Expert Systems With Applications 200 (2022) 117192

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Cooperative co-evolutionary differential evolution algorithm applied for parameters identification of lithium-ion batteries

Chuan Wang [*], Minyi Xu, Qinjin Zhang, Ruizheng Jiang, Jinhong Feng, Yi Wei, Yancheng Liu

Marine Engineering College, Dalian Maritime University, 1 Linghai Road, Dalian, Liaoning Province, People's Republic of China

ARTICLE INFO

Keywords:
Large scale optimization problem
Differential evolution
Cooperative co-evolution algorithm
Parameters identification of battery

ABSTRACT

Parameters identification of battery is a significant task for lithium-ion batteries. Some widely used techniques usually simplify the electrical circuit model (ECM) with non-linearity to a linear model or local linear model. However, by using such a methodology, the parameters in ECMs are not globally optimal, since the parameters may be not consistent at different linearized points. To address this issue, this paper proposed a cooperative co-evolution differential evolution (CCDE) algorithm to identify parameters of lithium-ion battery, without any linearization or pre-assumption. First, to describe the dynamic behaviors of battery, we presented a first-order RC equivalent circuit model ECM. Without making any approximation, improved Euler's numerical method was utilized to solve the differential equations directly. Second, an optimizing objective function was built to minimize errors between the true and optimized terminal voltages. In that optimization model, parameters of battery (R_0, R_P and C_P) and $v_{OCV}(t)$ at each sampling point were considered as variables to be optimized, resulting in a very high-dimension problem. Third, such an optimization problem was transformed into a large scale optimization problem (LSOP). Based on the character of parameters identification, we proposed a new *m*-decomposition method which is different from general grouping methods for benchmark functions and its corresponding differential evolution (DE) algorithm to solve this LSOP. Comprehensive experimental results demonstrated effectiveness of the proposed framework and methodology, compared with seven state-of-the-art cooperative co-evolution methods.

1. Introduction

Battery is the most important energy storage device in the current world. Many significant engineering domains, such as electric vehicle (EV), smart phone and power grid, need batteries as their dominating energy storage device (Kwak, Lkhagvasuren, Park, & You 2020). Within various categories of batteries, lithium-ion battery (LIB) is the most competitive and promising one due to high efficiency, long cycle life and high energy density (He, Xiong, & Peng, 2016; Lu, Han, Li, Hua, & Ouyang, 2013;Xing, Ma, Tsui, & Pecht, 2011). Some important states of lithium-ion battery, such as the state of charge (SOC), must be monitored by battery management system (BMS) during charging and discharging procedures. In recent years, estimating SOC methods based on electrical equivalent models (ECM) by using measured load current and terminal voltage became popular. Such a methodology requires accurate parameters of battery in ECMs. Battery state estimation, in which estimating SOC is very important, is a key advanced BMS feature in EVs. Precise modeling and state estimation will allow stable operation, facilitate optimal battery operation, and provide the fundamentals for security supervision. Note that, the parameters of LIB have a significant impact on precise modeling and state estimation. Therefore, it is significant to identify the parameter of lithium-ion battery accurately.

Many research works have attempted to identify the parameters of LIB precisely. For the convenience of reading, we would like to categorize such research works into several groups as shown in Table 1. The reviews on these related works in detail could be seen below.

Although parameters identification methods are applied for various models of battery, such as electric model (Rakhmatov, Vrudhula, & Wallach, 2003), thermal model (Jeon & Baek, 2011), aging model (Belt, Utgikar, & Bloom, 2011), Physical models (Ramadesigan et al., 2011; Smith, Rahn, & Wang, 2007; Xu, Li, & Liu, 2018), ECMs (Andre et al., 2011; Chen & Rincon-Mora, 2006; Hu, Yurkovich, Guezennec, & Yurkovich, 2009; Li, Lai, Wang, Lyu, & Wang, 2016), et al. The majority of identifying methods are based on ECMs representing the electric behavior of the battery. This is because ECMs make a good trade-off

[*] Corresponding author at: Marine Engineering College, Dalian Maritime University, 1 Linghai Road, Dalian, Liaoning Province 116026, People's Republic of China.

E-mail address: chuanwang0101@163.com (C. Wang).

https://doi.org/10.1016/j.eswa.2022.117192
Received 25 October 2021; Received in revised form 28 March 2022; Accepted 3 April 2022
Available online 6 April 2022

Nomenclature

CC	Cooperative co-evolution
CCDE	Cooperative co-evolution differential evolution
DE	Differential evolution
DST	Dynamic stress test
EA	Evolutionary algorithm
ECM	Equivalent circuit model
EV	Electric vehicle
FUDS	Federal urban dynamic schedule
LIB	Lithium-ion battery
LS	Least square
LSOP	Large scale optimization problem
OCV	Open-circuit voltage
SOC	State of charge
C_P	Capacitance describes the polarization characteristics of a battery (F)
Cr	Crossover rate
Cr_u	Upper limit of Cr
D	Length or dimension of a target vector or individual
D_{sub}	Dimension of a target vector in X_{sub}
F	Scale factor
F_u, F_l	Upper and lower limits of F
FES	Functional evaluations
G	Index of generations
G_{max}	Maximum of generations
K_1, K_2	Intermediate variables for improved Euler's method
MAX_FES	Maximum functional evaluations
MAX_FES_{sub}	Maximum functional evaluations for sub-populations.
N	Number of groups or sub-populations
NP	Population size
R_0	Inner resistance (Ω)
R_P	Resistance describes the polarization characteristics of a battery (Ω)
$\mathbf{U}_{i,G}$	Trail vector for ith individual at Gth generation
$\mathbf{V}_{i,G}$	Mutation vector for ith individual at Gth generation
V_L, V_U	Lower and upper cut-off voltage of battery, respectively (V)
$\mathbf{X}_{best,G}$	The target vector with the best fitness value at Gth generation
$\mathbf{X}_{i,G}$	Target vector, i.e., ith individual at Gth generation
$\mathbf{X}_{pnbest,G}$	The target vector with the best fitness value at Gth generation
$\mathbf{X}_{\min}$, $\mathbf{X}_{\max}$	lower and upper limits for variables to be optimized, respectively
$\mathbf{X}_{sub}$	Sub-population
h	Step size (s)
i_L	Load current (A)
k	Sampling time step in a dataset
k_{max}	Total number of data points in a dataset
m	Sampling interval when an optimization algorithm is used
t	Real sampling time (s)
v_C	Voltage of C_P (V)
v_{OCV}	Voltage of OCV (V)
v_{term}	Terminal voltage (V)
$\widehat{v}_{term}$	Estimated terminal voltage (V)
$\mathbf{x}^*$	Global best solution found by an algorithm

Table 1
Summary of related works on parameters identification of LIB based on ECMs.

Methods	References
Kalman filter family	Wei, Dong, Chen, & Kang (2017); Wang, Pan, Liu, Chen, & Ling (2018); Zeng, Tian, Li, & Tian (2018); Zeng, Tian, Li, & Tian (2018); Dai, Wei, Sun, Wang, & Gu (2012); Lee, Dai, & Chuang (2018); Sepasi, Ghorbani, & Liaw (2014); Zhu, Xu, Liu, & Zheng (2019); Chen, Yang, Zhao, Wang, & He (2019); Zeng, Zhang, Yang, Xie, & Shi (2019); Xuan, Shi, Chen, Zhang, & Wang (2020); Peng, Luo, He, & Lu (2019)
Genetic Algorithm (GA)	Mu, Xiong, Zheng, Chang, & Chen (2017)
Particle Swarm Optimization-Genetic Algorithm (PSO-GA)	Zhang, Wang, Liu, & Chen (2018)
Co-evolutionary Particle Swarm Optimization (CPSO)	Yu, Xiao, Li, Zhu, & Huai (2017)
Artificial Ecosystem-based Optimization (AEO)	Ferahtia et al. (2021)
Optimization Toolbox of Matlab	Miniguano, Barrado, Lazaro, Zumel, & Fernandez (2020)

between the mathematical models and the physical ones. Moreover, it is simple to implement for real time applications (Barcellona & Piegari, 2017). So we made a short overview which focuses on this mainstream. Numerous works have investigated parameters identification of battery by using various methods. Usually, SOC is simultaneously estimated. A typical representation is the recursive algorithm with iterative characteristics, including the Kalman filter (Wei, Dong, Chen, & Kang, 2017) and its derivative algorithms (Wang, Pan, Liu, Chen, & Ling, 2018), as well as least squares (LS) (Zeng, Tian, Li, & Tian, 2018). The family of Kalman filters consists of various derivative methods, such as linear Kalman filter (Mastali et al., 2013), extended Kalman filter (Dai, Wei, Sun, Wang, & Gu, 2012; Lee, Dai, & Chuang, 2018), adaptive extended Kalman filter (Sepasi, Ghorbani, & Liaw, 2014; Zhu, Xu, Liu, & Zheng, 2019), unscented Kalman filter (Chen, Yang, Zhao, Wang, & He, 2019; Zeng, Zhang, Yang, Xie, & Shi, 2019), central difference Kalman filter (Xuan, Shi, Chen, Zhang, & Wang, 2020), cubature difference Kalman filter (Peng, Luo, He, & Lu, 2019), et al. Generally speaking, Kalman filter algorithm can be regarded as a recursive mechanism which estimates and updates system states based on the feedback errors. Linear Kalman filter is optimal for linear system. Since the open-circuit voltage (OCV) function is non-linear for SOC estimation (as shown in Fig. 2), other methods such as extend Kalman filter usually linearize the non-linear system at each time step. However, the accuracy of SOC estimation by using Kalman filter family heavily relies on OCV function linearization and parameters in the ECM. The OCV function is non-linear, and the parameters of battery model is hard to identify. On the other hand, although LS was applied for identifying parameters of battery (Zheng et al., 2016), this method essentially is not suitable for non-convex optimization problems with many local optima. According to our previous experience, parameters identification of battery is probably a multi-modal problem. That is to say, LS is not a global search algorithm, resulting in stagnation due to local optima. Such methods could not find parameters of battery as globally optimal solutions. Moreover, when LS and Kalman filter are used for estimating SOC (Zheng et al., 2016), v_{OCV} is usually assumed as a constant value. In this situation, the uncertainty of v_{OCV} is ignored by using LS, resulting in not accurate parameters estimated. Some recent published papers focused on identifying parameters based on ECM and battery historical operating data directly. The discrete state-space equations of the impedance model were inferred by Grtinwald-Letnikov definition and parameters of the model including the order of the fractional element were identified together by genetic algorithm (Mu, Xiong, Zheng, Chang, & Chen, 2017). To predict the battery pack SOC, the method of particle swarm optimization-genetic algorithm was applied in battery pack model parameters identification (Zhang, Wang, Liu, & Chen, 2018). A co-evolutionary particle swarm optimization (CPSO) method was used to identify battery parameters (Yu, Xiao, Li, Zhu, & Huai, 2017). A parameter identification window (PIW), which has the features of a fixed data length and real-time response, was used to store a piece of data that indicates the battery operation at the current moment. However, these methods still assumed that the OCV curve could be partially linearized within a PIW, or only partial measured current and voltage were used for optimizing parameters of battery. As a results, the identified parameters are not global optimal. An optimal parameter identification strategy for the Shepherd model of LIB was presented by applying an artificial ecosystem-based optimization (AEO) algorithm (Ferahtia et al., 2021). However, this method was used for identifying parameters based on a polynomial model. As shown in Fig. 2, it is not appropriate and accurate to describe OCV by using a polynomial function, nor v_{term} and SOC. In a platform which mixed experimental instruments and Simulink of Matlab, the parameters of LIB models were identified with the Optimization Toolbox of Matlab (Miniguano, Barrado, Lazaro, Zumel, & Fernandez, 2020). Whereas, this parametric estimation procedure requires complicated interfaces between the data logger and Matlab. Moreover, the Optimization Toolbox of Matlab is designed for general optimization problem. It may be not appropriate to use the toolbox for identifying parameters of battery due to its distinctive fitness landscape.

Based on the above observations, this paper proposed a cooperative co-evolution differential evolution (CCDE) algorithm to identify parameters of LIB. First, to describe the dynamic behaviors of battery, we presented an ECM with first-order RC network. Without making any assumption or approximation, improved Euler's numerical method was utilized to solve the differential equations directly. Second, an optimizing objective function was built to minimize errors between the true and optimized terminal voltages. In that optimization model, parameters of battery (R_0, R_P and C_P) and $v_{OCV}(t)$ at each sampling point were considered as variables to be optimized, resulting in a very high-dimension problem. Third, such an optimization problem was transformed into a large scale optimization problem (LSOP). Based on the character of parameters identification, we proposed a new *m*-decomposition method which is different from general grouping methods for benchmark functions and its corresponding differential evolution (DE) algorithm to solve this LSOP. Fourth, to demonstrate effectiveness of the proposed methodology, comprehensive experiments on two commonly used dynamic test datasets were conducted, compared with seven state-of-the-art CC algorithms.

The main advantages of this methodology are listed as follows:

The proposed methodology showing generality and universality can be applied for various models of battery, such as ECMs (first-order, second-order, et al), electrochemical models, and other coefficients in math models of battery. Besides, the dimension/length of parameters in a model is allowed to be quite high.

Different from LS method and other gradient-based methods, this proposed methodology does not assume that models of battery are differential. This is a primary advantage compared with LS methods. This advantage allows the problem to be optimized could be with almost any non-linear characters, such as non-convex, discontinuity, many local optima, et al.

The proposed method is a global optimization algorithm. It is superior to LS which is a local optimization method.

The main contributions of this methodology are shown as follows:

The proposed CCDE methodology can be applied for any ECM of battery with non-linearization, without any linearization or pre-assumptions. When identifying parameters of battery based on ECM, v_{OCV} is a time-varying and non-linear variable. Most existing works made v_{OCV} linearized or partially linearized during a dynamic testing process. Also, some studies built a model for v_{OCV} and optimized its corresponding coefficients. In fact, it is difficult to obtain satisfactory precision in practical applications. In this investigation, v_{OCV} during the whole dynamic test procedure were optimized as well as parameters of battery based on the measured load current and terminal voltage. In other words, the proposed methodology can deal with parameters identification of battery with any non-linear characters.

Parameters identification of battery was transformed into a LSOP. Based on the ECM of battery and measured datasets of current and voltage, optimal parameters and OCV curve were optimized simultaneously, making the OCV curve most fitted to measured data. Compared with Kalman filter and observer-based methods, the parameters of battery could be globally optimal by using this methodology, and this method was not sensitive to non-linearity of the system.

Considering the character of parameters identification of battery, a new *m*-decomposition method and its corresponding optimizer were proposed. For general LSOPs, there are no overlapped variables in different sub-populations. In this paper, the optimal parameters of battery found so far were allotted to each sub-population in every updating cycle during evolving. As a result, the final optimal parameters of battery are the globally optimized variables which are consistent with the whole measured load current and terminal voltage.

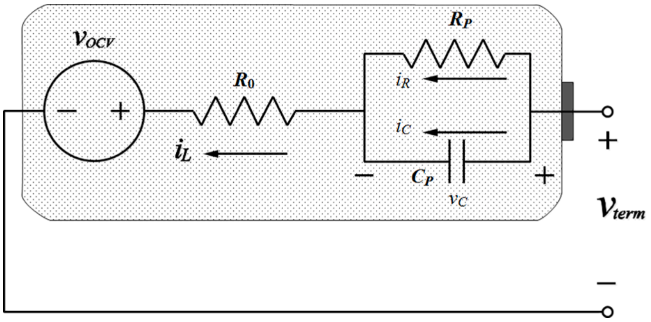


Fig. 1. First-order RC model of lithium-ion battery.

The rest part of this paper is organized as follows: Section 2 builds up the ECM for parameters identification of LIB. Section 3 gives a full description on the proposed CCDE algorithm in details. Section 4 conducts comprehensive experiments to test the performance of the proposed method on different dynamic test profiles. Finally, the whole work is summarized in Section 5.

2. Problem formulation of parameters identification for battery

This part builds up an overall ECM for parameters identification of LIB, and provides numerical solution. To simulate the dynamic characters of battery, Section 2.1 builds up a first-order RC model of battery. By using numerical integration method, the differential equations are solved. Section 2.2 formulates the math model of this optimization problem.

2.1. First-order RC model of battery and numerical solution

Parameters identification of battery is based on ECMs. Thus, an ECM which describes the static and dynamic behaviors of battery should be built up for practical applications. The ECM method has been widely used due to its clear mathematical expression and less parameters and relatively easy identification (Dubarry & Liaw, 2007). For LIBs, the first-order RC model that is also called the Thevenin model, revealing superiority over other ECMs (Zheng et al., 2016) was adopted in this study.

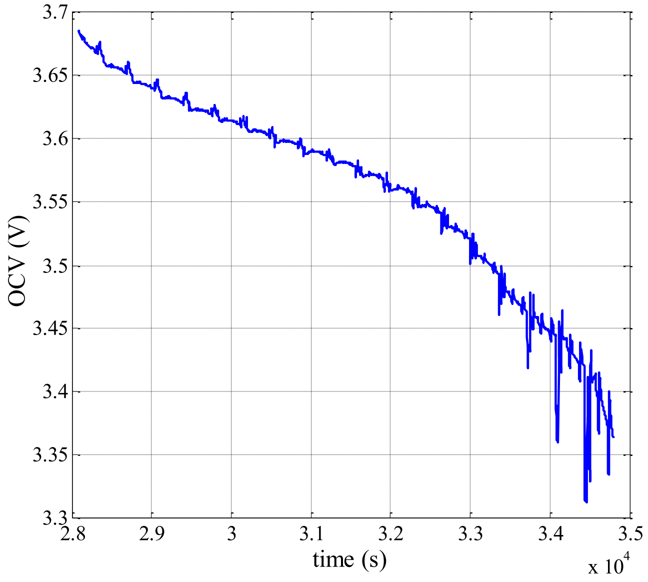


Fig. 2. A v_{OCV} curve during DST for 50% battery level at 25 °C.

It should be noted that, although first-order model was adopted in this paper, the proposed methodology could be applied for other ECMs with more complex structures. Fig. 1 shows the ECM of LIB.

As shown in Fig. 1, the first-order RC model is composed of an ideal voltage source v_{OCV}, an inner resistance R_0 and an RC paralleled circuit which is used to simulate polarization effect. According to Kirchhoff's voltage and current law, by introducing polarization resistance R_P and polarization capacitance C_P, the math equations of the adopted battery model can be described as follows:

$$\begin{cases} \dfrac{dv_C}{dt} = \dfrac{i_L}{C_P} - \dfrac{v_C}{R_P C_P} \\ v_{term} = v_{OCV} + i_L R_0 + v_C \end{cases} \tag{1}$$

Where, v_C denotes the voltage of C_P. i_L represents the load current. v_{OCV} is the open-circuit voltage of battery. v_{term} denotes the terminal voltage of battery. These variables in Eq. (1), excluding parameters of battery (R_0, R_P, C_P), vary versus sampling time t.

To solve this differential equation in Eq. (1), modified Euler's method (Allahviranloo & Salahshour, 2011); Kamrani, Hosseini, & Hausenblas, 2018) was used in this work. After that, the math model of battery can be discretized as follows:

$$\begin{cases} v_C(k) = v_C(k-1) + \dfrac{1}{2}h(K_1 + K_2) \\ \widehat{v}_{term}(k) = v_{OCV}(k) + i_L(k)R_0 + v_C(k) \end{cases} \tag{2}$$

Where, k represents the sampling time index. h denotes step size which also can be represented as Δk. $K_1 = i_L(k)/C_P$ - $v_C(k$-1$)/R_P C_P$, $K_2 = i_L(k)/C_P$ - $(v_C(k$-1$) + hK_1)/R_P C_P$. v_{term} with "^" denotes the terminal voltage is an estimated value which is used to compare with the true value. By using Eq. (2), given $i_L(k)$, $v_C(k$-1$)$, $v_{OCV}(k)$ and constant parameters which are R_0, R_P and C_P, the estimated $v_{term}(k)$ can be calculated to compare with the measured terminal voltage. Conversely, with $i_L(k)$, $v_C(k$-1$)$, $v_{OCV}(k)$ and $v_{term}(k)$, the parameters of battery can be identified. However, v_{OCV} which maps with SOC is a non-linear variable varying with time. Fig. 2 shows an example of OCV curve during DST test. As a result, least-square (LS) method is not suitable for identifying the parameters of battery due to the non-linearization of the model. This is because the function of OCV with non-linearity is unknown to us. To address this issue, the authors tried to optimize the parameters and v_{OCV} simultaneously by using a proposed cooperative co-evolutionary (CC) DE algorithm. The optimization math model of the problem will be built in the next section.

2.2. Optimizing model of parameters identification

This part gives the optimization objective function and constraints for identifying parameters of battery. This investigation is to find optimal parameters (R_0, R_P and C_P) and v_{OCV} to minimize the error between estimated and measured terminal voltage. Datasets of current and terminal voltage are used to verify the estimated terminal voltage. Thus, the optimizing objective function was built as follows:

$$\min fit(\boldsymbol{x}) = \sum_{k=1}^{k_{\max}} |\widehat{v}_{term}(k) - v_{term}(k)| \tag{3}$$

Where, * MERGEFORMAT denotes the estimated terminal voltage at kth time step, $v_{term}(k)$ represents the measured terminal voltage at kth time step. k_{max} denotes total number of data points in a dataset. × denotes the single row vector to be optimized in the above model, showing the variables as follows:

$$\boldsymbol{x} = [R_0, R_P, C_P, v_{OCV}(k)], k = 1, 2, ..., k_{max} \tag{4}$$

In addition, there are some constraints which should be satisfied for identifying parameters of battery. The constraints are shown as follows:

$$\begin{cases} R_0 > 0 \\ R_P > 0 \\ C_P > 0 \\ V_L \leqslant v_{OCV}(k) \leqslant V_U \end{cases} \tag{5}$$

Where, V_L and V_U represent the lower and upper cut-off voltage of battery, respectively. In this paper, the variable which violates its upper or lower bound will be re-initialized within the legal range.

Here, the authors would like to talk about the dimension of this optimization problem. Taking DST for 50% battery level at 25 °C as an example, there are up to 9501 data points. After discarding the initial part data of charging and discharging, k_{max} is up to 6650. That means the dimension of an optimization problem D should be set to $k_{max} + 3 = 6653$. This is the reason why many previous works simplified v_{OCV} to a polynomial function or even linear function. As we discussed in the introduction, such a linearization or simplification for v_{OCV} will lead to misuse of identification methods and reduction of SOC estimation accuracy. To tackle this problem, this paper insisted on optimizing the parameters of battery and v_{OCV} simultaneously by using the proposed cooperative co-evolutionary DE algorithm without any assumption to simplify the math model. The biggest advantage of this methodology is that, the model with the identified parameters is most consistent with the nonlinear characteristics of the system and the actual measured current and voltage. Due to the very big dimension D of the optimization problem, this problem should be classified as large scale optimization problem (LSOP). In the next section, we present a new cooperative co-evolutionary DE algorithm with local search strategy to solve this problem.

3. The proposed method

This part fully describes the proposed methodology. Section 3.1 illustrates the motivation of the proposed methodology. Section 3.2 gives a review of related work on decomposition methods. Section 3.3 takes standard DE algorithm as an example to introduce the optimizer. Section 3.4 presents the proposed cooperative co-evolutionary (CC) DE algorithm and the framework of the presented methodology. Section 3.5 shows the implement of the proposed method for identifying parameters.

3.1. Motivation

A good number of papers have investigated the parameters of LIB based on various models. In the majority of their research, some key variables, such as v_{OCV} and SOC, have been modeled up as some complex functions. However, it is quite difficult to build up such a function accurately. Taking DST data for 50% battery level at 25 °C as an example, given a set of referenced parameters, the v_{OCV} curve at each sampling time step k was plotted as shown in Fig. 2. Such a variable is hard to be presented by using a polynomial or exponential function accurately. If more complicated dynamic test is used, e.g. FUDS, its nonlinearity and fluctuation will become more violent. Therefore, this drove the authors to identify the key parameters in ECMs directly on based on measured i_L and v_{term}. On the other hand, as we claimed in the introduction, LS and its variants are widely used for identifying parameters of LIB due to its simplicity, but it is not a global optimal algorithm. For nonconvex problems, it will be easily trapped by local optima or saddle points and never escapes. Based on our previous experimental results, the problem of identifying parameters of LIB showed some characters of multi-modal problems. That is to say, essentially, LS is not suitable for solving the this problem. This also drove us to develop a new methodology to address this issue.

3.2. Related work on decomposition method

CC methodology can effectively deal with an LSOP by cooperatively optimizing the lower-dimensional sub-problems which are obtained through decomposition (Ren et al., 2019). Decomposition method plays an important role in improving the performance of CC. The early CC algorithms divided an D-dimensional problem into D sub-problems with one dimension (Bergh & Engelbrecht, 2000; Liu, Yao, Zhao, & Higuchi, 2001). A splitting-into-half decomposition strategy was used for designing the cooperative co-evolutionary DE algorithms (Potter & De Jong, 2000; Shi, Teng, & Li, 2005). A general decomposition guideline for PSO was suggested to divide an D-dimensional problem into n s-dimensional sub-problems where $ns = D$ and $s \ll D$ (van den Bergh & Engelbrecht, 2004). In addition, some researchers developed other decomposition methods on non-separable LSOPs, such as random decomposition methods (Omidvar, Li, Yang, & Yao, 2010; Yang et al., 2008), learning-based decomposition methods (Ge, Sun, Tan, Chen, & Chen, 2017; Ge, Sun, Yang, Yoshida, & Liang, 2015; Sun, Yoshida, Cheng, & Liang, 2012). delta grouping (Omidvar, Li, & Yao, 2010) and differential grouping (Hu, He, Chen, & Zhang, 2017; Mei, Omidvar, Li, & Yao, 2016; Omidvar, Li, Mei, & Yao, 2014; Omidvar, Yang, Mei, Li, & Yao, 2017). Based on these decomposition/grouping methods, we would like to give a general structure of CC for solving LSOPs to make it clearer. The pseudo code of general CC is shown in Algorithm 1.

Algorithm 1. The pseudo code of general CC.

```
Input: $NP$, MAX_$FES$, MAX_$FES_{sub}$, $D$, $\mathbf{X}_{min}$, $\mathbf{X}_{max}$, $N$
Output: fit($\mathbf{x}^*$), $\mathbf{x}^*$
[fit($\mathbf{X}$), $\mathbf{X}$] ← initialization($NP$, $D$, $\mathbf{X}_{min}$, $\mathbf{X}_{max}$);
[fit($\mathbf{x}^*$), $\mathbf{x}^*$] ← sort(fit($\mathbf{X}$), $\mathbf{X}$);
[$\mathbf{X}_{sub}$] ← decomposition($\mathbf{X}$, $N$); % $\mathbf{X}_{sub} = \{\mathbf{X}_{sub,1}, \ldots, \mathbf{X}_{sub,n}, \ldots, \mathbf{X}_{sub,N}\}$
$FES$ ← $NP$;
while $FES \le$ MAX_$FES$
    for $n$ ← 1 to $N$
        [$\mathbf{X}_{sub,n}$, $\mathbf{x}^*$] ← optimizer($\mathbf{X}_{sub,n}$, MAX_$FES_{sub}$, $\mathbf{x}^*$);
        $FES$ ← $FES$ + MAX_$FES_{sub}$;
    end
end
```

From the general framework of CC algorithm, the whole population is grouped into N sub-populations which contain NP individuals but with partial dimensions. For each sub-population, an optimizer is used to update the sub-population and the best individual $\mathbf{x}^*$ at the partial dimensions. This procedure is looped until the termination condition is met.

Although the general CC algorithm has provided an effective tool to deal with LSOPs, these techniques were developed for general optimization problems. The parameters identification of battery requires an CC framework and optimizer with specific and targeted operations. It may be not suitable to just simply incorporate general CC algorithm into parameters identification of battery. Later in this work, a more targeted CC and an optimizer with a local search strategy will be presented. In this next section, standard DE algorithm as an optimizer is described.

3.3. DE algorithm

Generally speaking, the conventional DE algorithm consists of four operations, which are initialization, mutation, crossover and selection. The last three operations are looped until the maximum functional evaluations (MAX_FES) is exhausted.

3.3.1. Initialization

Similar to other EAs, DE algorithm is a population-based stochastic search method which seeks for a global optima. The population consists of NP D-dimensional real-valued vectors, in which each vector, usually called a target vector, is denoted as $\mathbf{X}_{i,G} = \{x^1{}_{i,G}, x^2{}_{i,G}, \ldots, x^D{}_{i,G}\}$, $i = 1, 2, \ldots, NP$. NP is the size of the population, D is length or dimension of the target vector, G denotes the index of generations. Then, within the upper limits ($\mathbf{X}_{\max} = \{x^1{}_{\max}, x^2{}_{\max}, \ldots, x^D{}_{\max}\}$) and lower limits ($\mathbf{X}_{\min} = \{x^1{}_{\min}, x^2{}_{\min}, \ldots, x^D{}_{\min}\}$) of the search space, each target vector is uniformly randomized as follows:

$$\boldsymbol{X}_i = \boldsymbol{X}_{min} + \boldsymbol{U}[0,\ 1](\boldsymbol{X}_{max} - \boldsymbol{X}_{min}) \tag{6}$$

where, $\mathbf{U}[0, 1]$ is a row vector of random numbers uniformly generated within [0, 1].

3.3.2. Mutation

After initialization, for the corresponding target vector $\mathbf{X}_{i,G}$, a mutation operator is executed to generate each mutant vector $\mathbf{V}_{i,G}$ in the current population. We will list five most widely used mutation strategies as follows (Das, Mullick, & Suganthan, 2016):

DE/rand/1:

$$\boldsymbol{V}_{i,G} = \boldsymbol{X}_{r_1^i,G} + F\left(\boldsymbol{X}_{r_2^i,G} - \boldsymbol{X}_{r_3^i,G}\right) \tag{7}$$

DE/best/1:

$$\boldsymbol{V}_{i,G} = \boldsymbol{X}_{best,G} + F\left(\boldsymbol{X}_{r_1^i,G} - \boldsymbol{X}_{r_2^i,G}\right) \tag{8}$$

DE/current-to-best/1:

$$\boldsymbol{V}_{i,G} = \boldsymbol{X}_{i,G} + F(\boldsymbol{X}_{best,G} - \boldsymbol{X}_{i,G}) + F\left(\boldsymbol{X}_{r_1^i,G} - \boldsymbol{X}_{r_2^i,G}\right) \tag{9}$$

DE/best/2:

$$\boldsymbol{V}_{i,G} = \boldsymbol{X}_{best,G} + F\left(\boldsymbol{X}_{r_1^i,G} - \boldsymbol{X}_{r_2^i,G}\right) + F\left(\boldsymbol{X}_{r_3^i,G} - \boldsymbol{X}_{r_4^i,G}\right) \tag{10}$$

DE/rand/2:

$$\boldsymbol{V}_{i,G} = \boldsymbol{X}_{r_1^i,G} + F\left(\boldsymbol{X}_{r_2^i,G} - \boldsymbol{X}_{r_3^i,G}\right) + F\left(\boldsymbol{X}_{r_4^i,G} - \boldsymbol{X}_{r_5^i,G}\right) \tag{11}$$

where, $r^i{}_1$, $r^i{}_2$, $r^i{}_3$, $r^i{}_4$, $r^i{}_5$ are exclusive integers randomly selected within {1, 2…., NP}, and the indices are also different from i. F is called scale factor which is usually a positive value. $\mathbf{X}_{best,G}$ is the best target vector in current population at Gth generation.

After mutation, each mutant vector $\mathbf{V}_{i,G}$ may violate the upper and lower limits of the search space. For many real engineering problems, such as parameters identification of LIB in this paper, it is not allowable to violate upper and lower limits. In this paper, the following operators are used to avoid violating these constraints: at jth dimension, if $v^j{}_{i,G}$ is bigger than $x^j{}_{max}$, then $v^j{}_{i,G}$ will be set to $x^j{}_{max}$; if $v^j{}_{i,G}$ is smaller than $x^j{}_{min}$, then $v^j{}_{i,G}$ will be set to $x^j{}_{min}$. The algorithmic description on this operator is shown in lines 7 to 10 in Algorithm 2.

3.3.3. Crossover

In this operator, each trial vector $\mathbf{U}_{i,G}$ in current population at Gth

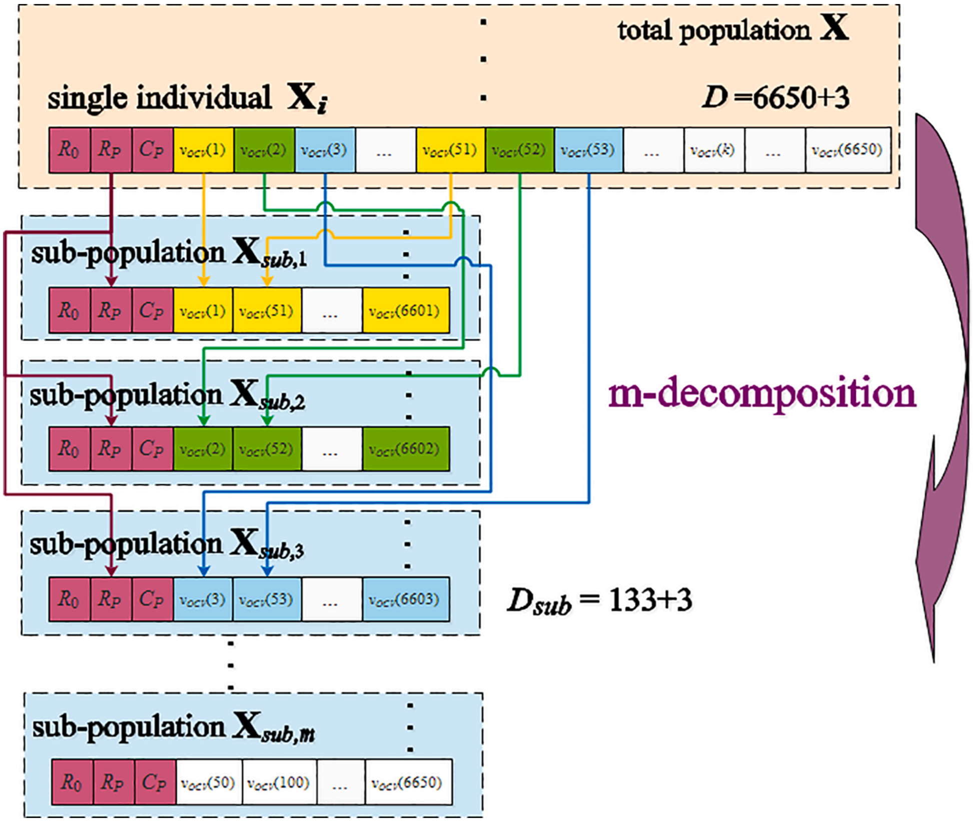


Fig. 3. Block diagram of the proposed decomposition method for a single individual. $N = m = 50$, $D = 3 + 6650$.

generation is formed by mixing the corresponding target vector $\mathbf{X}_{i,G}$ and mutant vector $\mathbf{V}_{i,G}$. Usually, a binomial crossover scheme is used as follows (Das & Suganthan, 2011):

$$u^j_{i,G+1} = \begin{cases} v^j_{i,G+1}, & \text{if } rand_j \leqslant Cr \text{ or } j = j_{rand} \\ x^j_{i,G}, & \text{otherwise} \end{cases}, j = 1, 2, ..., D \tag{12}$$

where, j_{rand} is designed as a randomly generated integer within {1, 2, …, D} to make the trial vector different from the corresponding target vector at least at one dimension. $rand_j$ is a randomly generated number obeying uniform distribution within [0, 1]. Cr is crossover rate which is usually a positive value.

3.3.4. Selection

In this operator, DE algorithm adopts a one-to-one comparison between the target vector $\mathbf{X}_{i,G}$ and the trial vector $\mathbf{U}_{i,G}$. If the fitness value of $\mathbf{U}_{i,G}$ is less than or equal to the fitness value of $\mathbf{X}_{i,G}$, then the target vector will be replaced by the trial vector. Otherwise, the target vector will be kept unchanged in the next generation:

$$X_{i,G+1} = \begin{cases} U_{i,G+1}, & \text{if } f(U_{i,G+1}) \leqslant f(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases} \tag{13}$$

where, $f(\bullet)$ is the objective function to be minimized. In the above equation, it is helpful for the population to accept the trial vector with then equal fitness value, resulting in a more diversified population. The algorithmic description of standard DE is shown as Algorithm 2.

Algorithm 2. *The pseudo code of standard DE algorithm.*

```
Input: NP, MAX_FES, D, $X_{min}$, $X_{max}$, F, Cr
Output: fit(x*), x*
[fit(X), X] ← initialization(NP, D, $X_{min}$, $X_{max}$);
G ← 1, FES ← NP;
```

(continued on next column)

(*continued*)

```
Input: NP, MAX_FES, D, $X_{min}$, $X_{max}$, F, Cr
while FES ≤ MAX_FES
  for i ← 1 to NP
    [$V_{i,G}$] ← mutation($X_{i,G}$);
    for j = 1 to D
      [$v^j_{i,G}$] ← min($v^j_{i,G}$, $x^j_{max}$);
      [$v^j_{i,G}$] ← max($v^j_{i,G}$, $x^j_{min}$);
    end
    [$U_{i,G}$] ← crossover($V_{i,G}$);
    [$X_{i,G+1}$] ← selection($U_{i,G}$, $X_{i,G}$);
    FES ← FES + 1;
    [fit(x*), x*] ← min(x*, $X_{i,G+1}$);
  end
  G = G + 1;
end
```

3.4. The proposed CCDE algorithm

3.4.1. m-decomposition

In this section, the proposed CC framework is described. First, the character of the problem in this paper should be reviewed. Based on the math model built up in Section 2, we try to find the most fitted × shown in Eq. (4) according to measured current and voltage during dynamic tests for battery. R_0, R_P, C_P are the parameters of battery to be optimized, and the three values do not change with time. v_{OCV} is a non-linear variable varying versus time, and it is strongly co-related with SOC. Here is an example shown in Fig. 2 to illustrate the character of v_{OCV} curve during a DST procedure.

In Fig. 2, v_{OCV} does not vary dramatically versus time, but fluctuate frequently. For such a variable, accurate approximations will not be resulted by using interpolation methods nor curve fittings. Thus, the authors believed that identifying parameters at each sampling time k could ensure the parameters and model are most consistent with the measured values. This is the reason why we transform the problem into a LSOP. Based on these characters of parameters identification for battery, we proposed a unique decomposition/grouping method. First, R_0, R_P, C_P are chosen into every sub-populations. Because the three parameters are constant and fitted to all sampled data, it is reasonable to optimized the three parameters in each cycle. Second, v_{OCV} is grouped evenly at every m sampling intervals. Different from other grouping methods, such a decomposition method is helpful to reflect the overall distribution of v_{OCV} versus time. Taking DST for 50% battery level at 25 °C as an example, there are up to 9501 data points. After discarding the initial part data of charging and discharging, k_{max} is up to 6650. That means, excluding R_0, R_P and C_P, the dimension D of the problem is 6650. Assuming that m is set to 50, then the number of sub-populations is $N = m = 50$. As a result, the dimension of each sub-population is $D/m = 133$. Fig. 3 shows an illustration on this decomposition method which is named m-decomposition in this paper.

So far, a block diagram of the presented m-decompostion method was illustrated. After initialization, the total population $\mathbf{X}$, which is enclosed within a rectangular filled with slight mauve, is a $NP * D$ matrix. Excluding R_0, R_P and C_P, at every m dimension/interval in each single individual $\mathbf{X}_i$, $v_{OCV}(k)$ is evenly sampled to compromise an individual in sub-population $\mathbf{X}_{sub,n}$, where $n = 1, 2, \ldots, N$. Such a m-decomposition method can dramatically reduce the dimension of the original LOSP. Besides, it could reflect the overall distribution of v_{OCV}, compared with other complex grouping methods. Moreover, the parameters of battery, i.e., R_0, R_P and C_P, are optimized in every sub-population. It could ensure the calculated parameters are globally optimal solutions, compared with LS methods.

3.4.2. Mutation strategy

Based on the above characters of this optimization model and the ranges of variables to be optimized in each sub-populations, a DE variant was proposed for solving this problem. Different from some famous and highly-cited DE variants with various mutation strategies, such as SaDE (Qin, Huang, & Suganthan, 2009), CoDE (Wang, Cai, & Zhang, 2011), EPSDE (Mallipeddi, Suganthan, Pan, & Tasgetiren, 2011), DMPSADE (Fan & Yan, 2015), MPEDE (Wu, Mallipeddi, Suganthan, Wang, & Chen, 2016) and IMSaDE (Wang, Li, Yang, & Liu, 2018), we need a DE algorithm with a good balance between exploration and exploitation. According to our previous experimental experience, we found this problem is probably a non-convex optimization problem with many local optima. Meanwhile, in each cycle/generation, MAX_FES_{sub} for every sub-population is limited. And, v_{OCV} is a continuous variable and does not vary greatly at adjacent sampling times. As a result, local searches in parallel are required. Thus, the new mutation strategy which is named as “current-to-pnbest” using a ring topology was developed as follows:

$$\mathbf{V}_{i,G} = \mathbf{X}_{i,G} + F\left(\mathbf{X}_{pnbest,G} - \mathbf{X}_{i,G}\right) + F\left(\mathbf{X}_{r_1^i,G} - \mathbf{X}_{r_2^i,G}\right) \tag{14}$$

Where, r^i_1 and r^i_2 are randomly selected indices of the left and the right target vectors of ith individual. In a ring-connected population topology, each individual has only two neighbors to share information with each other. That is to say, the ith individual only could communicate with the i-1th and $i + 1$th individual. In this paper, for ith target vector, the left target vector means the i-1th individual, and the right target vector denotes the $i + 1$th individual. For example, the 19th individual only connects with the 18th (the left target vector) and 20th (the right target vector) individual; the 20th individual only connects with the 19th (the left target vector) and 21th (the right target vector) individual, and so on. Such a population topology naturally builds up some overlapped niches (sub-populations) to search the solution space in parallel, without introducing any niching parameter. $\mathbf{X}_{pnbest,G}$ denotes the best personal best vector which is chosen from $\mathbf{X}_{pbest,i}$, $\mathbf{X}_{pbest,i\text{-}1}$ and $\mathbf{X}_{pbest,i+1}$. Different from previous designs in DE, the authors built up local memory for each target vector to “anchor” the whole population, avoiding being trapped by local optima. Such a mutant vector shown as Eq. (14) will search within a restricted niche with three members without introducing any niching parameters. In other words, the design of $\mathbf{X}_{pbest}$ is quite similar to the personal best swarm which is denoted as $\mathbf{P}_i$ in PSO. To make the proposed mutation operator more understandable, a full description on how to generate a mutant vector using this mutation operator is presented in Fig. 4. In a ring topology, each target vector only connects with its left and right neighbors by index. Every target vector has its own personal best named as $\mathbf{X}_{pbest}$.

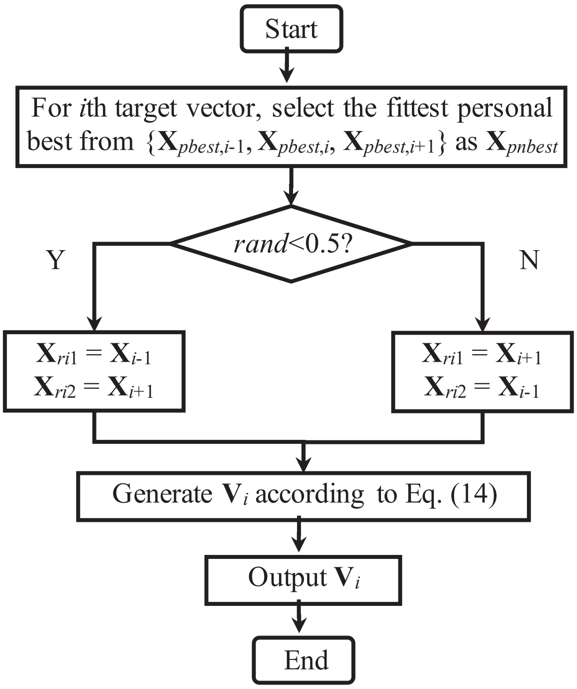

Start
For ith target vector, select the fittest personal best from $\{\mathbf{X}_{pbest,i\text{-}1}, \mathbf{X}_{pbest,i}, \mathbf{X}_{pbest,i+1}\}$ as $\mathbf{X}_{pnbest}$
$rand$<0.5?
Y
N
$\mathbf{X}_{ri1} = \mathbf{X}_{i\text{-}1}$
$\mathbf{X}_{ri2} = \mathbf{X}_{i+1}$
$\mathbf{X}_{ri1} = \mathbf{X}_{i+1}$
$\mathbf{X}_{ri2} = \mathbf{X}_{i\text{-}1}$
Generate $\mathbf{V}_i$ according to Eq. (14)
Output $\mathbf{V}_i$
End

Fig. 4. The flowchart of generating a mutant vector by “current-to-pnbest” using a ring topology.

Here, we would like to state the motivation and reason of designing such a mutation strategy. As well known, mutation operator plays a very significant role on performances of DE algorithms. For solving this problem, mutant vectors should be generated by using very few target vectors within a restricted neighborhood to locate local optima as many as possible. Therefore, the ring topology, which is “the slowest, most indirect communication pattern” (Kennedy & Mendes, 2002), was adopted as the population topology in this study. As a result, by using such a topology, each target vector and their immediate neighbors build up a niche automatically, without introducing any niching parameters. Besides, in PSO algorithm, we have noticed a quite unique characteristic. That is, each particle can keep its own memory or personal best. All these personal bests constitute a local memory population which is very stable, behaving like “anchor” points to provide the best solutions found so far. Such a mutation strategy may be very helpful to deal with the problem in this paper.

3.4.3. Parametric adaptions

Apart from mutation strategy, adaptions of the control parameters which are F and Cr also play crucial role on performance of DE algorithms. To reduce computational burden and avoid time-consuming adaptions of F and Cr, the authors adopted a adaption as follows:

$$F_{i,G+1} = \begin{cases} F_l + r_1(F_u - F_l), & \text{if } r_2 < \tau_1 \\ F_{i,G+1}, & \text{otherwise} \end{cases} \tag{15}$$

$$Cr_{i,G+1} = \begin{cases} r_3 Cr_u, & \text{if } r_4 < \tau_2 \\ Cr_{i,G}, & \text{otherwise} \end{cases} \tag{16}$$

Where, r_1, r_2, r_3, and r_4 are random numbers which obey uniform

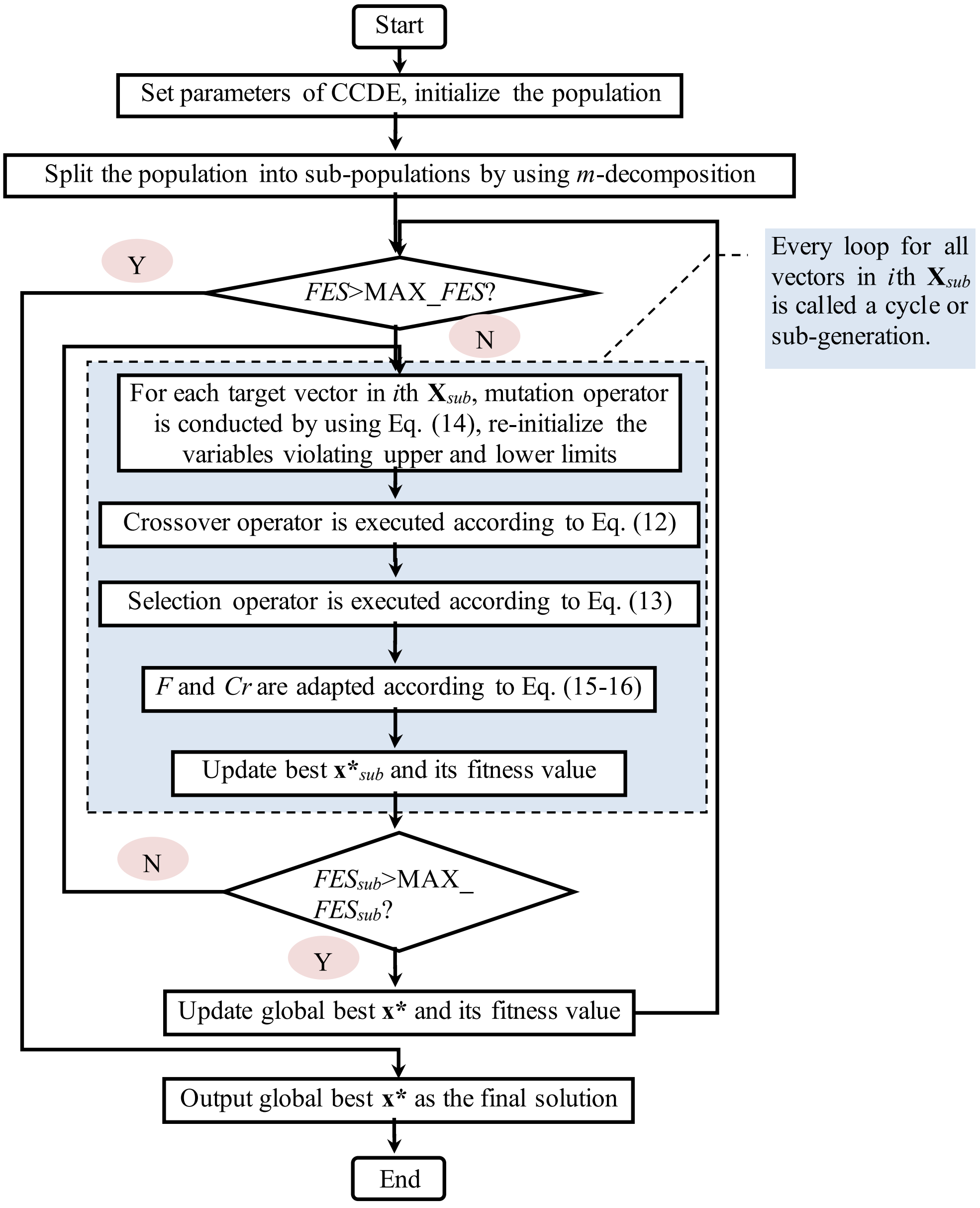

Start
Set parameters of CCDE, initialize the population
Split the population into sub-populations by using m-decomposition
Y
FES>MAX_FES?
N
Every loop for all vectors in ith $\mathbf{X}_{sub}$ is called a cycle or sub-generation.
For each target vector in ith $\mathbf{X}_{sub}$, mutation operator is conducted by using Eq. (14), re-initialize the variables violating upper and lower limits
Crossover operator is executed according to Eq. (12)
Selection operator is executed according to Eq. (13)
F and Cr are adapted according to Eq. (15-16)
Update best $\mathbf{x}^*_{sub}$ and its fitness value
N
FES_{sub}>MAX_ FES_{sub}?
Y
Update global best $\mathbf{x}^*$ and its fitness value
Output global best $\mathbf{x}^*$ as the final solution
End

Fig. 5. The flowchart of CCDE algorithm.

distribution within [0, 1]. τ_1 and τ_2 are two control parameters which are both set to 0.1. F_u and F_l are set to 0.9 and 0.1, respectively. Different from the adaption of Cr in jDE (Brest, Greiner, Bokovi, Mernik, & Zumer, 2006), $Cr_u = 0.5$ was adopted to limit Cr within [0, 0.5], reinforcing the local search capability of the proposed method. In fact, the reason why we have designed such a parametric adaption for F and Cr is that, this mechanism is simple but effective. Besides, the settings of τ_1 and τ_2 ($\tau_1 = \tau_2 = 0.1$) have been well proven and widely used. The problem of identifying parameters of battery has been transformed into a LSOP, so it is required a large MAX_FES to solve this problem. We should try to reduce total time cost for running simulations, without developing complex parametric adaptions.

3.4.4. General framework

In summary, by incorporating the m-decomposition method and new DE algorithm, the algorithmic description of the proposed CCDE for parameters identification of battery is shown as Algorithm 3.

Algorithm 3. The pseudo code of CCDE.

(continued on next column)

(continued)

Algorithm 3. The pseudo code of CCDE.

```
Input: $NP$, MAX_$FES$, MAX_$FES_{sub}$, $D$, $\mathbf{X}_{min}$, $\mathbf{X}_{max}$, $m$
Output: fit($\mathbf{x}^*$), $\mathbf{x}^*$
[fit($\mathbf{X}$), $\mathbf{X}$] ← initialization($NP$, $D$, $\mathbf{X}_{min}$, $\mathbf{X}_{max}$);
[fit($\mathbf{x}^*$), $\mathbf{x}^*$] ← sort(fit($\mathbf{X}$), $\mathbf{X}$);
$N \leftarrow m$, $G \leftarrow 1$;
[$\mathbf{X}_{sub}$, $fit_{sub}$] ← $m$-decomposition($\mathbf{X}$, $m$, $D$); % $\mathbf{X}_{sub} = \{\mathbf{X}_{sub,1}, \ldots, \mathbf{X}_{sub,n}, \ldots, \mathbf{X}_{sub,N}\}$
$FES \leftarrow NP + NP^*N$;
while $FES \leq$ MAX_$FES$
    for $n \leftarrow 1$ to $N$
        [$\mathbf{x}^*_{sub,n}$, $\mathbf{X}_{sub,n}$] ← update_sub_best($\mathbf{x}^*$, $\mathbf{X}_{sub,n}$);
        $FES \leftarrow FES + 1$, $FES_{sub} \leftarrow 0$;
        while $FES_{sub} \leq$ MAX_$FES_{sub}$
            for $i \leftarrow 1$ to $NP$
                [$\mathbf{V}_{sub,n,i,G}$] ← new_mutation($\mathbf{X}_{sub,n,i,G}$); % Eq. (14)
                for $j \leftarrow 1$ to $m + 3$
                    if $v^j_{sub,n,i,G} > x^j_{max}$ || $v^j_{sub,n,i,G} < x^j_{min}$
                        [$v^j_{sub,n,i,G}$] ← re-initialize($x^j_{min}$, $x^j_{max}$);
                    end
                end
```

(continued on next page)

(*continued*)

Algorithm 3. The pseudo code of CCDE.

```
18 [$\mathbf{U}_{sub,n,i,G}$] ← crossover($\mathbf{V}_{sub,n,i,G}$);
19 [$\mathbf{X}_{sub,n,i,G+1}$] ← selection($\mathbf{U}_{sub,n,i,G}$, $\mathbf{X}_{sub,n,i,G}$);
20 $FES_{sub} \leftarrow FES_{sub} + 1$, $FES \leftarrow FES + 1$;
21 [$F_{sub,n,i,G+1}$, $Cr_{sub,n,i,G+1}$] ← adaption($F_{sub,n,i,G}$, $Cr_{sub,n,i,G}$); % Eq. (15–16)
22 [fit($\mathbf{x}^*_{sub,n}$), $\mathbf{x}^*_{sub,n}$] ← min($\mathbf{x}^*_{sub,n}$, $\mathbf{X}_{sub,n,i,G+1}$);
23 end
24 end
25 [$\mathbf{x}^*$] ← update_global_best($\mathbf{x}^*$, $\mathbf{x}^*_{sub,n}$);
26 $FES \leftarrow FES + 1$;
27 $G = G + 1$;
28 end
29 end
```

It is noted that, there are some important modifications in Algorithm 3 compared with other CC algorithms. First, in line 4, a new *m*-decomposition method for solving parameters identification of battery was proposed. The presented grouping method divided the whole population into *N* sub-population with the three parameters (R_0, R_P and C_P) overlapped in each sub-population. This is a targeted improvement. Second, in line 12, the proposed mutation strategy which was designed to improve the exploitation capability was used. Third, in line 21, *F* and *Cr* are self-adapted to speed up convergence. Fourth, in line 8 and 25, an update operator was developed to renew global best individual $\mathbf{x}^*$. When a $\mathbf{x}^*_{sub,n}$ with better fitness value is found, then the corresponding and overlapped part in $\mathbf{x}^*$ should be updated. Note that, in line 25, partial variables in $\mathbf{x}^*$ will be replaced by $\mathbf{x}^*_{sub,n}$ after each cycle/sub-generation, resulting in possibly deteriorated fitness value of $\mathbf{x}^*$ with full length. It is normal and unavoidable when a CC methodology is used. For general CC algorithms, all the sub-populations are not overlapped among each other. In this paper, considering the importance of the three parameters (R_0, R_P and C_P) while estimating v_{OCV}, the three parameters were designed to renew $\mathbf{x}^*_{sub,n}$ and $\mathbf{x}^*$ when calling update_sub_best(·) and update_global_best(·), respectively. That is to say, in every cycle/sub-generation, R_0, R_P and C_P, as a part of $\mathbf{x}^*_{sub,n}$, will be continuously optimized with v_{OCV} at different sampling points. That is the reason why the proposed operator is different from other CC algorithms for solving general LSOPs.

For the convenience of understanding, we drew general the framework of the proposed CCDE as shown in Fig. 5. In the next section, we will introduce the steps on how to apply this algorithm for identifying parameters of battery.

3.4.5. *CCDE applied for identifying parameters of battery*

In this part, the authors illustrate the steps of how to identify parameters of battery by using the proposed CCDE algorithm. The procedures are listed below:

Step1 Load dynamic test data. Measured data from different dynamic tests (DST/FUDS) is loaded. Specifically, under a given ambient temperature *T*, load current $i_L(k)$ and terminal voltage $v_{term}(k)$ are prepared for optimizing, where $k = 1, 2, \ldots, k_{max}$. $i_L(k)$ and $v_{term}(k)$ are sampled at a real interval in dynamic tests. The data of initial charging and discharging are eliminated, and the abnormal data points are also cleaned.

Step2 Set simulation parameters. *D* is set to $k_{max} + 3$. Considering the value of *D*, we set simulation parameters, such as MAX_*FES*, *NP* and the number of independent runs. To make a balance between computational burden and accuracy, *m* and the number of sub-populations *N* are set. Based on the value of *m*, MAX_FES_{sub} is set.

Step3 Run simulation. Each single simulation is repeated for some independent runs. For each run, R_0, R_P and C_P are randomly initialized as positive values, and $v_{OCV}(k)$ is randomly initialized within $[V_L, V_U]$. By inputting simulation parameters in Step2 into CCDE, for minimizing the objective fitness value of Eq. (3), optimal parameters of parameters and $v_{OCV}(k)$ will be outputted. After finishing all simulations, the statistical results of fitness values and the best solutions ($\mathbf{x}^*$) are summarized.

Table 2
Basic specifications of INR 18650-20R battery.

Parameters	Values
Capacity Rating	2000 mAh
Cell Chemistry	LNMC/Graphite
Weight	45.0 g
Dimensions	18.33 ± 0.07 mm
Length	64.85 ± 0.15 mm
Nominal voltage	3.6 V
Upper/lower cut-off voltage	4.2 V/2.5 V

4. Experimental results

This section reports the experimental results of the proposed CCDE and compared algorithms on parameters identification of battery. Section 4.1 gives experimental datasets, settings and involved algorithms. Section 4.2 conducts experiments to test the performance of CCDE with different *m* values. Based on the experimental results, *m* value and the corresponding MAX_*FES* was set for parameter identification of LIB. Section 4.3 conducts experiments to show the influence of each component on the performance of CCDE. Section 4.4 comprehensively tests the performance of CCDE compared with 7 state-of-the-art CC algorithms on DST and FUDS data for identifying parameters of LIB. Section 4.5 reports experimental results of CCDE and 7 classic DE algorithms. 4.1 Experimental datasets, settings and involved algorithms.

All test data in this paper comes from battery research group of Center for Advanced Life Cycle Engineering (CALCE) in University of Maryland. Related datasets can be downloaded by visiting https://web.calce.umd.edu/batteries/data.htm. The INR 18650-20R cell was chosen as the test target. All the tests were conducted at 25 °C. Basic descriptions on this battery are shown in Table 2.

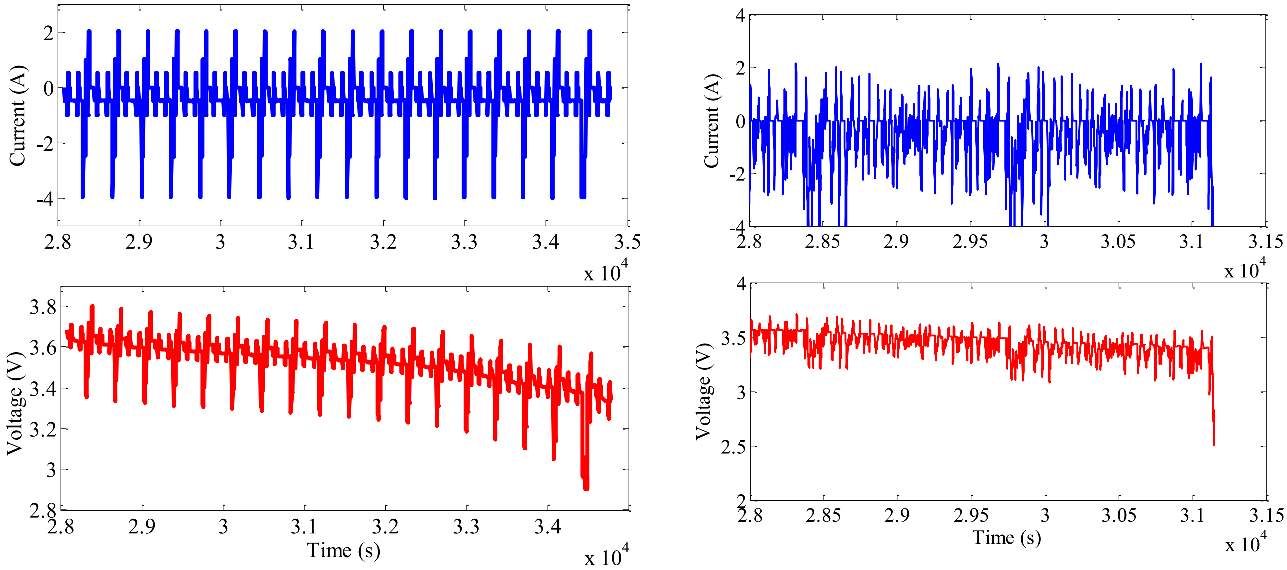
Dynamic test profiles like dynamic stress test (DST) is the most commonly used test procedure were used to identifying parameters of battery and to validate SOC. DST simulates a dynamic discharge process which is a typical driving cycle that is often used to evaluate various battery models and SOC estimation algorithms. two test profiles for a battery at 25 °C is shown in Fig. 6(a). Apart from DST, some more complicated dynamic current profiles are also widely used to evaluate the SOC estimation results, such as federal urban dynamic schedule (FUDS) (Duong, 2000) whose dynamic profile is shown in Fig. 6(b). FUDS test is more sophisticated than DST in terms of the changing/discharging rate of the current. All datasets of the dynamic tests can be downloaded via https://web.calce.umd.edu/batteries/data.htm.

In this work, a Windows 7 operating system and the MATLAB 2010b development environment are used as the simulation environment. The hardware platform is a PC with the following features: Intel(R) Core (TM) i5-3230 M CPU @ 2.60 GHz 2.60 GHz, 8.00 GB RAM.

It is worth noting that, different from evaluating algorithms on benchmark functions, some indicators, such as successful rate (SR) and minimum *FES* to find $\mathbf{x}^*$, could not be used or compared for evaluating methods on identifying parameters of battery in this paper. This is because for a real-world problem, we have no prior knowledge of exact values of $\mathbf{x}^*$. For evaluating an algorithm on identifying parameters of battery, accuracy of fitness value is a primary criterion. Also, convergence speed and time complexity are required to evaluating the involved algorithms in general.

4.1. *Experiments on CCDE with different m values*

In this part, we conduct experiments to test the proposed CCDE with different *m* values, i.e., we test the sensitivity of *m* on CCDE. After that, the *m* value with the best performance was chosen for other experiments. In this paper, the authors believe that *m* significantly affect the performance of CCDE. Other simulation parameters are strongly related



(a) DST

(b) FUDS

Fig. 6. Battery test profiles for 50% battery level at 25 °C.

Table 3
CCDE algorithms with different *m* values and corresponding parametric settings.

Algorithms	*m*	*NP*	Fitness evaluations
CCDE1	50	300	MAX_FES_{sub} = 600000, MAX_FES = 6.00E + 07
CCDE2	50	200	MAX_FES_{sub} = 500000, MAX_FES = 5.00E + 07
CCDE3	110	100	MAX_FES_{sub} = 80000, MAX_FES = 1.76E + 07
CCDE4	110	60	MAX_FES_{sub} = 40000, MAX_FES = 8.80E + 06
CCDE5	330	50	MAX_FES_{sub} = 15000, MAX_FES = 9.90E + 06
CCDE6	330	30	MAX_FES_{sub} = 9000, MAX_FES = 5.94E + 06
CCDE7	665	50	MAX_FES_{sub} = 10000, MAX_FES = 1.33E + 07
CCDE	665	20	MAX_FES_{sub} = 4000, MAX_FES = 5.32E + 06

with *m*, such as D_{sub}, *NP*, MAX_FES_{sub}, and MAX_FES. For instance, k_{max} is 6650 in DST data for 50% SOC level at 25 °C. When *m* is set to 665, i.e., the whole population is divided into 665 sub-populations, D_{sub} for each sub-population is consequently 13 (R_0, R_P, C_P and 10 v_{OCV} values). Based on D_{sub} and the character of parameters identification for battery, *NP* is set to 20 and MAX_FES_{sub} is set to 4000. Each sub-population should be optimized at least twice, so MAX_FES = MAX_FES_{sub}*m*2 = 5320000. When *m* is set to 50, D_{sub} is consequently 136. As a result, *NP* should be set to at least 200, and MAX_FES_{sub} should be set to at least 500000. And MAX_FES should be set to 50000000. Obviously, less groups with bigger size of sub-population will significantly increase computational cost and difficulty. Therefore, we adopted the *m* value which represents more groups with smaller size of sub-population. Nevertheless, extra experiments are required to make this setting more reliable. Table 3 shows CCDE with different parametric settings. *NP* which should be bigger than D_{sub} is conventionally set to [3*D, 5*D] for each sub-problem. In Table 3, a criterion of setting parameters was followed. The criterion is, bigger D_{sub} requires much bigger *NP* and MAX_FES_{sub} to make sure sub-populations converge.

To test the performances of CCDEs with different *m* values, DST and FUDS datasets were used. The comprehensive results of all involved algorithms over 31 independent runs are reported in Table 4. Besides, we also conducted Wilcoxon rank sum test at a 0.05 significance level. For the involved algorithms, the same operators and control parameters were used for fair comparison. The best results with the minimum value in Table 4 are shown in bold.

From the results reported in Table 4, we found that CCDE with small sub-populations outperformed CCDE with other settings. In fact, it is not surprising to obtain such a conclusion. For DST dataset, CCDE7 performed similar to CCDE. For FUDS dataset, CCDE6 and CCDE7 also performed similar to CCDE. Moreover, the statistical results displayed a trend. That is, as *m* decreased, the performance of CCDE deteriorated obviously. It is an also an evidence that it is much difficult to optimize the sub-problem with bigger D_{sub}. Therefore, in this work, we set *m* to a relatively big value to reduce the size of sub-population. In summary, the presented experimental results demonstrated that the settings of CCDE in the following sections are reasonable.

4.2. Influence of modified mechanisms on CCDE

Before test the performance of the proposed CCDE compared with other state-of-the-art algorithms, we would like to conduct some

Table 4
Mean and variance of fitness values of CCDE with different simulation settings.

Algorithms	DST data				FUDS data			
	Mean	Std	W-test	*p*-value	Mean	Std	W-test	*p*-value
CCDE1	1.41E + 03	1.48E + 02	+	7.05E-07	1.91E + 03	1.95E + 03	+	4.25E-03
CCDE2	1.47E + 03	7.90E + 02	+	7.05E-07	1.97E + 03	1.58E + 03	+	8.00E-03
CCDE3	1.29E + 03	1.24E + 03	+	1.21E-03	1.74E + 03	1.74E + 03	+	8.85E-03
CCDE4	1.24E + 03	1.25E + 03	+	1.53E-03	1.74E + 03	1.75E + 03	+	1.05E-02
CCDE5	1.11E + 03	1.08E + 03	+	5.27E-03	1.69E + 03	1.68E + 03	+	2.86E-02
CCDE6	9.57E + 02	9.55E + 02	+	8.85E-03	1.53E + 03	1.56E + 03	≈	7.90E-02
CCDE7	6.15E + 02	4.24E + 02	≈	1.68E-01	1.21E + 03	1.22E + 03	≈	2.86E-01
CCDE	**3.59E + 02**	**3.39E + 02**	–	–	**9.67E + 02**	**1.00E + 03**	–	–

Table 5
Mean and variance of fitness values of CCDE with different components.

Algorithms	DST data				FUDS data			
	Mean	Std	W-test	*p*-value	Mean	Std	W-test	*p*-value
CCDE-P1	5.97E + 02	4.48E + 02	+	6.38E-05	1.54E + 03	1.32E + 03	+	1.87E-02
CCDE-P2	5.55E + 02	4.39E + 02	+	1.17E-02	1.27E + 03	1.26E + 03	+	2.17E-02
CCDE-P3	6.07E + 02	5.63E + 02	+	3.98E-02	1.57E + 03	1.36E + 03	+	1.54E-03
CCDE-P4	4.26E + 02	3.96E + 02	≈	2.97E-01	1.15E + 03	1.16E + 03	≈	6.83E-02
CCDE-P5	5.02E + 02	4.37E + 02	+	3.59E-02	1.34E + 03	1.26E + 03	+	9.98E-03
CCDE-P6	5.89E + 02	4.01E + 02	+	1.17E-02	1.46E + 03	1.22E + 03	+	4.32E-04
CCDE-P7	4.38E + 02	3.77E + 02	≈	1.05E-01	1.09E + 03	**9.97E + 02**	≈	5.74E-02
CCDE	**3.59E + 02**	**3.39E + 02**	–	–	**9.67E + 02**	1.00E + 03	–	–

experiments to test the influence of the developed mechanisms on the performance of CCDE. In Section 3.4, three modified mechanisms have been developed. The influence of *m*-decomposition method with different settings (*m*, *NP* and MAX_*FES*) on the performance of CCDE has been tested in Section 4.2. Therefore, CCDE with different mutation strategies, population topology and parametric adaptions will be tested in this section. CCDE-P1 denotes the proposed CCDE algorithm which adopts DE/rand/1 mutation strategy (Eq. (7)) and a full-connected/*g*-best population topology. This topology means every individual could exchange information with any other individual in the whole population. CCDE-P2 represents the proposed CCDE which uses DE/rand/1 mutation strategy and a ring topology. CCDE-P3 is CCDE adopting DE/current-to-best/1 mutation strategy (Eq. (9)) and a full-connected population topology. CCDE-P4 is the proposed CCDE using DE/current-to-best/1 mutation strategy and a ring topology. The above four variants of CCDE use the proposed parameter adaptions as shown in Eq. (15–16). That means, we try to test the influence of mutation strategy and topology on the performance of CCDE. CCDE-P5 denotes the proposed CCDE which uses a fixed parametric setting: $F = 0.5$, $Cr = 0.9$. CCDE-P6 represents the proposed CCDE which adopts a random parametric setting for *F* and *Cr*. That is to say, at every iteration, *F* and *Cr* are randomized by using a generator which obeys a uniform distribution within [0, 1]. CCDE-P7 is the variant of CCDE which uses the parametric adaption in jDE: $F_l = 0.1$, $F_u = 0.9$, $Cr_l = 0$, $Cr_u = 1$, $\tau_1 = \tau_2 = 0.1$. The above three variants of CCDE do not modify the mutation strategy and population topology. That means the influence of the proposed parametric adaptions will be tested. For DST data, $k_{max} = 6650$, $D_{sub} = 13$, MAX_$FES_{sub} = 4000$, $NP = 20$, MAX_$FES = 5.32$E + 06. For FUDS data, $k_{max} = 6946$, $D_{sub} = 14$, MAX_$FES_{sub} = 4000$, $NP = 20$, MAX_$FES = 5.55$E + 06. The number of independent runs is set to 31. We also conducted Wilcoxon rank sum test at a 0.05 significance level. Experimental results of the above CCDE variants on DST and FUDS data are shown in Table 5.

From the results reported in Table 5, we may conclude that the proposed mechanisms do affect the final performance of CCDE. First, based on both statistical and hypothetical results, these CCDE variants performed similarly, showing some common patterns. Second, mutation strategy coupled with population topology has a more significant impact on parameter adaption. CCDE variants with other mutation strategies and topologies (CCDE-P1 to CCDE-P4) performed slightly worse than CCDE variants with other parametric settings (CCDE-P5 to CCDE-P7) in general. Thus, we may conclude that mutation strategy is more important to some extent. Third, for identifying parameters of LIB, a ring topology seemed better than a full-connected topology. This is based on the observations: CCDE-P2 performed better than CCDE-P1, and CCDE-P4 performed better than CCDE-P3. Fourth, parametric adaptions also affected the performance of CCDE. From the experimental results we can see, self-adaptive tunning for *F* and *Cr* was better than fixed values. Additionally, there was no significant difference between the adaption from jDE and the proposed parametric adaption. This is because the tunning mechanisms of the two adaptions are similar. Overall, by incorporating the proposed mutation strategy, population topology and parametric adaptions, the presented CCDE has demonstrated its effectiveness on parameters identification of LIB.

4.3. *Experimental results on DST and FUDS data*

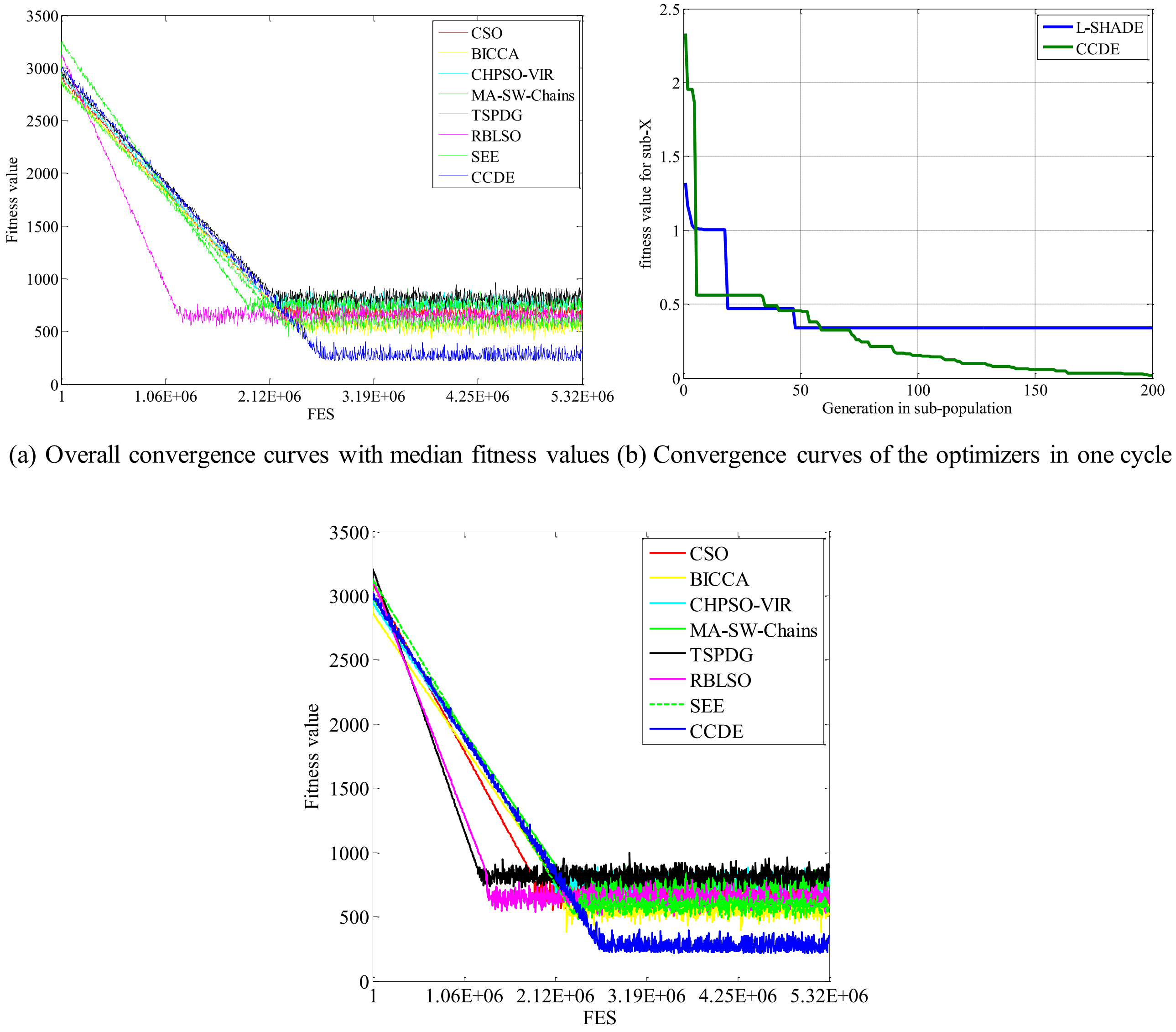
In order to show effectiveness of the proposed CCDE method on parameters identification of battery, seven state-of-the-art CC algorithms were chosen for comparison. In this section, the compared algorithms are CSO (Cheng & Jin, 2015), BICCA (Ge et al., 2020), CHPSO-VIR (Ge et al., 2017), MA-SW-Chains (Molina, Lozano, Sanchez, & Herrera, 2011), TSPDG (Xue et al., 2020), RBLSO (Deng, Peng, Zhang, Yang, & Chen, 2019) and SEE (Yang, Tang, & Yao, 2018). CSO utilized a pairwise competitive strategy for the whole swarm. BICCA proposed a generalized framework with evolutions in the pattern space and search space. CHPSO-VIR firstly developed a strategy which used a small part of variable interactions to approximate the decomposition. MA-SW-Chains defined a local search indicator for each individual to guide the evolution. TSPDG employed variable interactions and population topology into the grouping process to improve the efficiency of optimization. RBLSO uses a principle of maximizing the fitness difference between learners and exemplars to improve the performance of the optimization algorithms. SEE used *meta*-models to find a better solution for a problem. Following the recommends of BICCA (Ge et al., 2020), L-SHADE (Tanabe & Fukunaga, 2014) was selected as the optimizer for these competitors. In this part, we followed the selections of the compared CC algorithms and the optimizer in BICCA (Ge et al., 2020). The parametric settings of the above methods were used as claimed in their original papers. The proposed CCDE used the modified version of DE algorithm as shown in Section 3.4. Based on the experimental results reported in Section 4.2, the values of *m*, *NP* and MAX_*FES* were set. Specifically, for DST data, $k_{max} = 6650$, $D_{sub} = 13$, MAX_$FES_{sub} = 4000$, $NP = 20$, MAX_$FES = 5.32$E

Table 6
Statistical results of fitness values and hypothesis tests on DST data.

Method	Min	Med	Max	Mean	Std	Wilcoxon-test	*p*-value
CSO	4.30E + 01	7.28E + 02	2.08E + 03	6.71E + 02	5.05E + 02	+	5.34E-04
BICCA	6.50E + 01	8.14E + 02	2.22E + 03	5.48E + 02	8.37E + 02	+	1.61E-03
CHPSO-VIR	3.69E + 01	8.21E + 02	2.38E + 03	7.70E + 02	7.88E + 02	+	6.24E-04
MA-SW-Chains	4.63E + 01	6.40E + 02	1.92E + 03	7.39E + 02	6.17E + 02	+	1.20E-03
TSPDG	4.82E + 01	8.22E + 02	1.76E + 03	8.08E + 02	4.66E + 02	+	5.66E-05
RBLSO	5.42E + 01	4.93E + 02	2.13E + 03	6.35E + 02	6.99E + 02	+	3.72E-02
SEE	4.62E + 01	4.50E + 02	2.27E + 03	5.94E + 02	6.42E + 02	≈	1.77E-01
CCDE	**3.57E + 01**	**3.71E + 02**	**7.72E + 02**	**3.48E + 02**	**2.11E + 02**	–	–

Table 7
Best partial optimal solutions solved by the competitive algorithms on DST data.

Method	$R_0(\Omega)$	$R_P(\Omega)$	C_P(F)	v_{OCV}(1)V	v_{OCV}(666)V	v_{OCV}(1331)V	v_{OCV}(3326)V	v_{OCV}(4656)V	v_{OCV}(6650)V
CSO	0.0689	0.0243	1184.69	3.8759	3.7519	3.7482	3.6974	3.5780	3.4814
BICCA	0.0637	0.0224	1231.28	3.8510	3.6694	3.6244	3.5438	3.4807	3.4559
CHPSO-VIR	0.0712	0.0247	1155.40	3.7249	3.6410	3.5256	3.4865	3.4114	3.4017
MA-SW-Chains	0.0732	0.0232	1226.15	3.8809	3.7316	3.6439	3.5175	3.4955	3.4486
TSPDG	0.0623	0.0244	1063.67	3.8151	3.7643	3.6524	3.6125	3.5634	3.4797
RBLSO	0.0587	0.0242	1207.05	3.8267	3.8253	3.6804	3.5983	3.5511	3.4317
SEE	0.0679	0.0259	1239.29	3.7751	3.6364	3.6108	3.5501	3.5114	3.4684
CCDE	0.0705	0.0269	1201.41	3.6889	3.6848	3.6680	3.5825	3.5059	3.4563



(a) Overall convergence curves with median fitness values (b) Convergence curves of the optimizers in one cycle

(c) Convergence curves with averaged fitness values over all the independent runs

Fig. 7. Convergence curves of compared algorithms on DST dataset.

+ 06. For FUDS data, $k_{max} = 6946$, $D_{sub} = 14$, MAX_$FES_{sub} = 4000$, $NP = 20$, MAX_FES = 5.55E + 06. All the simulations were repeated for 31 independent runs. The performance of one involved algorithm is evaluated based on the statistical results over the 31 independent runs.

4.3.1. *Experimental results on DST data*

First, DST data for 50% SOC level was used to test performance of the proposed method. In initial part of DST test, the battery was charged fully with 1A to 100% SOC, then the cell was discharged with 1A to 50% SOC. After that, dynamic stress test started. The dynamic part of this test procedure lasted from 28000 s to 34800 s. During the dynamic test procedure, k_{max} is 6650. For CCDE, m was set to 665 to build up a relatively small sub-population for convenience. Thus, NP was set to 20 in this experiment. MAX_FES_{sub} was set to 20*NP = 4000. The

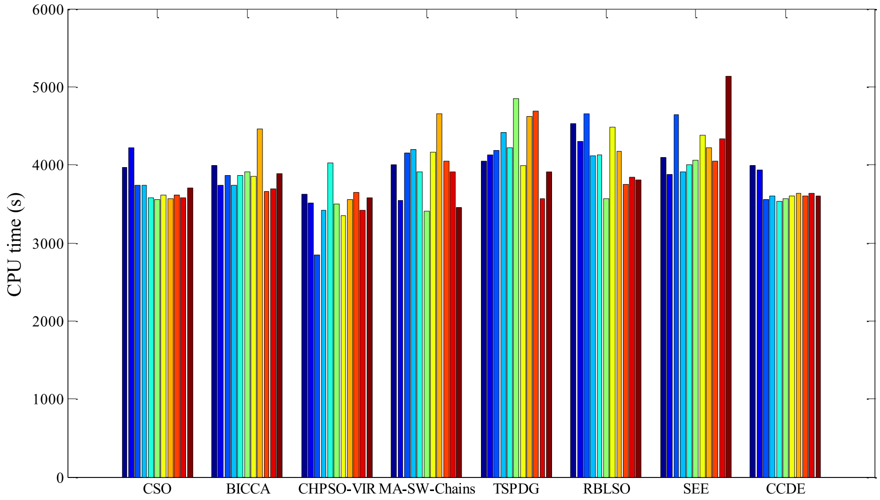


Fig. 8. CPU time of the algorithms over partial independent runs (11 out of 31) on DST dataset.

comprehensive results of all involved algorithms over 31 independent runs are reported in Table 6. Min denotes the minimum fitness value. Med represents the median value over 31 runs. Max denotes the maximum fitness value. Mean and std are the average and standard deviation of the 31 fitness values, respectively. The minimum values in each column are shown in bold. To show the significant difference between CCDE and the corresponding compared algorithm, we also conducted Wilcoxon rank sum test at a 0.05 significance level. Sign "+" indicates CCDE is significantly better than the peer competitor. Sign "-" means CCDE is significantly worse than the peer competitor. Sign "$\approx$" denotes that the two algorithms perform similarly.

From Table 6 we can see, the proposed CCDE outperformed other competitors in general. In each column, CCDE obtained the minimum value among the involved algorithms. From the results of Wilcoxon tests, SEE obtained similar results compared with CCDE. Whereas, the average fitness value of CCDE is better than SEE. This is partially because that the number of independent runs is relatively small. Further, the average error obtained by CCDE is 3.48E + 02. Considering k_{max} in DST dataset is up to 6650, the average error between measured and estimated terminal voltages on each sampling point is about 0.05 V. It is acceptable for the proposed methodology to meet such an accuracy with a very big dimension (D = 6653). The overall results in Table 6 demonstrated that the proposed m-decomposition method and optimizer are effective for identifying parameters of battery. Besides, the optimal solutions optimized by all the algorithms are shown in Table 7.

Table 7 reports different optimal solutions solved by the compared algorithms. The distribution ranges of R_0, R_P and C_P are roughly the same. Whereas, for $v_{OCV}(k)$, the solutions varies obviously. This is because slight difference of the three parameters (R_0, R_P and C_P) will generate relatively significant variances on v_{OCV}. In general, the parameters of the best solutions found by the algorithms are within reasonable ranges. To show the convergence performances, the convergence curves of the involved methods with median fitness values were plotted as shown in Fig. 7.

In Fig. 7(a), convergence curves with median fitness values over 31 independent runs were plotted. MAX_FES = MAX_FES_{sub}*m*2 = 4000*665*2 = 5.32E + 06. This setting means that there are 665 cycles/sub-generations which constitute a main loop, and the main loop is executed twice. It is worth mentioning that, the convergence curves in Fig. 7(a) are different from other convergence curves on general optimization problems. Usually, convergence curves will not fluctuate, because the global best fitness value recorded during evolving never deteriorates for a minimum optimization problem. Whereas, for parameters identification of battery in this paper, the global best fitness value may be bigger than the value in the last generation due to the updating mechanism of **x*** as shown in line 25 in Algorithm 3. We found that all the compared algorithms showed their good exploration capabilities. Nevertheless, the proposed CCDE performed better than other methods on convergence. In the later stage, all the algorithms met stagnation. In Fig. 7(b), we plotted two convergence curves of CCDE and L-SHADE in one cycle which is a sub-generation. Within a cycle, the maximum FES is MAX_FES_{sub} which is set to 4000 in this paper. Considering NP is set to 20, the maximum number of generations in a cycle is 200. In our experiments, all the compared CC algorithm all chose L-SHADE as the optimizer, so we compared CCDE with L-SHADE on convergence performance. From Fig. 7(b) we can see, the proposed CCDE could continuously improve the fitness value during a cycle, but L-SHADE met a stagnation in the later stage. This demonstrated that the proposed mutation strategy and parametric adaption are effective for identifying parameters of battery. Furthermore, the fitness value found by CCDE after a cycle is about 0.02 V. Considering in each sub-population D_{sub} is set to 13, which is consisted of R_0, R_P, C_P and 10 v_{OCV} points, the average error of v_{term} and $\widehat{v}_{term}$ over the 10 sampling points is about 0.002 V. Such an accuracy is quite acceptable for an optimizer. Besides, it was also demonstrated that the parametric settings of NP and MAX_FES_{sub} are suitable for solving this problem. In Fig. 7(c), we plotted the convergence curves which denote the average fitness values over all the independent runs. From this figure we can see, the averaged convergence curves look similar to the convergence curves with the median fitness values, but the averaged convergence curves fluctuated more violently. Also, this comparison on averaged convergence curves demonstrated that the proposed methodology could find better solutions on average. In summary, the proposed CCDE outperformed other methods on convergence.

Further, comparisons of the involved algorithms on time cost are shown in Fig. 8. We recorded CPU time cost of each algorithm on randomly selected 11 out of 31 independent runs for showing comparisons clearly. From the bar chart, significant differences among the algorithms may not exist. For most runs, the average time cost is about 3500 s. That means the performance of CCDE on time complexity seemed similar to other state-of-the-art CC algorithms. Note that, the total consuming time of different algorithms vary in different simulation platforms, but it does not affect the comparisons among different algorithms.

4.3.2. *Experimental results on FUDS data*

In the section, FUDS data for 50% SOC level was used to test performance of the proposed method. In initial part of FUDS test, the battery was charged fully with 1A to 100% SOC, then the cell rested for about 7200 s. After that, the battery was discharged with 1A to 50%

Table 8
Statistical results of fitness values and hypothesis tests on FUDS data.

Method	Min	Med	Max	Mean	Std	Wilcoxon-test	*p*-value
CSO	9.81E + 01	1.69E + 03	5.53E + 03	1.43E + 03	1.38E + 03	+	7.47E-03
BICCA	1.39E + 02	1.94E + 03	5.37E + 03	1.31E + 03	1.71E + 03	+	4.46E-03
CHPSO-VIR	1.01E + 02	1.57E + 03	3.58E + 03	1.33E + 03	1.55E + 03	+	1.49E-02
MA-SW-Chains	2.58E + 02	1.90E + 03	4.67E + 03	1.60E + 03	1.37E + 03	+	8.13E-03
TSPDG	1.24E + 02	1.49E + 03	3.50E + 03	1.49E + 03	1.34E + 03	+	2.43E-02
RBLSO	1.94E + 02	1.49E + 03	4.30E + 03	1.44E + 03	1.47E + 03	+	4.26E-03
SEE	**9.20E + 01**	1.55E + 03	3.50E + 03	1.44E + 03	1.35E + 03	+	2.02E-02
CCDE	9.47E + 01	**8.82E + 02**	**2.20E + 03**	**9.36E + 02**	**9.58E + 02**	–	–

Table 9
Best partial optimal solutions solved by the competitive algorithms on FUDS data.

Method	R_0(Ω)	R_P(Ω)	C_P(F)	v_{OCV}(1)V	v_{OCV}(601)V	v_{OCV}(1201)V	v_{OCV}(3001)V	v_{OCV}(6001)V	v_{OCV}(6601)V
CSO	0.0315	0.0504	1518.34	3.6425	3.6562	3.6209	3.6452	3.4563	3.4361
BICCA	0.0754	0.0382	899.22	3.6844	3.6479	3.6379	3.6052	3.4546	3.4303
CHPSO-VIR	0.0511	0.0885	1174.42	3.6612	3.6733	3.6449	3.6733	3.4774	3.4496
MA-SW-Chains	0.0367	0.0222	1923.34	3.6472	3.6448	3.6125	3.7097	3.4445	3.4265
TSPDG	0.0800	0.0519	1594.956	3.6883	3.6573	3.7545	3.6210	3.4630	3.5368
RBLSO	0.0640	0.0533	1666.57	3.6732	3.6584	3.8073	3.6303	3.4722	3.5375
SEE	0.0130	0.0222	1219.49	3.6251	3.7034	3.6015	3.6888	3.4416	3.4957
CCDE	0.0505	0.0569	888.41	3.6609	3.6552	3.6332	3.6429	3.4591	3.4667

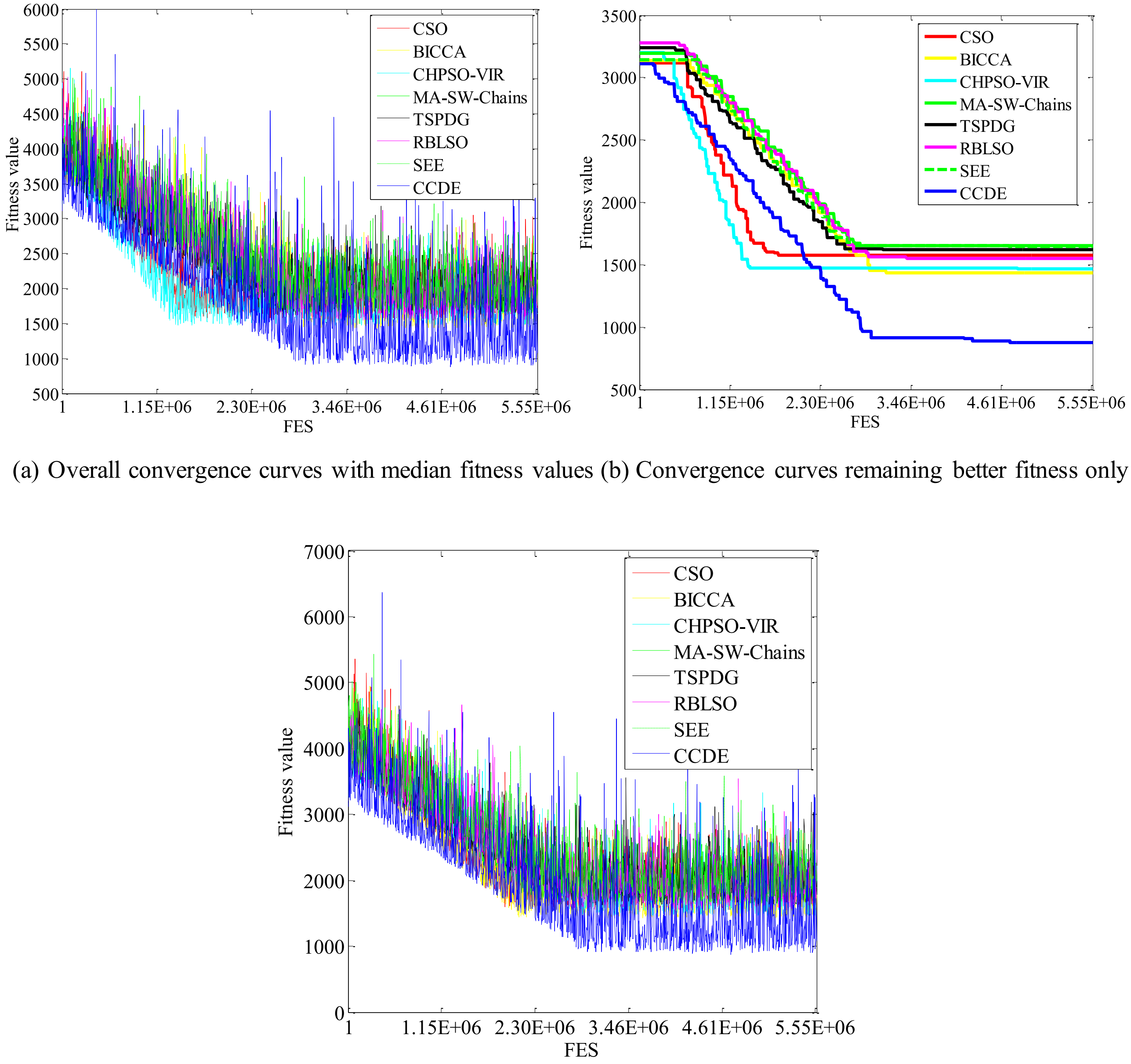
SOC, then the battery rested for about 7100 s again. After initialization, current began to fluctuate from 24086 s to 31148 s. Excluding the data points whose voltage are below 3 V, there are 6946 data points during the dynamic test procedure. That is to say, k_{max} is 6946. Compared with DST data, k_{max} in FUDS data is slightly bigger. In this part, m was set to 694 to build up a relatively small sub-population for convenience. It is noted that, by using this FUDS dataset, D_{sub} in each sup-population is not always the same, because k_{max} is not exactly divided by m with no reminder. For most sub-populations, D_{sub} is 13. For a few sub-populations, D_{sub} is 14. Nevertheless, there is no influences on the performances of m-decomposition and CCDE with D_{sub} set to 13 or 14. Thus, *NP* was still set to 20 in this experiment. MAX_FES_{sub} was set to 200**NP* = 4000. MAX_*FES* = MAX_FES_{sub}*m*2 = 4000*694*2 = 5.55E + 06. Also, each algorithm was run over 31 times on FUDS dataset. The comprehensive results are shown in Table 8. The minimum values in each column are shown in bold. Wilcoxon rank sum test at a 0.05 significance level was also conducted between CCDE and every peer algorithm.

From Table 8 we can see, CCDE outperformed other competitors on this FUDS dataset in general. According to the reported fitness values, CCDE was the best algorithm on four statistical indicators, such as median, maximum, average and standard deviation value over all the runs. SEE obtained the minimum fitness value, but its average performance was not stable. It may be caused by a good initial population. From the non-parametric test results, CCDE was significantly better than all the other peer algorithms. Compared with results on DST dataset, reported results of all these compared algorithms deteriorated to some extent, since FUDS dataset is more complicated than DST dataset. Moreover, the optimal solutions optimized by all the algorithms are shown in Table 9.

From Table 9 we can see, there was no significant differences among most of the optimal v_{OCV} solutions. However, the three parameters (R_0, R_P and C_P) solved by different algorithms varied quite dramatically, compared with the reported results shown in Table 7. It is meant that, using FUDS dataset, there may be multiple feasible parameters which all could perform dynamic behaviors of battery well. This phenomenon was rarely reported by all other previous published investigations. Generally speaking, when researchers estimate SOC by using Kalman filter methods or other similar methods, the optimal parameters of battery are already known and unique by default. Multiple acceptable parameters of battery may make those conclusions obtained by such methods not reliable. The authors believed that it is an interesting topic for further investigations in the future. To show the convergence performances, the convergence curves of the involved methods with median fitness values were plotted as shown in Fig. 9..

In Fig. 9(a), all the methods showed similar convergence but with much more fluctuation, compared with Fig. 7(a). Again, this is because FUDS data is more complicated than DST data. In FUDS test, discharging and charging current in dynamic procedure change more dramatically with no regularity. The difference between DST and FUDS could be seen clearly by comparing the profiles of i_L and v_{term} in Fig. 7 (a) and (b). The authors would like to explain the fluctuation of the convergence curves again. That is, for parameters identification of battery in this paper, the global best fitness value may be bigger than the value in the last generation due to the updating mechanism of $\mathbf{x}^*$ as shown in line 25 in Algorithm 3. After each cycle/sub-generation, some variables in $\mathbf{x}^*$ will be replaced by the best solution in each $\mathbf{X}_{sub}$, resulting in possibly deteriorated fitness value of $\mathbf{x}^*$ with full length. Roughly speaking, CCDE still performed best on convergence with median fitness value over 31 runs. To make this observation more clear, we re-plotted Fig. 9(a) by eliminating the global best fitness values which are bigger than the current global best fitness value, as shown in Fig. 9(b). In Fig. 9(b), all the methods showed continuously exploring capability in the former evolving stage, but they all failed to improve their global fitness values in the later stage. This means that they may be trapped by local optima. In Fig. 9(c), we plotted the convergence curves which denote the average fitness values over all the independent runs. From this figure we can see, the averaged convergence curves look similar to the convergence curves with the median fitness values, but the averaged convergence curves fluctuated more violently. Also, this comparison on averaged convergence curves demonstrated that the proposed methodology could find better solutions on average. Despite all this, among the comparators, CCDE still showed the best performance on convergence. Further, comparisons of the involved algorithms on time cost are shown in Fig. 10.

From Fig. 10 we can see, time cost of the compared algorithms behaved with slightly more fluctuation compared with Fig. 8. Besides, the average time cost on FUDS data is more than the time cost of DST data. This is another evidence to show that FUDS dataset is more complicated than DST dataset. Generally speaking, it seems no significant differences on time complexity among the involved algorithms.



(a) Overall convergence curves with median fitness values (b) Convergence curves remaining better fitness only

(c) Convergence curves with averaged fitness values over all the independent runs

Fig. 9. Convergence curves of compared algorithms on FUDS dataset.

4.4. Comparisons with classical DE algorithms

In order to demonstrate effectiveness of the proposed CCDE further, we chose the following seven classic highly-cited DE algorithms for comparison. In this section, the compared algorithms are SaDE (Qin et al., 2009), JADE (Zhang & Sanderson, 2009), jDE (Brest et al., 2006), CoDE (Wang et al., 2011), EPSDE (Mallipeddi et al., 2011), MPEDE (Wu et al., 2016) and DEPSO (Wang, Li, & Yang, 2019). The parametric settings of these compared algorithms are shown in Table 10. In this part, the same CC framework of parameters identification is used for all the algorithms. In each cycle, every sub-population is optimized by these involved algorithms in Table 10.

All the settings on simulation and parameters are the same as the settings in Section 4.4. For DST data, $k_{max} = 6650$, $D_{sub} = 13$, $MAX\_FES_{sub} = 4000$, $NP = 20$, $MAX\_FES = 5.32E + 06$. For FUDS data, $k_{max} = 6946$, $D_{sub} = 14$, $MAX\_FES_{sub} = 4000$, $NP = 20$, $MAX\_FES = 5.55E + 06$. The number of independent runs is set to 31. We also conducted Wilcoxon rank sum test at a 0.05 significance level. Sign "+" indicates CCDE is significantly better than the peer competitor. Sign "-" means CCDE is significantly worse than the peer competitor. Sign "≈" denotes that the two algorithms perform similarly. Experimental results on DST and FUDS data are shown in Table 11.

From the statistical results reported in Table 11, the proposed CCDE still showed promising capability for solving this problem, by being compared with some classic DE variants. First, under the proposed CC framework in this work, all the algorithm could identifying parameters and obtain competitive results. For example, MPEDE obtained the minimum value of Std on DST data, and DEPSO obtained the minimum value of Std on FUDS data. Whereas, CCDE still performed best on the averaged fitness value over all the runs. From the Wilcoxon test results, CCDE performed significantly better than jDE, EPSDE, MPEDE and DEPSO on DST data, and CCDE tied with SaDE, JADE and CoDE. On FUDS data, CCDE only performed similarly to SaDE. The other algorithms were beaten by the proposed method. Considering these

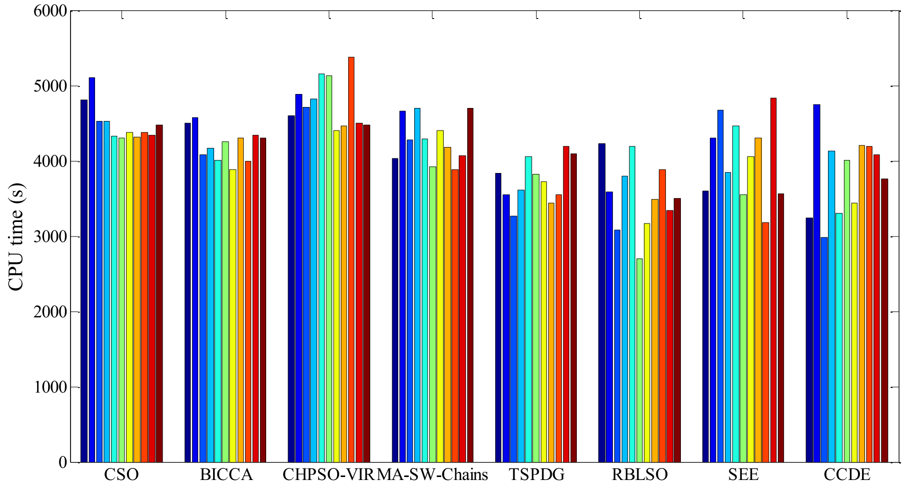


Fig. 10. CPU time of the algorithms over partial independent runs (11 out of 31) on FUDS dataset.

Table 10
Parameter configurations of compared algorithms.

Method	Parameters settings
SaDE	$F = \text{Norm}(0.5, 0.3)$, $Cr = \text{Norm}(Cr_{mk}, 0.1)$, $LP = 50$
JADE	$c = 0.1$, $p = 0.05$, $\mu_F = 0.5$, $\mu_{CR} = 0.5$
jDE	$F_l = 0.1$, $F_u = 0.9$, $\tau_1 = \tau_2 = 0.1$
CoDE	Three trial vector generation strategies and three control parameter settings
EPSDE	Cr range [0.1, 0.9] and F range [0.4, 0.9]
MPEDE	$\lambda_1 = 0.2$, $ng = 20$
DEPSO	$SEP = 30$, $\gamma = 0.001$, $F_l = 0.1$, $F_u = 0.8$, $Cr_l = 0.3$, $Cr_u = 1.0$
CCDE	$F_l = 0.1$, $F_u = 0.9$, $Cr_u = 0.5$, $\tau_1 = \tau_2 = 0.1$

compared algorithm are well designed for solving various optimization problems, we may conclude that the performance of the proposed CCDE is acceptable and satisfying.

In general, we make a short discussion on the above experimental results. First, the proposed methodology which consists of m-decomposition and the corresponding DE algorithm is able to solve the problem which is parameters identification of battery. Simulations on two dynamic test datasets demonstrated that the presented CCDE could find outstanding solutions with higher accuracy, compared with 7 other state-of-the-art CC methods. Second, it is feasible to transform parameters identification of battery into a LSOP, which could be solved by using CC methods. This methodology could be consistent with measured i_L and v_{term} based on the ECM without introducing any pre-assumptions. Compared with LS methods, the presented methodology could be applied for parameters identification of battery with higher orders and more complex models with strong non-linearization. Third, the parameters of battery need further investigations. As shown in the experimental results, with ECM of battery, there may be multiple feasible parameters which all could perform dynamic behaviors of battery well. It is necessary to research how to choose a final parameter setting from the candidate pool.

5. Conclusions

This paper proposed a new cooperative co-evolution algorithm named as CCDE for identifying parameters of LIB. Under this framework, this problem was innovatively transformed into a LSOP to realize global optimization for the parameters of battery. The first-order RC model was solved numerically by using improved Euler's method. Based on the framework in this work, R_0, R_P, C_P and partial $v_{OCV}(k)$ could be found simultaneously in each sub-population by using CCDE algorithm. Compared with several newly published state-of-the-art CC methods, comprehensive experimental results demonstrated that, the proposed methodology is feasible and effective for identifying parameters of battery. That means, given a set of measured i_L and v_{term}, the most fitted parameters of LIB could be optimized directly with any non-linearity. It also demonstrated that the proposed methodology could identify parameters of complex ECMs with strong non-linearity. Besides, the authors have found an interesting phenomenon which revealed that there may be multiple optimal parameters for battery. In the future, further investigations should be conducted on how to select a final parametric setting and how to define the limits of the variables. Meanwhile, it is also important to apply this methodology for estimating SOC or identifying parameters in electrochemical model in future works.

CRediT authorship contribution statement

Chuan Wang: Conceptualization, Methodology, Software, Writing – original draft, Formal analysis, Investigation. **Minyi Xu:** Funding acquisition, Validation. **Qinjin Zhang:** Resources. **Ruizheng Jiang:** Funding acquisition, Writing – review & editing. **Jinhong Feng:**

Table 11
Mean and variance of fitness values of CCDE and 7 classic DE variants.

Algorithms	DST data				FUDS data			
	Mean	Std	W-test	p-value	Mean	Std	W-test	p-value
SaDE	5.38E + 02	4.26E + 02	≈	4.22E-01	1.36E + 03	1.22E + 03	≈	1.18E-01
JADE	4.37E + 02	3.74E + 02	≈	2.31E-01	1.29E + 03	1.14E + 03	+	1.87E-02
jDE	5.80E + 02	4.28E + 02	+	2.91E-02	1.17E + 03	9.99E + 02	+	3.02E-02
CoDE	4.71E + 02	3.94E + 02	≈	3.31E-01	1.33E + 03	1.04E + 03	+	2.97E-03
EPSDE	5.49E + 02	4.22E + 02	+	2.84E-03	1.31E + 03	1.19E + 03	+	1.32E-02
MPEDE	5.74E + 02	**3.10E + 02**	+	5.34E-04	1.37E + 03	1.11E + 03	+	7.66E-04
DEPSO	4.98E + 02	3.31E + 02	+	1.55E-02	1.69E + 03	**9.59E + 02**	+	5.33E-05
CCDE	**3.59E + 02**	3.39E + 02	–	–	**9.67E + 02**	1.00E + 03	–	–

Funding acquisition, Writing – review & editing. **Yi Wei:** Visualization. **Yancheng Liu:** Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work would like to thank the anonymous reviewers and their comments.

Funding

This work was supported by National Natural Science Foundation of China [grant numbers 51709027, 51506019].

References

Allahviranloo, T., & Salahshour, S. (2011). Euler method for solving hybrid fuzzy differential equation. *Soft Computing, 15*(7), 1247–1253. https://doi.org/10.1007/s00500-010-0659-y

Andre, D., Meiler, M., Steiner, K., Walz, H., Soczka-Guth, T., & Sauer, D. U. (2011). Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. II: Modelling. *Journal of Power Sources, 196*(12SI), 5349–5356. https://doi.org/10.1016/j.jpowsour.2010.07.071

Barcellona, S., & Piegari, L. (2017). Lithium ion battery models and parameter identification techniques. *Energies, 10*(200712). https://doi.org/10.3390/en10122007

Belt, J., Utgikar, V., & Bloom, I. (2011). Calendar and PHEV cycle life aging of high-energy, lithium-ion cells containing blended spinel and layered-oxide cathodes. *Journal of Power Sources, 196*(23), 10213–10221. https://doi.org/10.1016/j.jpowsour.2011.08.067

Bergh, F., & Engelbrecht, A. P. (2000). Cooperative learning in neural networks using particle swarm optimizers. *South African Computer Journal, 26*, 84–90.

Brest, J., Greiner, S., Bokovi, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation, 10*(6), 646–657. https://doi.org/10.1109/TEVC.2006.872133

Chen, M., & Rincon-Mora, G. A. (2006). Accurate electrical battery model capable of predicting, runtime and I-V performance. *IEEE Transactions on Energy Conversion, 21*(2), 504–511. https://doi.org/10.1109/TEC.2006.874229

Chen, Z., Yang, L., Zhao, X., Wang, Y., & He, Z. (2019). Online state of charge estimation of Li-ion battery based on an improved unscented Kalman filter approach. *Applied Mathematical Modelling, 70*, 532–544. https://doi.org/10.1016/j.apm.2019.01.031

Cheng, R., & Jin, Y. (2015). A competitive swarm optimizer for large scale optimization. *IEEE Transactions on Cybernetics, 45*(2), 191–204. https://doi.org/10.1109/TCYB.2014.2322602

Dai, H., Wei, X., Sun, Z., Wang, J., & Gu, W. (2012). Online cell SOC estimation of Li-ion battery packs using a dual time-scale Kalman filtering for EV applications. *Applied Energy, 95*, 227–237. https://doi.org/10.1016/j.apenergy.2012.02.044

Das, S., Mullick, S. S., & Suganthan, P. N. (2016). Recent advances in differential evolution-An updated survey. *Swarm and Evolutionary Computation, 27*, 1–30. https://doi.org/10.1016/j.swevo.2016.01.004

Das, S., & Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation, 15*(1), 4–31. https://doi.org/10.1109/TEVC.2010.2059031

Deng, H., Peng, L., Zhang, H., Yang, B., & Chen, Z. (2019). Ranking-based biased learning swarm optimizer for large-scale optimization. *Information Sciences, 493*, 120–137. https://doi.org/10.1016/j.ins.2019.04.037

Dubarry, M., & Liaw, B. Y. (2007). Development of a universal modeling tool for rechargeable lithium batteries. *Journal of Power Sources, 174*(2), 856–860. https://doi.org/10.1016/j.jpowsour.2007.06.157

Duong, T. Q. (2000). USABC and PNGV test procedures. *Journal of Power Sources, 89*(2), 244–248. https://doi.org/10.1016/S0378-7753(00)00439-0

Fan, Q., & Yan, X. (2015). Self-adaptive differential evolution algorithm with discrete mutation control parameters. *Expert Systems with Applications, 42*(3), 1551–1572. https://doi.org/10.1016/j.eswa.2014.09.046

Ferahtia, S., Djeroui, A., Rezk, H., Chouder, A., Houari, A., et al. (2021). Optimal parameter identification strategy applied to lithium-ion battery model. *International Journal of Energy Research, 45*(11), 16741–16753. https://doi.org/10.1002/er.6921

Ge, H., Sun, L., Tan, G., Chen, Z., & Chen, C. L. P. (2017). Cooperative hierarchical PSO with two stage variable interaction reconstruction for large scale optimization. *IEEE Transactions on Cybernetics, 47*(9SI), 2809–2823. https://doi.org/10.1109/TCYB.2017.2685944

Ge, H., Sun, L., Yang, X., Yoshida, S., & Liang, Y. (2015). Cooperative differential evolution with fast variable interdependence learning and cross-cluster mutation. *Applied Soft Computing, 36*, 300–314. https://doi.org/10.1016/j.asoc.2015.07.016

Ge, H., Zhao, M., Hou, Y., Kai, Z., Sun, L., Tan, G., … Chen, C. L. P. (2020). Bi-space interactive cooperative coevolutionary algorithm for large scale black-box optimization. *Applied Soft Computing, 97*(106798A). https://doi.org/10.1016/j.asoc.2020.106798

He, H., Xiong, R., & Peng, J. (2016). Real-time estimation of battery state-of-charge with unscented Kalman filter and RTOS COS-II platform. *Applied Energy, 162*, 1410–1418. https://doi.org/10.1016/j.apenergy.2015.01.120

Hu, X., He, F., Chen, W., & Zhang, J. (2017). Cooperation coevolution with fast interdependency identification for large scale optimization. *Information Sciences, 381*, 142–160. https://doi.org/10.1016/j.ins.2016.11.013

Hu, Y., Yurkovich, S., Guezennec, Y., & Yurkovich, B. J. (2009). A technique for dynamic battery model identification in automotive applications using linear parameter varying structures. *Control Engineering Practice, 17*(10), 1190–1201. https://doi.org/10.1016/j.conengprac.2009.05.002

Jeon, D. H., & Baek, S. M. (2011). Thermal modeling of cylindrical lithium ion battery during discharge cycle. *Energy Conversion and Management, 52*(8–9), 2973–2981. https://doi.org/10.1016/j.enconman.2011.04.013

Kamrani, M., Hosseini, S. M., & Hausenblas, E. (2018). Implicit Euler method for numerical solution of nonlinear stochastic partial differential equations with multiplicative trace class noise. *Mathematical Methods in the Applied Sciences, 41*(13), 4986–5002. https://doi.org/10.1002/mma.4946

Kennedy, J., & Mendes, R. (2002, 2002-01-01). *Population structure and particle swarm performance*. Paper presented at the 2002 Congress on Evolutionary Computation, Honolulu, HI, United states.

Kwak, M., Lkhagvasuren, B., Park, J., & You, J. (2020). Parameter identification and SOC estimation of a battery under the hysteresis effect. *IEEE Transactions on Industrial Electronics, 67*(11), 9758–9767. https://doi.org/10.1109/TIE.2019.2956394

Lee, K., Dai, M., & Chuang, C. (2018). Temperature-compensated model for lithium-ion polymer batteries with extended Kalman filter state-of-charge estimation for an implantable charger. *IEEE Transactions on Industrial Electronics, 65*(1), 589–596. https://doi.org/10.1109/TIE.2017.2721880

Li, J., Lai, Q., Wang, L., Lyu, C., & Wang, H. (2016). A method for SOC estimation based on simplified mechanistic model for LiFePO4 battery. *Energy, 114*, 1266–1276. https://doi.org/10.1016/j.energy.2016.08.080

Liu, Y., Yao, X., Zhao, Q., & Higuchi, T. (2001, 2001-01-01). *Scaling up fast evolutionary programming with cooperative coevolution*. Paper presented at the Congress on Evolutionary Computation IEEE, 2001.

Lu, L., Han, X., Li, J., Hua, J., & Ouyang, M. (2013). A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of Power Sources, 226*, 272–288. https://doi.org/10.1016/j.jpowsour.2012.10.060

Mallipeddi, R., Suganthan, P. N., Pan, Q. K., & Tasgetiren, M. F. (2011). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing, 11*, 1679–1696. https://doi.org/10.1016/j.asoc.2010.04.024

Mastali, M., Vazquez-Arenas, J., Fraser, R., Fowler, M., Afshar, S., et al. (2013). Battery state of the charge estimation using Kalman filtering. *Journal of Power Sources, 239*, 294–307. https://doi.org/10.1016/j.jpowsour.2013.03.131

Mei, Y., Omidvar, M. N., Li, X., & Yao, X. (2016). A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization. *ACM Transactions on Mathematical Software, 42*(132). https://doi.org/10.1145/2791291

Miniguano, H., Barrado, A., Lazaro, A., Zumel, P., & Fernandez, C. (2020). General parameter identification procedure and comparative study of Li-Ion battery models. *IEEE Transactions on Vehicular Technology, 69*(1), 235–245. https://doi.org/10.1109/TVT.2019.2952970

Molina, D., Lozano, M., Sanchez, A. M., & Herrera, F. (2011). Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains. *Soft Computing, 15*(11SI), 2201–2220. https://doi.org/10.1007/s00500-010-0647-2

Mu, H., Xiong, R., Zheng, H., Chang, Y., & Chen, Z. (2017). A novel fractional order model based state-of-charge estimation method for lithium-ion battery. *Applied Energy, 207*, 384–393. https://doi.org/10.1016/j.apenergy.2017.07.003

Omidvar, M. N., Li, X., Mei, Y., & Yao, X. (2014). Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on Evolutionary Computation, 18*(3), 378–393. https://doi.org/10.1109/TEVC.2013.2281543

Omidvar, M. N., Li, X., Yang, Z., & Yao, X. (2010). Cooperative Co-evolution for Large Scale Optimization Through More frequent Random Grouping *IEEE Congress on Evolutionary Computation*. (Reprinted.

Omidvar, M. N., Li, X., & Yao, X. (2010). Cooperative Co-evolution with Delta Grouping for Large Scale Non-separable Function Optimization *IEEE Congress on Evolutionary Computation*. (Reprinted.

Omidvar, M. N., Yang, M., Mei, Y., Li, X., & Yao, X. (2017). DG2: A faster and more accurate differential grouping for large-scale black-box optimization. *IEEE Transactions on Evolutionary Computation, 21*(6), 929–942. https://doi.org/10.1109/TEVC.2017.2694221

Peng, J., Luo, J., He, H., & Lu, B. (2019). An improved state of charge estimation method based on cubature Kalman filter for lithium-ion batteries. *Applied Energy, 253*(113520). https://doi.org/10.1016/j.apenergy.2019.113520

Potter, M. A., & De Jong, K. A. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation, 8*(1), 1–29. https://doi.org/10.1162/106365600568086

Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation, 13*(2), 398–417. https://doi.org/10.1109/TEVC.2008.927706

Rakhmatov, D., Vrudhula, S., & Wallach, D. A. (2003). A model for battery lifetime analysis for organizing applications on a pocket computer. *IEEE Transactions on Very*

Large Scale Integration (VLSI) Systems, 11(6), 1019–1030. https://doi.org/10.1109/TVLSI.2003.819320

Ramadesigan, V., Chen, K., Burns, N. A., Boovaragavan, V., Braatz, R. D., … Subramanian, V. R. (2011). Parameter estimation and capacity fade analysis of lithium-ion batteries using reformulated models. *Journal of the Electrochemical Society, 158*(9), A1048–A1054. https://doi.org/10.1149/1.3609926

Ren, Z., Liang, Y., Zhang, A., Yang, Y., Feng, Z., … Wang, L. (2019). Boosting cooperative coevolution for large scale optimization with a fine-grained computation resource allocation strategy. *IEEE Transactions on Cybernetics, 49*(12), 4180–4193. https://doi.org/10.1109/TCYB.2018.2859635

Sepasi, S., Ghorbani, R., & Liaw, B. Y. (2014). A novel on-board state-of-charge estimation method for aged Li-ion batteries based on model adaptive extended Kalman filter. *Journal of Power Sources, 245*, 337–344. https://doi.org/10.1016/j.jpowsour.2013.06.108

Shi, Y. J., Teng, H. F., & Li, Z. Q. (2005). Cooperative co-evolutionary differential evolution for function optimization. In L. Wang, K. Chen & Y. S. Ong (Eds.), *Lecture Notes in Computer Science* (3611, pp. 1080-1088). (Reprinted.

Smith, K. A., Rahn, C. D., & Wang, C. (2007). Control oriented ID electrochemical model of lithium ion battery. *Energy Conversion and Management, 48*(9), 2565–2578. https://doi.org/10.1016/j.enconman.2007.03.015

Sun, L., Yoshida, S., Cheng, X., & Liang, Y. (2012). A cooperative particle swarm optimizer with statistical variable interdependence learning. *Information Sciences, 186*(1), 20–39. https://doi.org/10.1016/j.ins.2011.09.033

Tanabe, R., & Fukunaga, A. S. (2014). *Improving the Search Performance of SHADE Using Linear Population Size Reduction*. Paper presented at the IEEE Congress on Evolutionary Computation (CEC), Beijing, PEOPLES R CHINAfrom http://gateway.isiknowledge.com/gateway/Gateway.cgi?GWVersion=2&SrcAuth=AegeanSoftware&SrcApp=NoteExpress&DestLinkType=FullRecord&DestApp=WOS&KeyUT=000356684602045.

van den Bergh, F., & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*(3), 225–239. https://doi.org/10.1109/tevc.2004.826069

Wang, S., Li, Y., Yang, H., & Liu, H. (2018). Self-adaptive differential evolution algorithm with improved mutation strategy. *Soft Computing, 22*(10), 3433–3447. https://doi.org/10.1007/s00500-017-2588-5

Wang, S., Li, Y., & Yang, H. (2019). Self-adaptive mutation differential evolution algorithm based on particle swarm optimization. *Applied Soft Computing Journal, 81*. https://doi.org/10.1016/j.asoc.2019.105496

Wang, Y., Cai, Z., & Zhang, Q. (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation, 15*(1), 55–66. https://doi.org/10.1109/TEVC.2010.2087271

Wang, Y., Pan, R., Liu, C., Chen, Z., & Ling, Q. (2018). Power capability evaluation for lithium iron phosphate batteries based on multi-parameter constraints estimation. *Journal of Power Sources, 374*, 12–23. https://doi.org/10.1016/j.jpowsour.2017.11.019

Wei, J., Dong, G., Chen, Z., & Kang, Y. (2017). System state estimation and optimal energy control framework for multicell lithium-ion battery system. *Applied Energy, 187*, 37–49. https://doi.org/10.1016/j.apenergy.2016.11.057

Wu, G., Mallipeddi, R., Suganthan, P. N., Wang, R., & Chen, H. (2016). Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences, 329*, 329–345. https://doi.org/10.1016/j.ins.2015.09.009

Xing, Y., Ma, E. W. M., Tsui, K. L., & Pecht, M. (2011). Battery management systems in electric and hybrid vehicles. *Energies, 4*(11), 1840–1857. https://doi.org/10.3390/en4111840

Xu, K., Li, H., & Liu, Z. (2018). ISOMAP-based spatiotemporal modeling for lithium-ion battery thermal process. *IEEE Transactions on Industrial Informatics, 14*(2), 569–577. https://doi.org/10.1109/TII.2017.2743260

Xuan, D., Shi, Z., Chen, J., Zhang, C., & Wang, Y. (2020). Real-time estimation of state-of-charge in lithium-ion batteries using improved central difference transform method. *Journal of Cleaner Production, 252*(119787). https://doi.org/10.1016/j.jclepro.2019.119787

Xue, X., Zhang, K., Li, R., Zhang, L., Yao, C., Wang, J., … Yao, J. (2020). A topology-based single-pool decomposition framework for large-scale global optimization. *Applied Soft Computing, 92*(106295). https://doi.org/10.1016/j.asoc.2020.106295

Yang, P., Tang, K., & Yao, X. (2018). Turning high-dimensional optimization into computationally expensive optimization. *IEEE Transactions on Evolutionary Computation, 22*(1SI), 143–156. https://doi.org/10.1109/TEVC.2017.2672689

Yang, Z., Tang, K., & Yao, X. (2008). Large scale evolutionary optimization using cooperative coevolution. *Information Sciences, 178*(15), 2985–2999. https://doi.org/10.1016/j.ins.2008.02.017

Yang, Z., Tang, K., & Yao, X. (2008b). Multilevel Cooperative Coevolution for Large Scale Optimization *IEEE Congress on Evolutionary Computation* (1663). (Reprinted.

Yu, Z., Xiao, L., Li, H., Zhu, X., & Huai, R. (2017). Model parameter identification for lithium batteries using the coevolutionary particle swarm optimization method. *IEEE Transactions on Industrial Electronics, 64*(7), 5690–5700. https://doi.org/10.1109/TIE.2017.2677319

Zeng, M., Zhang, P., Yang, Y., Xie, C., & Shi, Y. (2019). SOC and SOH joint estimation of the power batteries based on fuzzy unscented Kalman filtering algorithm. *Energies, 12*(312216). https://doi.org/10.3390/en12163122

Zeng, Z., Tian, J., Li, D., & Tian, Y. (2018). An online state of charge estimation algorithm for lithium-ion batteries using an improved adaptive cubature Kalman Filter. *Energies, 11*(591). https://doi.org/10.3390/en11010059

Zhang, J., & Sanderson, A. C. (2009). JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation, 13*(5), 945–958. https://doi.org/10.1109/TEVC.2009.2014613

Zhang, X., Wang, Y., Liu, C., & Chen, Z. (2018). A novel approach of battery pack state of health estimation using artificial intelligence optimization algorithm. *Journal of Power Sources, 376*, 191–199. https://doi.org/10.1016/j.jpowsour.2017.11.068

Zheng, F., Xing, Y., Jiang, J., Sun, B., Kim, J., … Pecht, M. (2016). Influence of different open circuit voltage tests on state of charge online estimation for lithium-ion batteries. *Applied Energy, 183*, 513–525. https://doi.org/10.1016/j.apenergy.2016.09.010

Zhu, Q., Xu, M., Liu, W., & Zheng, M. (2019). A state of charge estimation method for lithium-ion batteries based on fractional order adaptive extended kalman filter. *Energy, 187*(115880). https://doi.org/10.1016/j.energy.2019.115880