np = 10 \times D, Cr = 0.9, F = 0.8$
CoDE [71]	$Np = 30$
JADE [80]	$Np = 100$ for PEMFC model; $Np = 50$ for SOFC model
TRADE	$Np = 100$ for PEMFC model; $Np = 50$ for SOFC model

Table 4: Parameter Values and Operation Conditions of PEMFC and SOFC Stacks.

Item	Fuel Cell Name	n_{cell}	T (K)	$P^*_{\text{H}_2}$ (atm)	$P^*_{\text{O}_2}$ (atm)	Number of Data Points
P01	WNS-PEMFC [68]	48	313	1.5	1	144
P02	WNS-PEMFC [68]	48	333	1.5	1	147
P03	WNS-PEMFC [68]	48	353	1.5	1	148
P04	Ballard Mark V PEMFC [15]	1	343	1	1	14
P05	SR-12 PEM Generator [15]	48	323	1.47628	0.2095	37
P06	BCS 500-W PEMFC [15]	32	333	1	0.2095	7
P07	Temasek PEMFC [34]	20	323	0.5	0.5	50
P08	WN-SOFC [67]	96	1073	3	3	317
P09	WN-SOFC [67]	96	1173	3	3	317
P10	WN-SOFC [67]	96	1273	3	3	317
P11	ASC-SOFC [20]	1	873	NA	NA	9
P12	ASC-SOFC [20]	1	923	NA	NA	17
P13	ASC-SOFC [20]	1	923	NA	NA	21
P14	ASC-SOFC [20]	1	973	NA	NA	21
P15 - P18	ASC-SOFC [20]	1	923	NA	NA	9

5.2. Data Description

To evaluate the performance of our proposal, seven V-I datasets obtained from different PEMFC models and eleven V-I datasets obtained from different SOFC models are used. The parameter values and operation conditions of PEMFC and SOFC stacks are briefly presented in Table 4. For the seven PEMFC datasets (P01 - P07), P01 - P03 are simulated data generated by the WNS-PEMFC MATLAB/SIMULINK [69], and the other 4 data are experimental data obtained from the literature. For the eleven SOFC datasets (P08 - P11), there are three simulated data generated by WN-SOFC MATLAB/SIMULINK [69] (P08 - P10), and the rest 8 data are experimental data obtained from [20], which are generated by the Elcogen 10 × 10 cm^2 ASC-10B planar single cell under different operation conditions and/or parameter values.

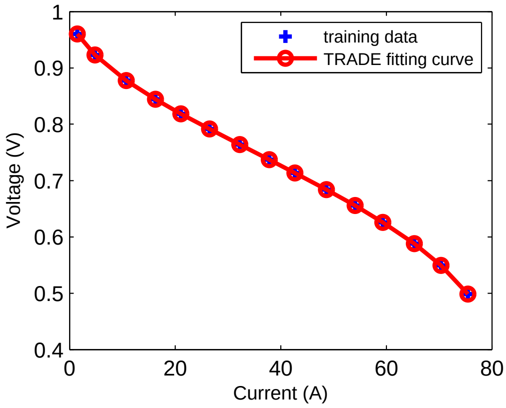


Figure 2: The fitting results of TRADE on the training data.

5.3. *Training of TRADE*

When TRADE is used to solve the first problem, if there is no *prior* knowledge about the problem, the parameter database is empty. In this case, TRADE is the same as JADE [80], and $\mu_{CR} = \mu_F = 0.5$ is set. In this work, since TRADE is used to extract the parameters of PEMFC and SOFC models, TRADE can be trained by using the simple training data to get μ_{CR} and μ_F. Afterwards, the saved parameters are able to be reused in the subsequent experiments. Based on this consideration, we adopt the PEMFC model shown in Equation (1) to generate the training data. The operation conditions are: $T = 343$ K, $P^*_{\mathrm{H}_2} = 1.0$ atm, $P^*_{\mathrm{O}_2} = 1.0$ atm, and $n_{\mathrm{cell}} = 1$. The input parameters $\mathbf{x}^* = \{-0.85596174, 0.0026494826, 4.0359126E-5, -0.00015, 24, 0.0008, 0.072756043, 30, 862.27683, 199.59928, 100\}$. For the training purpose, the training data contains fifteen V-I data points. TRADE is used to extract the parameters from the V-I data points for the PEMFC model. The Max_NFEs is set to be $500,000$ to obtain better values of μ_{CR} and μ_F. After running TRADE on this dataset, the extracted parameters $\mathbf{x} = \{-0.85523964, 0.0034407883, 9.6971264E-5, -0.00015035437, 24, 0.0008, 0.072751056, 29.997848, 862.2696, 199.62503, 100\}$ with $f_{\mathrm{PEMFC}}(\mathbf{x}) = 6.019E-14$. The fitting curve is plotted in Figure 2. The results clearly indicate that TRADE fit the simple training data very well. Finally, the final μ_{CR} and μ_F are saved in the database.

5.4. *Compared with Other EAs*

In this section, TRADE is compared with other state-of-the-art EAs through the 18 datasets for the parameter extraction problems of PEMFC and SOFC models. For all of the compared algorithms, the parameter settings are shown in Table 3.

Table 5: Comparison on the objective function values of all algorithms in all datasets.

	P01	P02	P03	P04	P05	P06
rcGA	1.83E-01 ± 2.6E-02‡	2.29E-01 ± 3.8E-02‡	2.81E-01 ± 3.9E-02‡	8.86E-05 ± 3.0E-05‡	0.357013 ± 1.1E-01‡	0.127354 ± 9.0E-03‡
FEP	1.80E-01 ± 3.7E-02‡	2.07E-01 ± 4.2E-02‡	2.80E-01 ± 5.1E-02‡	7.82E-05 ± 1.5E-05‡	0.361469 ± 1.0E-01‡	0.111526 ± 1.0E-02‡
ABC	2.65E-02 ± 1.9E-02‡	2.60E-02 ± 2.1E-02‡	3.22E-02 ± 2.4E-02‡	5.08E-05 ± 3.8E-06‡	0.153940 ± 3.3E-02‡	0.099237 ± 1.1E-02‡
CLPSO	1.92E-02 ± 1.9E-02‡	2.18E-02 ± 2.4E-02‡	1.66E-02 ± 2.0E-02‡	4.74E-05 ± 1.1E-05‡	0.095182 ± 4.8E-03‡	0.082603 ± 2.7E-04‡
jDE	1.08E-02 ± 5.8E-03‡	8.06E-03 ± 4.6E-03‡	7.26E-03 ± 1.2E-02‡	4.47E-05 ± 5.9E-07‡	0.102931 ± 6.2E-03‡	0.083856 ± 6.1E-04‡
SaDE	6.93E-05 ± 1.2E-04‡	2.96E-05 ± 5.7E-05‡	1.58E-05 ± 3.4E-05‡	4.29E-05 ± 1.7E-07‡	0.088684 ± 1.8E-03‡	0.081871 ± 2.1E-05‡
DEGL	*3.34E-05 ± 3.5E-05*‡	*1.70E-05 ± 1.3E-05*‡	*5.82E-06 ± 4.8E-06*‡	*4.28E-05 ± 6.9E-08*‡	*0.087195 ± 2.8E-04*‡	*0.081864 ± 1.5E-05*‡
CoDE	1.50E-04 ± 2.5E-04‡	9.16E-05 ± 1.8E-04‡	5.73E-05 ± 1.5E-04‡	4.31E-05 ± 2.4E-07‡	0.089896 ± 2.8E-03‡	0.081964 ± 6.0E-05‡
JADE	4.71E-04 ± 3.0E-03‡	4.38E-04 ± 3.6E-03‡	1.17E-04 ± 9.1E-04‡	4.28E-05 ± 2.8E-07‡	0.088036 ± 3.1E-03‡	0.081873 ± 9.0E-05‡
TRADE	**2.74E-05 ± 2.7E-05**	**1.34E-05 ± 8.8E-06**	**4.43E-06 ± 1.3E-06**	**4.28E-05 ± 2.7E-11**	**0.087126 ± 7.0E-06**	**0.081862 ± 8.7E-09**

	P07	P08	P09	P10	P11	P12
rcGA	2.05E-01 ± 1.2E-01‡	6.60E+03 ± 1.0E+04‡	3.17E+03 ± 2.3E+03‡	2.24E+03 ± 2.3E+03‡	2.68E-02 ± 5.0E-02‡	1.04E-01 ± 1.0E-01‡
FEP	3.37E-02 ± 9.8E-03‡	6.51E-01 ± 2.2E+00‡	4.78E-01 ± 4.4E-01‡	5.64E-01 ± 6.3E-01‡	1.16E-04 ± 9.4E-05‡	5.94E-05 ± 9.8E-05‡
ABC	5.70E-03 ± 3.7E-03‡	9.37E-01 ± 7.0E-01‡	9.79E-01 ± 6.6E-01‡	1.02E+00 ± 7.9E-01‡	7.26E-05 ± 4.5E-05‡	4.40E-05 ± 3.1E-05‡
CLPSO	2.73E-03 ± 4.0E-03‡	4.89E+02 ± 4.9E+03‡	5.28E+02 ± 5.3E+03‡	5.50E+02 ± 5.5E+03‡	1.02E-05 ± 3.6E-05‡	1.26E-05 ± 6.2E-05‡
jDE	1.24E-03 ± 3.1E-04‡	1.33E-01 ± 6.6E-02‡	1.16E-01 ± 5.1E-02‡	9.75E-02 ± 4.3E-02‡	7.45E-06 ± 3.9E-06‡	3.55E-06 ± 1.8E-06‡
SaDE	6.25E-04 ± 2.4E-05‡	2.73E-03 ± 3.6E-03‡	1.50E-03 ± 3.1E-03‡	3.05E-04 ± 7.9E-04‡	*4.05E-06 ± 2.7E-08*‡	1.45E-06 ± 1.5E-08‡
DEGL	*6.01E-04 ± 8.2E-06*‡	1.19E-02 ± 2.1E-02‡	6.88E-03 ± 1.7E-02‡	5.04E-03 ± 1.4E-02‡	4.53E-06 ± 4.1E-06‡	1.45E-06 ± 3.7E-08‡
CoDE	6.43E-04 ± 3.0E-05‡	1.15E-02 ± 1.1E-02‡	7.96E-03 ± 6.9E-03‡	5.08E-03 ± 5.2E-03‡	4.34E-06 ± 2.1E-07‡	1.61E-06 ± 1.4E-07‡
JADE	6.17E-04 ± 6.1E-05‡	*1.70E-03 ± 3.4E-03*‡	*6.24E-04 ± 3.8E-03*‡	*2.55E-04 ± 1.9E-03*‡	4.05E-06 ± 5.1E-08‡	*1.44E-06 ± 1.4E-08*‡
TRADE	**5.97E-04 ± 1.7E-06**	**1.35E-03 ± 2.4E-14**	**1.67E-04 ± 4.9E-14**	**4.85E-05 ± 3.3E-12**	**4.04E-06 ± 3.2E-13**	**1.44E-06 ± 1.0E-15**

	P13	P14	P15	P16	P17	P18
rcGA	1.46E-01 ± 1.8E-01‡	1.29E-01 ± 1.8E-01‡	3.54E-02 ± 2.7E-02‡	2.54E-02 ± 2.7E-02‡	2.30E-02 ± 2.5E-02‡	2.47E-02 ± 2.7E-02‡
FEP	3.57E-05 ± 4.0E-05‡	2.24E-05 ± 8.2E-05‡	2.91E-04 ± 2.0E-03‡	6.22E-06 ± 4.2E-05‡	1.49E-05 ± 1.2E-04‡	2.41E-06 ± 2.3E-06‡
ABC	5.19E-05 ± 2.6E-05‡	2.65E-05 ± 1.8E-05‡	5.46E-05 ± 3.6E-05‡	4.50E-06 ± 4.0E-06‡	5.50E-06 ± 7.6E-06‡	7.10E-06 ± 6.0E-06‡
CLPSO	3.23E-05 ± 8.9E-05‡	2.32E-05 ± 1.8E-04‡	8.73E-06 ± 4.0E-05‡	6.14E-07 ± 1.3E-07‡	1.73E-07 ± 5.8E-08‡	1.49E-06 ± 1.7E-07‡
jDE	6.07E-06 ± 2.6E-06‡	2.97E-06 ± 2.0E-07‡	3.93E-06 ± 3.4E-06‡	6.01E-07 ± 1.5E-08‡	1.81E-07 ± 1.9E-08‡	1.52E-06 ± 4.4E-08‡
SaDE	2.42E-06 ± 6.2E-08‡	2.53E-06 ± 1.7E-08‡	1.07E-06 ± 1.5E-08‡	*5.75E-07 ± 1.8E-09*‡	1.55E-07 ± 1.3E-09‡	1.42E-06 ± 1.8E-08‡
DEGL	2.43E-06 ± 2.2E-07‡	2.52E-06 ± 3.9E-08‡	2.67E-06 ± 1.1E-05‡	5.75E-07 ± 5.8E-09‡	*1.53E-07 ± 7.9E-10*‡	*1.40E-06 ± 1.8E-08*‡
CoDE	2.59E-06 ± 1.6E-07‡	2.63E-06 ± 7.8E-08‡	1.18E-06 ± 6.0E-08‡	5.83E-07 ± 3.9E-09‡	1.58E-07 ± 3.5E-09‡	1.45E-06 ± 2.7E-08‡
JADE	*2.38E-06 ± 2.4E-08*‡	*2.51E-06 ± 4.0E-09*‡	*1.06E-06 ± 5.5E-09*‡	5.76E-07 ± 4.7E-09‡	1.54E-07 ± 2.0E-09‡	*1.40E-06 ± 1.8E-08*‡
TRADE	**2.34E-06 ± 6.4E-08**	**2.51E-06 ± 6.0E-15**	**1.06E-06 ± 2.9E-20**	**5.73E-07 ± 4.3E-10**	**1.52E-07 ± 6.0E-10**	**1.38E-06 ± 6.5E-10**

"‡" indicates TRADE is significantly better than its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$.

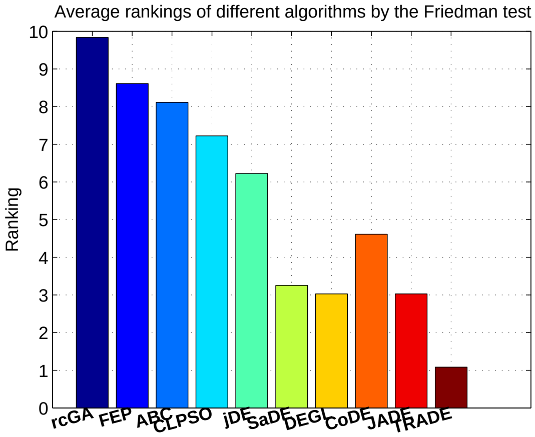


Figure 3: The average rankings of all algorithms by the Friedman test.

5.4.1. *On the Quality of Final Solutions*

The quality of final solutions of all algorithms are reported in Table 5, where the overall best and the second best results are highlighted in **boldface** and *italic*, respectively. All results are averaged over 100 runs. In Table 5, the mean and standard deviation values are provided. To make the comparison statistically meaningful, the Wilcoxon test is used to compare TRADE with other EAs. In Table 5, "‡" indicates TRADE is significantly better than its competitor according to the Wilcoxon signed-rank test at $\alpha = 0.05$. Moreover, based on the mean values of all datasets, the Friedman test[2] is used to calculate the average rankings of all algorithms. The results are given in Figure 3.

According to the results shown in Table 5 and Figure 3, it can be clearly observed that:

- TRADE consistently gets the best results in all problems in terms of the mean objective function values. It significantly outperforms other 9 algorithms based on the Wilcoxon test at $\alpha = 0.05$ in all problems. In addition, TRADE is the most robust algorithm according to the standard deviation values. Figure 3 also reveals that TRADE obtains the first average ranking among all algorithm by the Friedman test.

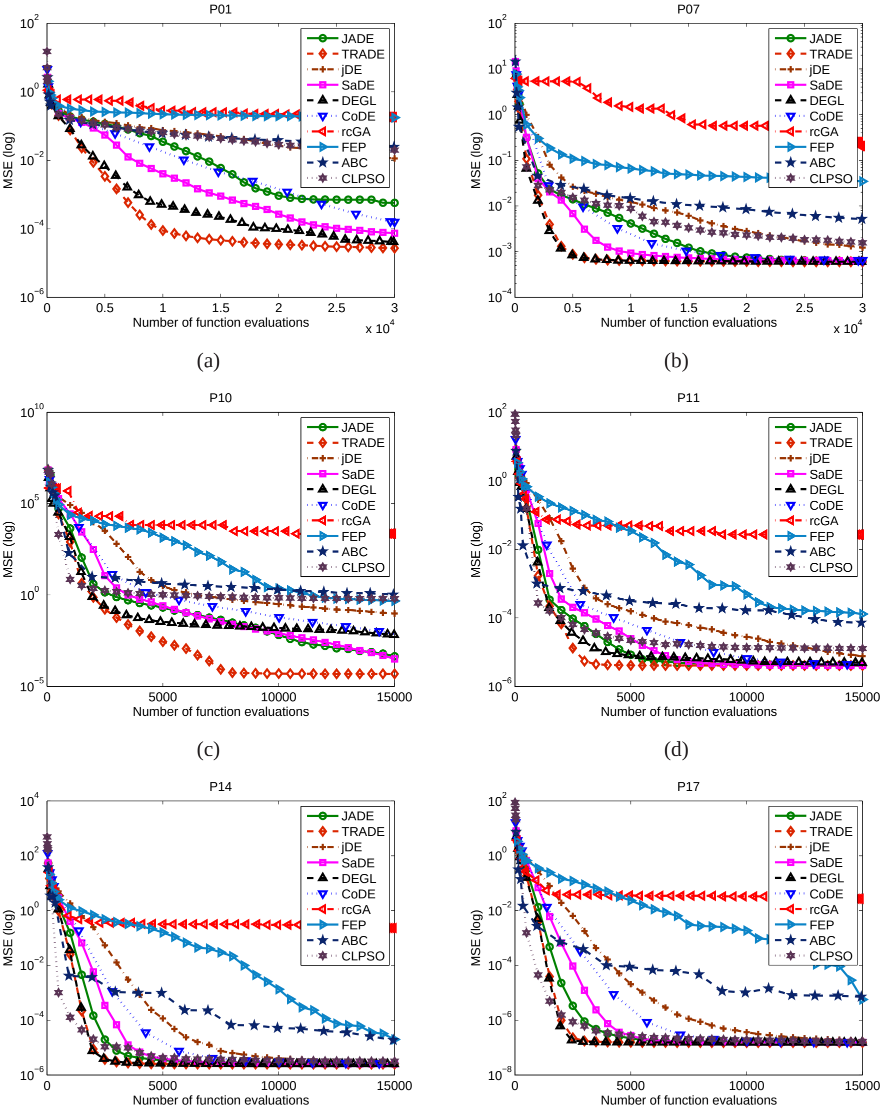
- Compared the results between TRADE and JADE, TRADE can improve the performance of JADE significantly in all problems. Even in the first dataset P01, there only contains one item of (μ_{CR}, μ_F) in the database, TRADE is still able to provide significantly better results than those of JADE. This phenomenon indicates the benefits of properly reusing previous parameters for extracting the parameters of FC models effectively.

It is worth pointing out that although the objective function values of SaDE, DEGL and JADE are very close to those of TRADE in some problems, however, since the simulated data and experimental data are generated from the different FC models, no information is available about the accurate values of the parameters of them; 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.

5.4.2. *On the Convergence Speed*

In this section, the convergence curves of all algorithms are compared and plotted in Figure 4. For the sake of brevity, only some representative problems are chosen. From Figure 4, we can see that TRADE is able to consistently converge to the approximate optimal solutions in all problems. It obtains faster convergence speed than its predecessor JADE. In addition, compared with other EAs, TRADE also provides the fastest convergence speed in overall. By carefully looking at the results, we find that: i) For the PEMFC model, DEGL converges fastest in the beginning of the evolution process, however, it converges slowly after NFEs $= 5,000$. ii) For the SOFC model, both ABC and CLPSO converge faster than TRADE in early evolution process, however, they stagnate quickly and get the poor quality of the final solutions as shown in Table 5.

[2] The results of the Wilcoxon and Friedman tests are calculated by the KEEL software [1].



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: Comparison on the convergence speed among different EAs for the selected problems.

5.5. *On the V-I Characteristics*

In this work, TRADE is used to extract the unknown parameters of PEMFC and SOFC models. Therefore, to further evaluate the performance of TRADE, it is important to check the *V*-*I* characteristics obtained by TRADE. To achieve *V*-*I* characteristics, the optimal parameter values extracted by TRADE are fed back to the PEMFC and SOFC mathematical models as shown in Equations (1) and (3), respectively. Figure 5 provides comparisons between the data of different FC models and the data obtained by TRADE for all problems. As shown in Figure 5, it is clear that the *V*-*I* characteristics obtained by TRADE are highly coincide with the data of FC models in all cases. Thus, we can conclude that TRADE can be an effective and efficient alternative for the parameter extraction problems of PEMFC and SOFC models.

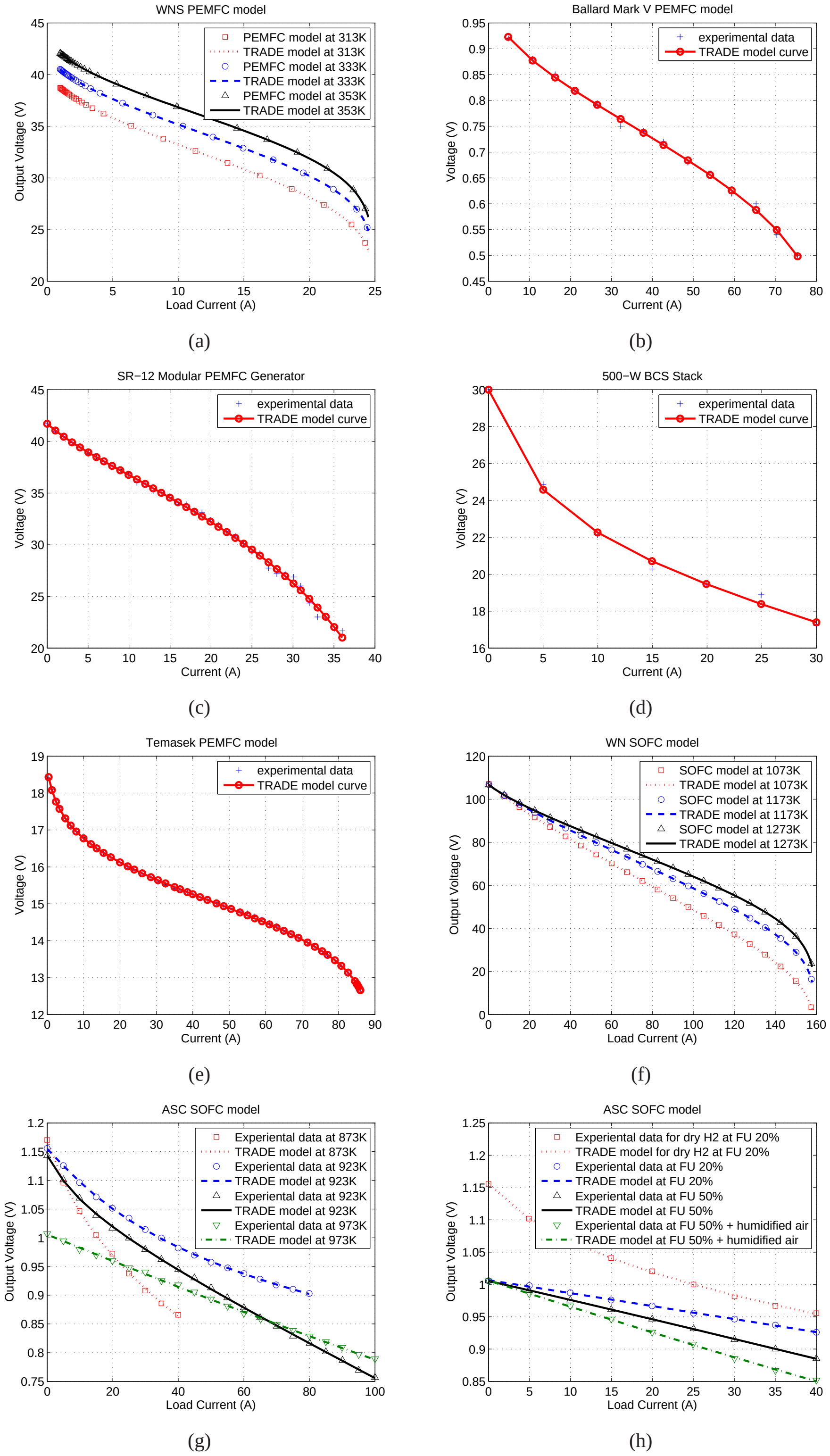


Figure 5: Comparisons between the data of different FC models and the data obtained by TRADE for all problems. (a): P01 - P03; (b)-(e): P04 - P07; (f): P08 - P10; (g): P11 - P14; (h): P15 - P18.

Table 6: Comparison on the reported results in [3] with those of TRADE for the Ballard Mark V PEMFC model.

Parameter	BMO	BSA	BSABCM-1	BSABCM-2	BSABCM-3	TRADE
A	85.52841	85.66426	90.41865	78.62501	63.67867	100
ℓ (μm)	162.36927	200	132.9658	186.9986	176.17291	199.59871
R_c (Ω)	0.000491	0.0008	0.000194	0.0008	0.000735	0.0008
ξ_1	-1.081107	-1.16954	-0.98686	-0.91636	-1.09537	-1.196411
ξ_2	0.003311	0.003612	0.003395	0.003497	0.003623	0.004153
ξ_3	4.239E-05	5.671E-05	6.879E-05	9.665E-05	6.927E-05	7.684E-05
ξ_4	-1.469E-04	-9.540E-05	-1.476E-04	-1.193E-04	-1.192E-04	-1.504E-04
J_n (mA/cm^2)	29.99945	1.1147	2.6467	0.93234	4.07185	30
$J_{\max}$ (mA/cm^2)	1020.70957	872.68143	847.72451	953.4404	1170.406	862.27594
λ	21.71831	13.92213	10.01962	17.07371	19.54295	24
B (V)	0.08265	0.020932	0.025832	0.027605	0.024743	0.072756
$f_{\text{PEMFC}}(\mathbf{x})$	9.2823E-05	8.4986E-05	8.6210E-05	*8.3398E-05*	8.4455E-05	**4.2751E-05**

Table 7: Comparison on the reported results in [3] with those of TRADE for the SR-12 modular PEM generator.

Parameter	BMO	BSA	BSABCM-1	BSABCM-2	BSABCM-3	TRADE
A	97.07073	100	100	81.157156	100	99.999988
ℓ (μm)	20.0004	20	23.1476	34.7919	20	20.000014
R_c (Ω)	0.000107	0.000214	0.0001	0.0001	0.0001	0.0001
ξ_1	-1.181004	-0.8532	-1.182389	-0.987138	-0.85712	-0.937314
ξ_2	0.003986	0.002777	0.004269	0.00301	0.002584	0.003465
ξ_3	7.805E-05	6.493E-05	9.582E-05	5.475E-05	5.179E-05	9.308E-05
ξ_4	-9.540E-05	-9.540E-05	-9.540E-05	-9.540E-05	-9.540E-05	-9.540E-05
J_n (mA/cm^2)	29.99993	30	30	30	30	29.999998
$J_{\max}$ (mA/cm^2)	510.66316	500	500	599.33339	503.1227	500.45661
λ	23.99247	24	24	24	20.81857	23.999991
B (V)	0.23399	0.24	0.237949	0.22379	0.241591	0.237502
$f_{\text{PEMFC}}(\mathbf{x})$	0.088359	0.10572	*0.087957*	0.089777	0.089114	**0.087124**

5.6. *Compared with Reported Results*

In the previous sections, TRADE is used to extract the unknown parameters for both PEMFC and SOFC models under different conditions. Additionally, TRADE is also compared with other advanced EAs directly. In this section, to further understand the performance of TRADE, it is compared with the reported results presented in [3], because both of them use the same PEMFC model with 11 unknown parameters. In [3], two datasets in Table 4 (*i.e.*, P04 and P05) are used, and the extracted parameters by BMO, BSA, BSABCM-1, BSABCM-2, and BSABCM-3 are reported. The extracted parameters and their corresponding objective function values are described in Table 6 and Table 7, respectively. Note that, to make a fair comparison, the objective function values of different algorithms in [3] are re-calculated by using the parameters in Tables 6 and 7. The reason is that the data in P04 and P05 originates from the graphical diagrams in [49], this procedure may lead to some extra variability. The results in Tables 6 and 7 clearly reveal that TRADE is able to provide the best objective function values for both P04 and P05 compared with other five methods in [3].

6. Conclusions and Future Work

In general, no problems exist in isolation, which always share some similar features. Based on this consideration, in this paper, we propose a transferred adaptive differential evolution framework, where the previous problem-solving parameters of DE can be reused and transferred to solve new similar problems. According to the framework, we present an improved JADE method, in which the parameters μ_{CR} and μ_F are transferred across different problems. The modifications in TRADE are simple and minor compared with its predecessor JADE. Afterwards, the proposed TRADE method is used to solve the parameter extraction problems of two different fuel cell models, *i.e.*, PEMFC and SOFC models. Eighteen V-I datasets generated by different PEMFC and SOFC models are used to evaluate the performance of TRADE, and the experimental results indicate that

- By transferring the previous problem-solving parameters saved in the database, TRADE yields significantly better results than JADE in terms of the quality of final solutions and the convergence speed in all problems.

- Compared with other state-of-the-art EAs, TRADE also provides very promising results. It is able to obtain the first average ranking among all compared methods.

- The V-I characteristics obtained by TRADE fit both the simulated data and experimental data pretty well in all cases. Therefore, it may be used to solve other complex optimization problems of fuel cell models.

Even with minor modifications on JADE, TRADE is capable of providing very promising results when solving similar problems. This motivates us to investigate more generic improved EAs based on transfer learning for the numerical optimization problems in the near future.

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

References

[1] Alcalá-Fdez, J., Sánchez, L., García, S., 2012. KEEL: A software tool to assess evolutionary algorithms to data mining problems.
URL http://www.keel.es/

[2] Ang, S. M. C., Fraga, E. S., Brandon, N. P., Samsatli, N. J., Brett, D. J., 2011. Fuel cell systems optimisation - methods and strategies. International Journal of Hydrogen Energy 36 (22), 14678 – 14703.

[3] Askarzadeh, A., dos Santos Coelho, L., 2014. A backtracking search algorithm combined with burger's chaotic map for parameter estimation of PEMFC electrochemical model. International Journal of Hydrogen Energy 39 (21), 11165 – 11174.

[4] Askarzadeh, A., Rezazadeh, A., 2011. Artificial immune system-based parameter extraction of proton exchange membrane fuel cell. International Journal of Electrical Power & Energy Systems 33 (4), 933 – 938.

[5] Askarzadeh, A., Rezazadeh, A., 2011. A grouping-based global harmony search algorithm for modeling of proton exchange membrane fuel cell. International Journal of Hydrogen Energy 36 (8), 5047 – 5053.

[6] Askarzadeh, A., Rezazadeh, A., 2011. A new artificial bee swarm algorithm for optimization of proton exchange membrane fuel cell model parameters. Journal of Zhejiang University SCIENCE C 12, 638–646.

[7] Askarzadeh, A., Rezazadeh, A., 2011. Optimization of PEMFC model parameters with a modified particle swarm optimization. International Journal of Energy Research 35 (14), 1258–1265.

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

[9] Askarzadeh, A., Rezazadeh, A., 2013. A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: Bird mating optimizer. International Journal of Energy Research 37 (10), 1196–1204.

[10] Bozorgmehri, S., Hamedi, M., 2012. Modeling and optimization of anode-supported solid oxide fuel cells on cell parameters via artificial neural network and genetic algorithm. Fuel Cells 12 (1), 11–23.

[11] Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V., Dec 2006. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Trans. on Evol. Comput. 10 (6), 646–657.

[12] Carton, J., Lawlor, V., Olabi, A., Hochenauer, C., Zauner, G., 2012. Water droplet accumulation and motion in PEM (proton exchange membrane) fuel cell mini-channels. Energy 39 (1), 63 – 73.

[13] Chakraborty, U. K., Abbott, T. E., Das, S. K., 2012. PEM fuel cell modeling using differential evolution. Energy 40 (1), 387 – 399.

[14] Chan, S., Khor, K., Xia, Z., 2001. A complete polarization model of a solid oxide fuel cell and its sensitivity to the change of cell component thickness. Journal of Power Sources 93 (1 - 2), 130 – 140.

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

[16] Corrêa, J., Farret, F., Popov, V., Simoes, M., march 2005. Sensitivity analysis of the modeling parameters used in simulation of proton exchange membrane fuel cells. IEEE Transactions on Energy Conversion 20 (1), 211 – 218.

[17] Dai, C., Chen, W., Cheng, Z., Li, Q., Jiang, Z., Jia, J., 2011. 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), 369 – 376.

[18] Das, S., Abraham, A., Chakraborty, U. K., Konar, A., 2009. Differential evolution using a neighborhood-based mutation operator. IEEE Trans. on Evol. Comput. 13 (3), 526–553.

[19] Das, S., Suganthan, P. N., 2011. Differential evolution: A survey of the state-of-the-art. IEEE Trans. on Evol. Comput. 15 (1), 4–31.

[20] Elcogen, 2013. SOFC fuel cell technical data.
URL http://www.elcogen.com/single-cell-technical-data

[21] Feng, L., Ong, Y.-S., Tan, A.-H., Tsang, I. W.-H., 2012. Meme as building block for evolutionary optimization of problem instances. CoRR abs/1207.0702.

[22] Feng, L., Ong, Y.-S., Tsang, I. W.-H., Tan, A.-H., 2012. An evolutionary search paradigm that learns with past experiences. In: WCCI 2012 IEEE World Congress on Computational Intelligence, IEEE CEC. pp. 2708–2715.

[23] Fong, K., Lee, C., Chow, C., Yuen, S., 2011. Simulation-optimization of solar-thermal refrigeration systems for office use in subtropical Hong Kong. Energy 36 (11), 6298 – 6307.

[24] Gämperle, R., Müler, S., Koumoutsakos, P., 2002. A parameter study for differential evolution. In: Proc. WSEAS Int. Conf. Advances Intell. Syst., Fuzzy Syst., Evol. Comput. pp. 293–298.

[25] Gong, W., Cai, Z., 2013. Accelerating parameter identification of proton exchange membrane fuel cell model with ranking-based differential evolution. Energy 59, 356 – 364.

[26] Gong, W., Cai, Z., 2013. Differential evolution with ranking-based mutation operators. IEEE Transactions on Cybernetics 43 (6), 2066 – 2081.

[27] Gong, W., Cai, Z., 2014. Parameter optimization of PEMFC model with improved multi-strategy adaptive differential evolution. Engineering Applications of Artificial Intelligence 27, 28 – 40.

[28] Gong, W., Cai, Z., Ling, C. X., Li, H., 2011. Enhanced differential evolution with adaptive strategies for numerical optimization. IEEE Transactions on Systems, Man, and Cybernetics: Part B – Cybernetics 41 (2), 397–413.

[29] Gong, W., Cai, Z., Yang, J., Li, X., Jian, J., 2014. Parameter identification of an SOFC model with an efficient, adaptive differential evolution algorithm. International Journal of Hydrogen Energy 39 (10), 5086–5096.

[30] Hacibeyoglu, M., Kocr, B., Arslan, A., 2012. Transfer learning for artificial bee colony algorithm to optimize numerical functions. In: International Conference on Computer Engineering and Network Security. pp. 271–274.

[31] Hauschild, M. W., Pelikan, M., Sastry, K., Goldberg, D. E., 2012. Using previous models to bias structural learning in the hierarchical BOA. Evolutionary Computation 20 (1), 135–160.

[32] Herrera, F., Lozano, M., Verdegay, J., 1998. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. Artificial Intelligence Review 12 (4), 265–319.

[33] Iqbal, M., Browne, W., Zhang, M., 2014. Reusing building blocks of extracted knowledge to solve complex, large-scale boolean problems. IEEE Trans. on Evol. Comput., 1–16To appear.

[34] Jia, J., Wang, Y., Li, Q., Cham, Y., Han, M., march 2009. Modeling and dynamic characteristic simulation of a proton exchange membrane fuel cell. IEEE Transactions on Energy Conversion 24 (1), 283 – 291.

[35] Kaedi, M., Ghasem-Aghaee, N., 2011. Biasing bayesian optimization algorithm using case based reasoning. Knowledge-Based Systems 24, 1245–1253.

[36] Kaedi, M., Ghasem-Aghaee, N., 2012. Improving case-based reasoning in solving optimization problems using Bayesian optimization algorithm. Intelligent Data Analysis 16 (2), 199–210.

[37] Karaboga, D., Basturk, B., Nov 2007. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. J. of Global Optimization 39 (3), 459–471.

[38] Karimi, M., Askarzadeh, A., Rezazadeh, A., 2012. Using tournament selection approach to improve harmony search algorithm for modeling of proton exchange membrane fuel cell. Int. J. Electrochem. Sci. 7, 6426 – 6435.

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

[40] Lawlor, V., 2013. Review of the micro-tubular solid oxide fuel cell (Part II: Cell design issues and research activities). Journal of Power Sources 240 (0), 421 – 441.

[41] Lawlor, V., Griesser, S., Buchinger, G., Olabi, A., Cordiner, S., Meissner, D., 2009. Review of the micro-tubular solid oxide fuel cell: Part I. Stack design issues and research activities. Journal of Power Sources 193 (2), 387 – 399.

[42] Li, Q., Chen, W., Wang, Y., Liu, S., Jia, J., June 2011. Parameter identification for PEM fuel-cell mechanism model based on effective informed adaptive particle swarm optimization. IEEE Transactions on Industrial Electronics 58 (6), 2410 – 2419.

[43] Li, Y., Shen, J., Lu, J., 2011. Constrained model predictive control of a solid oxide fuel cell based on genetic optimization. Journal of Power Sources 196 (14), 5873 – 5880.

[44] Liang, J. J., Qin, A. K., Suganthan, P. N., Baskar, S., 2006. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans. on Evol. Comput. 10 (3), 281–295.

[45] Liu, J., Lampinen, J., 2005. A fuzzy adaptive differential evolution algorithm. Soft Comput. 9 (6), 448–462.

[46] Louis, S. J., Li, G., 2000. Case injected genetic algorithms for traveling salesman problems. Information Sciences 122 (2-4), 201 – 225.

[47] Louis, S. J., McDonnell, J., 2004. Learning with case-injected genetic algorithms. IEEE Trans. on Evol. Comput. 8 (4), 316–328.

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

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

[50] Ohenoja, M., Leiviskä, K., 2010. Validation of genetic algorithm results in a fuel cell model. International Journal of Hydrogen Energy 35 (22), 12618 – 12625.

[51] Outeiro, M., Chibante, R., Carvalho, A., de Almeida, A., 2008. A parameter optimized model of a proton exchange membrane fuel cell including temperature effects. Journal of Power Sources 185 (2), 952 – 960.

[52] Outeiro, M., Chibante, R., Carvalho, A., de Almeida, A., 2009. A new parameter extraction method for accurate modeling of PEM fuel cells. International Journal of Energy Research 33 (11), 978–988.

[53] Pan, S. J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22 (10), 1345–1359.

[54] Peighambardoust, S., Rowshanzamir, S., Amjadi, M., 2010. Review of the proton exchange membranes for fuel cell applications. International Journal of Hydrogen Energy 35 (17), 9349 – 9384.

[55] Pelikan, M., Hauschild, M. W., 2012. Learn from the past: Improving model-directed optimization by transfer learning based on distance-based bias. Tech. Rep. MEDAL Report 2012007, Missouri Estimation of Distribution Algorithms Laboratory.

[56] Pelikan, M., Hauschild, M. W., Lanzi, P. L., 2012. Transfer learning, soft distance-based bias, and the hierarchical BOA. In: Coello, C. C. (Ed.), PPSN 2012. Vol. LNCS 7491. pp. 173–183.

[57] Pérez, E. I., Coello, C. A. C., Aguirre, A. H., 2005. Extraction and reuse of design patterns from genetic algorithms using case-based reasoning. Soft Computing 9 (1), 44–53.

[58] Price, K., Storn, R., Lampinen, J., 2005. Differential Evolution: A Practical Approach to Global Optimization. Springer-Verlag, Berlin.

[59] Qin, A. K., Huang, V. L., Suganthan, P. N., Apr 2009. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. on Evol. Comput. 13 (2), 398–417.

[60] Santana, R., Mendiburu, A., Lozano, J. A., 2012. Structural transfer using EDAs: An application to multi-marker tagging SNP selection. In: WCCI 2012 IEEE World Congress on Computational Intelligence, IEEE CEC. pp. 1–8.

[61] Schluckner, C., Subotić, V., Lawlor, V., Hochenauer, C., 2014. Three-dimensional numerical and experimental investigation of an industrial-sized SOFC fueled by diesel reformat - Part I: Creation of a base model for further carbon deposition modeling. International Journal of Hydrogen Energy 39 (33), 19102 – 19118.

[62] Secanell, M., Wishart, J., Dobson, P., 2011. Computational design and optimization of fuel cells and fuel cell systems: A review. Journal of Power Sources 196 (8), 3690 – 3704.

[63] Shi, J., Xue, X., 2012. Inverse estimation of electrode microstructure distributions in NASA Bi-electrode supported solid oxide fuel cells. Chemical Engineering Journal 181-182 (0), 607 – 613.

[64] Storn, R., Price, K., 1995. Differential evolution–A simple and efficient adaptive scheme for global optimization over continuous spaces. Tech. rep., Berkeley, CA, tech. Rep. TR-95-012.

[65] Storn, R., Price, K., Dec 1997. Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces. J. of Global Optim. 11 (4), 341–359.

[66] Vaisakh, K., Srinivas, L., 2010. A genetic evolving ant direction DE for OPF with non-smooth cost functions and statistical analysis. Energy 35 (8), 3155 – 3171.

[67] Wang, C., Nehrir, M., Dec. 2007. A physically based dynamic model for solid oxide fuel cells. IEEE Transactions on Energy Conversion 22 (4), 887–897.

[68] Wang, C., Nehrir, M., Shaw, S., 2005. Dynamic models and model validation for PEM fuel cells using electrical circuits. IEEE Transactions on Energy Conversion 20 (2), 442 – 451.

[69] Wang, C., Nehrir, M. H., 2013. Dynamic models for PEMFs and tubular SOFCs.
URL `http://www.coe.montana.edu/ee/fuelcell/`

[70] Wang, K., Hissel, D., Pra, M., Steiner, N., Marra, D., Sorrentino, M., Pianese, C., Monteverde, M., Cardone, P., Saarinen, J., 2011. A review on solid oxide fuel cell models. International Journal of Hydrogen Energy 36 (12), 7212 – 7228.

[71] Wang, Y., Cai, Z., Zhang, Q., 2011. Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans. on Evol. Comput. 15 (1), 55–66.

[72] Wang, Y., Zhou, J., Mo, L., Zhang, R., Zhang, Y., 2012. Short-term hydrothermal generation scheduling using differential real-coded quantum-inspired evolutionary algorithm. Energy 44 (1), 657 – 671.

[73] Wu, X., Zhu, X., Cao, G., Tu, H., 2007. Modeling a SOFC stack based on GA-RBF neural networks identification. Journal of Power Sources 167 (1), 145 – 150.

[74] Wu, X., Zhu, X., Cao, G., Tu, H., 2008. Predictive control of SOFC based on a GA-RBF neural network model. Journal of Power Sources 179 (1), 232 – 239.

[75] Yang, J., Li, X., Jiang, J. H., Jian, L., Zhao, L., Jiang, J. G., Wu, X. G., Xu, L. H., 2011. Parameter optimization for tubular solid oxide fuel cell stack based on the dynamic model and an improved genetic algorithm. International Journal of Hydrogen Energy 36 (10), 6160 – 6174.

[76] Yang, S., Wang, N., 2012. A novel P systems based optimization algorithm for parameter estimation of proton exchange membrane fuel cell model. International Journal of Hydrogen Energy 37 (10), 8465 – 8476.

[77] Yao, X., Liu, Y., Lin, G., Jul 1999. Evolutionary programming made faster. IEEE Trans. on Evol. Comput. 3 (2), 82–102.

[78] Ye, M., Wang, X., Xu, Y., 2009. Parameter identification for proton exchange membrane fuel cell model using particle swarm optimization. International Journal of Hydrogen Energy 34 (2), 981 – 989.

[79] Zhang, H., Zhou, J., Fang, N., Zhang, R., Zhang, Y., 2013. Daily hydrothermal scheduling with economic emission using simulated annealing technique based multi-objective cultural differential evolution approach. Energy 50 (1), 24 – 37.

[80] Zhang, J., Sanderson, A. C., Oct 2009. JADE: Adaptive differential evolution with optional external archive. IEEE Trans. on Evol. Comput. 13 (5), 945–958.

[81] Zhang, L., Wang, N., 2013. An adaptive RNA genetic algorithm for modeling of proton exchange membrane fuel cells. International Journal of Hydrogen Energy 38 (1), 219 – 228.