

Evolutionary Sensor Placement for Spatiotemporal Modeling of Battery Thermal Process

Yong Wang, *Senior Member, IEEE*, Shi-Hui He, and Bing-Chuan Wang

Abstract—Spatiotemporal modeling is critical to the simulation, optimization, and control of the thermal process of a lithium-ion battery which is a typical kind of distributed parameter system (DPS). Data-driven spatiotemporal modeling methods are of practical interest to construct an analytical model of the thermal process of a lithium-ion battery, since they only need some sampled data rather than the structure descriptions or parameters of a DPS. How to sample data optimally for data-driven spatiotemporal modeling is still an open question. In this paper, with the aim of minimizing the spatiotemporal modeling error, we propose a novel evolutionary algorithm to optimally place sensors for data sampling. First, an objective function that can quantify both the spatial error and the temporal error is designed. Additionally, a novel differential evolution algorithm with two kinds of encoding mechanisms (called DETEM) is proposed to optimize the objective function. Numerical simulations and experimental studies have shown that the proposed method is competitive. Besides, both the objective function and DETEM are critical to the proposed method. In summary, the proposed method provides an effective way to obtain the optimal sensor placement for spatiotemporal modeling of the thermal process of a lithium-ion battery.

Index Terms—Distributed parameter system, spatiotemporal modeling, evolutionary algorithm, sensor placement, lithium-ion battery.

I. INTRODUCTION

THE thermal process of a lithium-ion battery is a typical kind of distributed parameter system (DPS) that can be described by partial differential equations (PDEs) [1]. Modeling of the thermal process plays a very important role in battery management. It is difficult to construct a first-principle-based model for the thermal process, because it is an infinite-dimensional system and exhibits strongly nonlinear spatiotemporal dynamics [2]. Alternatively, the design of analytical models by spatiotemporal modeling methods is of practical interest, because spatiotemporal modeling methods are able to reduce the infinite-dimensional systems into finite-dimensional ones and to deal with spatiotemporal dynamics [3].

During the past two decades, various spatiotemporal modeling methods have been proposed to model DPSs. Among them, spectral methods have been proven to be accurate

This work was supported by the National Natural Science Foundation of China under Grants 61976225 and 61673397. (*Corresponding author*: Bing-Chuan Wang.)

Y. Wang and B.-C. Wang are with the School of Automation, Central South University, Changsha 410083, China, and also with the Hunan Xiangjiang Artificial Intelligence Academy, Changsha 410083, China (e-mail: ywang@csu.edu.cn; bewang@csu.edu.cn).

S.-H. He is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China, and also with the Hunan Xiangjiang Artificial Intelligence Academy, Changsha 410083, China (e-mail: 1550119126@qq.com).

and efficient [4]. Conventionally, a spectral method can only work for a DPS under the assumption that the first-principle-based PDE is known and its boundary conditions are homogeneous [5]. However, in many processes, it would be difficult to satisfy these conditions due to the fact that process knowledge is missed [6]. Under such circumstances, data-driven spatiotemporal modeling methods would be more helpful, because they do not rely on these prerequisites [7], [8]. A data-driven spatiotemporal modeling method constructs an analytical model by using data sampled from a DPS. To be specific, some dominant spatial basis functions (DSBFs) are first extracted from the sampled data. Subsequently, the infinite-dimensional DPS is reduced into a finite-dimensional temporal system which is expanded on the DSBFs and can be modeled by a variety of techniques [9]. Finally, an analytical model of the DPS is obtained by combining the DSBFs with the temporal model. Thus, it is potential to build an analytical model of the thermal process of a lithium-ion battery by data-driven spatiotemporal modeling methods.

Since the analytical model is built based on sampled data, the modeling accuracy highly depends on the process information conveyed by the data. The more information the sampled data conveys, the higher the modeling accuracy is. Generally speaking, the amount of information is related to the amount of data, i.e., more data contains more process information. In theory, an adequate number of sensors can be placed to sample data; thus, arbitrary modeling accuracy can be achieved [7]. Unfortunately, in practice, it is challenging to maintain a large number of sensors due to physical or economical constraints [10]. A sound way is to place a limited number of sensors optimally to extract the critical process information. In this manner, an analytical model with satisfying accuracy can be obtained [11]. In summary, it is of great significance to design an efficient sensor placement method for spatiotemporal modeling of the thermal process of a lithium-ion battery.

The sensor placement (selection) problem arises in various applications, including scheduling of wireless sensor networks [12], state estimation of dynamical systems [13], and target tracking [14]. In this problem, k sensor placements will be selected from m potential candidate locations properly, with the aim of minimizing the estimation or modeling error [15]. In fact, it is a typical combinatorial optimization problem and its complexity is NP-hard [16]. Because it is non-trivial to design a polynomial-time algorithm, various attempts have been made by researchers to find an approximated solution. As a result, some state-of-the-art methods have been proposed. In general, these methods can be divided into four categories:

- convex optimization methods [15], [16],
- greedy methods [13], [17],
- machine learning methods [11], [18],
- heuristic methods [19], [20].

Convex optimization methods relax the original sensor placement problem into a convex optimization problem by transforming nonconvex constraints into convex ones. Along with this line, Joshi and Boyd [15] derived a bound of the best performance that can be achieved. Besides, they designed a local optimization method to obtain a feasible placement. Mo *et al.* [16] considered the correlations among measurements of different sensors and then proposed a multi-step sensor selection method, with the aim of minimizing an objective function that is relaxed into a convex form. Although convex optimization methods can find an approximated solution quickly, they need to transform the objective function and constraints into convex ones, which is not a trivial work.

Greedy methods that select sensors greedily at each step are computationally efficient and simple to implement. Moreover, if the objective function is submodular, a greedy method could obtain better performance than some computationally expensive methods [17]. Jawaaid and Smith [13] figured out that the objective functions of most sensor selection problems are not submodular indeed. To address this issue, they tried to construct an objective function with submodularity by setting some restrictive conditions. Due to their simplicity and efficacy, greedy methods have been applied to a spectrum of areas [21], [22]. Although some techniques can be used to construct a submodular objective function, most greedy methods are only suitable for linear systems.

Some machine learning methods have also been proposed for sensor placement recently [11], [18]. Semaan [18] utilized random forests for optimal sensor placement. In this study, each sensor signal is regarded as an input variable; thus, the most important variables are the optimal sensors. Wang *et al.* [11] formulated the sensor selection for modeling of DPSs as a Markov decision process which is solved by a reinforcement learning method. Simulations and experiments validated that the reinforcement learning method is competitive.

Heuristic methods have emerged and become popular for sensor placement recently, which leverage heuristic information to search for promising solutions [19], [23]. Naeem *et al.* [19] proposed a binary particle swarm optimization (BPSO) algorithm to solve the sensor selection problem. Chisari *et al.* [23] applied a genetic algorithm (GA) to seek an optimal sensor layout for structural parameter estimation. In particular, heuristic methods have been widely used for sensor placement optimization in the area of mechanics. Gomes *et al.* [24] used GA to place sensors optimally for structural health monitoring (SHM) systems. A multiobjective sensor placement optimization method for SHM systems is presented which uses the finite element method to model a laminated composite plate [25]. In addition, the firefly algorithm is applied for sensor placement optimization and damage identification in a fuselage structure [26]. Different from these three methods, we focus on spatiotemporal modeling of the thermal process of a lithium-ion battery.

From the literature review, some interesting findings are

summarized as follows:

- Although numerous studies have been carried out to place sensors optimally, few efforts have been devoted to the sensor placement for modeling of complex DPSs. Wang *et al.* [11] made an interesting attempt to resolve this problem by using reinforcement learning. In this method, only the spatial objective function is considered, whereas the error of the temporal model is neglected.
- EAs do not require the objective function to be convex or submodular. Additionally, they do not need the gradient information of the objective function. Moreover, they can find a satisfying solution within a given time. Thus, EAs have been used to search for the optimal sensor placement in many complex processes. Among various kinds of EAs, GA and PSO are frequently adopted.
- Due to its numerous merits including ease of implementation, simple structure, and powerful search capability, DE performs better than GA and PSO in some layout optimization problems [27], [28]. However, as a typical kind of EA, it has been used scarcely to search for the optimal sensor placement.

Based on the above observations, a novel EA is proposed to seek the optimal sensor placement for spatiotemporal modeling of a typical kind of DPS (i.e., the thermal process of a lithium-ion battery) under the assumption that the objective function is neither convex nor submodular. First, an objective function is constructed by combining a spatial objective function with a temporal objective function. In order to optimize this objective function, a novel DE algorithm with two kinds of encoding mechanisms (called DETEM) is developed. It is worth noting that one encoding mechanism is employed for global exploration, while the other is for local exploitation.

In summary, the main contributions of this paper are highlighted as follows:

- We make the first attempt to optimize the sensor placement for spatiotemporal modeling of the thermal process of a lithium-ion battery by using DE.
- An objective function that can quantify both the spatial error and the temporal error is derived.
- A novel DE algorithm with two kinds of encoding mechanisms is proposed for optimal sensor placement.
- Simulations and experiments on the thermal process of a lithium-ion battery verify that our method is effective.

The rest of this paper is organized as follows. Section II presents the description of the considered problem. In Section III, the proposed method including the objective function construction and DETEM is elaborated. Simulations and experiments are described in Section IV and Section V, respectively. Finally, Section VI summarizes the concluding remarks.

II. PROBLEM DESCRIPTION

We consider a typical kind of DPS: the thermal process of a lithium-ion battery which can be described by the following PDE:

$$\rho C_p \frac{\partial y}{\partial t} = \frac{\partial}{\partial x} (\lambda_x \frac{\partial y}{\partial x}) + \frac{\partial}{\partial z} (\lambda_z \frac{\partial y}{\partial z}) + q(x, z, t), \quad (1)$$

subject to the boundary condition:

$$-\lambda_x \frac{\partial y}{\partial x} - \lambda_z \frac{\partial y}{\partial z} = h_0(y - y_{air}), \quad (2)$$

and the following initial condition:

$$y(x, z, 0) = y_0(x, z), \quad (3)$$

where $y(x, z, t)$ is the spatiotemporal state (i.e., temperature of the battery), x and z are the spatial coordinates, $t \in [0, \infty)$ represents the time, $q(x, z, t)$ denotes the heat generation term which is a nonlinear function of x , z , and t , ρ is the battery density, C_p is the cell heat capacity, λ_x and λ_z are the thermal conductivities across x and z directions, respectively, h_0 is the convection coefficient, y_{air} represents the ambient temperature, and $y_0(x, z)$ is a nonlinear function of x and z .

In practice, a data-driven spatiotemporal modeling method such as the Karhunen-Loéve (KL) decomposition method [1], [7] is used to design analytical models of the thermal process of a lithium-ion battery. To this end, some sensors need to be placed for data sampling. The sensor placement is an optimization problem intrinsically, which presents some challenges to conventional methods:

- The thermal process involves strongly nonlinear spatiotemporal dynamics; thus, it is non-trivial to construct an objective function for sensor placement.
- The sensor placement problem has NP-hard complexity. Its objective function is neither convex nor submodular. Thus, it is difficult for a conventional method to provide a satisfying solution.

Next, we try to leverage the advantages of EAs especially DE to solve this problem.

III. EVOLUTIONARY SENSOR PLACEMENT FOR SPATIOTEMPORAL MODELING

The proposed method includes objective function construction and DETEM. To address the first issue described in Section II, the objective function including a spatial objective function and a temporal objective function is constructed. DETEM does not require the objective function to be convex or submodular. Thus, it can address the second issue. The details are presented in the following.

A. Objective Function Construction

1) *Spatial objective function*: Suppose that $\mathbf{Y} = \{\mathbf{y}_i \in \mathbb{R}^m\}_{i=1}^N$ includes N true snapshots of the thermal process of a lithium-ion battery. Each snapshot can be expanded on DSBFs (i.e., Φ) through the KL decomposition method:

$$\mathbf{y}_i = \Phi \mathbf{a}_i + \boldsymbol{\delta}_i, \quad i = 1, \dots, N \quad (4)$$

where $\mathbf{a}_i \in \mathbb{R}^h$ is a snapshot in the finite-dimensional system and $\boldsymbol{\delta}_i$ is the model reduction error. Let \mathbf{P} be the project operator decided by sensors. We can derive that:

$$\mathbf{t}_i = \mathbf{P}(\Phi \mathbf{a}_i + \boldsymbol{\delta}_i) + \boldsymbol{\epsilon}_i, \quad i = 1, \dots, N \quad (5)$$

where $\mathbf{T} = \{\mathbf{t}_i \in \mathbb{R}^m\}_{i=1}^N$ denotes snapshots sampled by the sensors and $\boldsymbol{\epsilon}_i$ represents the random error and includes independent and identically distributed random variables that

obey Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$. As a result, the least-squares estimation of \mathbf{a}_i can be derived as follows:

$$\tilde{\mathbf{a}}_i = (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{R}^T (\mathbf{t}_i - \mathbf{P} \boldsymbol{\delta}_i), \quad i = 1, \dots, N \quad (6)$$

where $\mathbf{R} = \mathbf{P}\Phi$. The estimation error, i.e., $(\mathbf{a}_i - \tilde{\mathbf{a}}_i)$, has zero mean and its covariance can be obtained:

$$\Sigma = \sigma^2 \mathbf{Q} \quad (7)$$

where $\mathbf{Q} = (\mathbf{R}^T \mathbf{R})^{-1}$. Similar to [15], the volume of the η -confidence ellipsoid is used to measure the quality of estimation:

$$vol = \frac{(\alpha\pi)^{\frac{h}{2}}}{\Gamma(\frac{h}{2+1})} \det(\Sigma)^{\frac{1}{2}} \quad (8)$$

where $\alpha = F_{\chi_h^2}^{-1}(\eta)$, $F_{\chi_h^2}$ is the cumulative distribution function of a χ^2 -squared random variable with h degrees of freedom, and Γ represents the Gamma function. Since a small value of vol means a good estimation, the spatial objective function is constructed as follows:

$$\min F_s(\mathbf{z}) = \log \det((\widetilde{\mathbf{R}}^T \widetilde{\mathbf{R}})^{-1}) \quad (9)$$

Note that $\widetilde{\mathbf{R}} = \widetilde{\mathbf{P}}\Phi$ and $\widetilde{\mathbf{P}}$ is described as follows:

$$\widetilde{\mathbf{P}} = \mathbf{P} \cdot [\mathbf{z} \mathbf{1}^T] \quad (10)$$

where \cdot represents the dot product, $\mathbf{1}$ denotes a vector of ones, and $\mathbf{1}^T \mathbf{z} = k$ with $z_i \in \{0, 1\}$, $i = 1, \dots, m$. Note that \mathbf{P} is not a reduction operator as described in [11]. Besides, we do not make the assumption that the model reduction error (i.e., $\boldsymbol{\delta}_i$) obeys Gaussian distribution. Thus, the proposed spatial objective function is more general.

2) *Temporal objective function*: Through the KL decomposition method, the considered process is reduced into a finite-dimensional system which can be calculated as $\mathbf{C} = \Phi^T \mathbf{T}$. Traditionally, a temporal model can be constructed for the finite-dimensional system by a machine learning technique. In view of its ability of universal approximation, the radial basis function neural network is used to construct the temporal model. Suppose that $\widetilde{\mathbf{C}}$ includes the outputs of the temporal model. As a result, the temporal objective function is constructed as follows:

$$\min F_t(\mathbf{z}) = \sum_{i=1}^h \sum_{j=1}^{N_v} (c_{i,j} - \tilde{c}_{i,j})^2 \quad (11)$$

where $c_{i,j}$ and $\tilde{c}_{i,j}$ are the (i, j) th entries of \mathbf{C} and $\widetilde{\mathbf{C}}$, respectively, and N_v is the size of the validation set.

By considering $F_s(\mathbf{z})$ and $F_t(\mathbf{z})$ at the same time, the sensor placement is a multiobjective optimization problem (MOP). As described in [29], the computation costs of MOPs are high. Additionally, it is difficult for engineers to select one or a few optimal solutions from the approximated Pareto set. Thus, we use the weight method to formulate the sensor placement problem as follows:

$$\begin{aligned} \min F(\mathbf{z}) &= \widetilde{F}_s(\mathbf{z}) + \alpha \cdot \widetilde{F}_t(\mathbf{z}) \\ \text{subject to: } &\mathbf{1}^T \mathbf{z} = k \\ &z_i \in \{0, 1\}, i = 1, \dots, m \end{aligned} \quad (12)$$

where $\tilde{F}_s(\mathbf{z})$ and $\tilde{F}_t(\mathbf{z})$ are the normalized $F_s(\mathbf{z})$ and $F_t(\mathbf{z})$, respectively, and $\alpha \in [0, 1]$ is a weight parameter. In this paper, we consider that $\tilde{F}_s(\mathbf{z})$ and $\tilde{F}_t(\mathbf{z})$ are equally important and set α to 1. \mathbf{z} represents a solution to the sensor placement problem, where $z_i = 1$ means that the i th location is selected. In the next section, DETEM is presented to solve this problem.

B. DETEM

1) *Motivation:* DE owns various merits. First, it does not require the objective function to be convex or submodular. Besides, the population-based search enables it to avoid a local optimum. Among various kinds of EAs, DE has a simple structure and can be implemented with only a few lines of codes. Moreover, its outperformed performance has been verified in academic competitions [30] and industrial applications [27]. Thus, we apply DE to design an optimization algorithm that can solve Eq. (12) effectively.

The major differences between DE and other EAs lie in the way of offspring generation and the selection mechanism. DE is characterized by the one-to-one spawning and selection mechanism. In this manner, it can explore many search points and thus avoid a local optimum. However, DE sometimes explores too many search points before locating the global optimum. That is to say, its capability of exploitation would be sacrificed to a certain degree. As we know, the tradeoff between exploration and exploitation is critical to EAs. Thus, it is necessary to enhance the ability of conventional DE for local exploitation [31].

Motivated by [27], DE with a new encoding mechanism is designed for local exploitation. Different from the conventional encoding mechanism, the new mechanism considers each dimension as an individual. By searching around a specific solution, DE with the new encoding mechanism is similar to a local search method. By applying it to the best solution in the population, the exploitation can be enhanced.

Based on these observations, we propose a novel DE algorithm called DETEM by combining the conventional encoding mechanism with the new encoding mechanism. To be specific, the former is used for exploration, while the latter for exploitation. Thus, DETEM can achieve a tradeoff between exploration and exploitation. As a result, it can find a satisfying solution finally.

2) *Framework:* The framework of DETEM is described in Fig. 1. First, a population including P_s solutions (i.e., $\{\mathbf{z}_1, \dots, \mathbf{z}_{P_s}\}$) is sampled from $[0, 1]^m$ randomly. Subsequently, DE with the conventional encoding mechanism and DE with a new encoding mechanism will be executed iteratively for global exploration and local exploitation, respectively. To be specific, the global exploration will be executed for N_g generations. By exploring the whole search space, it can locate some promising regions. Afterward, the local exploitation will be conducted for N_l generations. By searching around the best solution in the population, a better one would be located. These two steps will be repeated until the maximal generation number (i.e., T) is achieved.

3) *Global exploration:* DE with the conventional encoding mechanism is used for the global exploration. As shown in

Algorithm 1: Global Exploration

```

1  $t_g \leftarrow 0;$ 
2 while  $t_g < N_g$  do
3   for  $i = 1 : P_s$  do
4     Generate  $\mathbf{u}_i$  for  $\mathbf{z}_i$  according to Eq. (13) and Eq. (14);
5     Evaluate  $\mathbf{u}_i$  according to Eq. (12);
6     if  $\mathbf{u}_i$  is better than  $\mathbf{z}_i$  then
7       | Replace  $\mathbf{z}_i$  with  $\mathbf{u}_i$ ;
8     end
9   end
10   $t_g \leftarrow t_g + 1;$ 
11 end
12 Output the best solution (i.e.,  $\mathbf{z}_{best}$ ) in terms of Eq. (12);

```

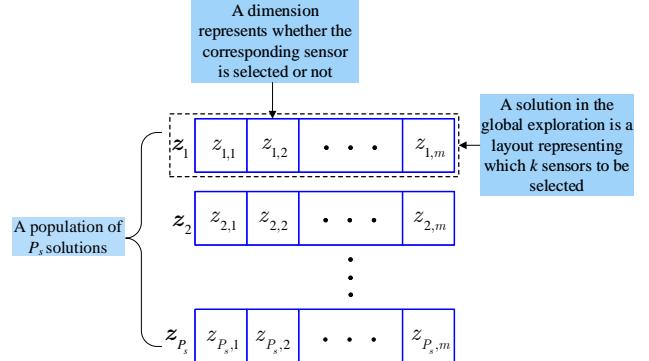


Fig. 2. The conventional encoding mechanism used for the global exploration.

Fig. 2, in this mechanism, each solution is a layout representing which k sensors to be selected. DE is applied to evolve the whole population of P_s solutions. Due to its powerful search capability, DE/rand/1/bin [30] is used to generate offsprings:

$$\mathbf{v}_i = \mathbf{z}_{r1} + K(\mathbf{z}_{r2} - \mathbf{z}_{r3}) \quad (13)$$

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } rand_j < CR \text{ or } j = j_{rand} \\ z_{i,j}, & \text{otherwise} \end{cases} \quad (14)$$

where \mathbf{z}_{r1} , \mathbf{z}_{r2} , and \mathbf{z}_{r3} are three different solutions selected from $\{\mathbf{z}_1, \dots, \mathbf{z}_{P_s}\}/\mathbf{z}_i$ randomly, \mathbf{v}_i and \mathbf{u}_i are the mutant vector and the offspring, respectively, $z_{i,j}$, $v_{i,j}$, and $u_{i,j}$ are the j th dimensions of \mathbf{z}_i , \mathbf{v}_i , and \mathbf{u}_i , respectively, $rand_j \in [0, 1]$ is a random value generated from a uniform distribution, K and CR are the scaling factor and the crossover control parameter, respectively, and j_{rand} is an integer selected from $\{1, \dots, m\}$. If \mathbf{u}_i is better than \mathbf{z}_i according to Eq. (12), it replaces \mathbf{z}_i . The details of the global exploration are described in Algorithm 1.

4) *Local exploitation:* The aim of the local exploitation is to seek a better solution by searching around the best solution (denoted as $\mathbf{z}_{best} = (z_{best,1}, \dots, z_{best,m})$) in the current population. To this end, a new encoding mechanism is adopted. As shown in Fig. 3, in this mechanism, an individual is one dimension of \mathbf{z}_{best} and represents whether the corresponding sensor is selected or not. Thus, \mathbf{z}_{best} is transformed into a population of m individuals. First, DE is used to evolve the population. Next, the individuals are utilized to update \mathbf{z}_{best} . Specifically, for the i th individual (i.e., $z_{best,i}$), DE/rand/1 is used to generate an offspring u_i :

$$u_i = z_{best,r1} + K(z_{best,r2} - z_{best,r3}) \quad (15)$$

where $z_{best,r1}$, $z_{best,r2}$, and $z_{best,r3}$ are three different individuals selected from $\{z_{best,1}, \dots, z_{best,m}\}/z_{best,i}$ randomly.

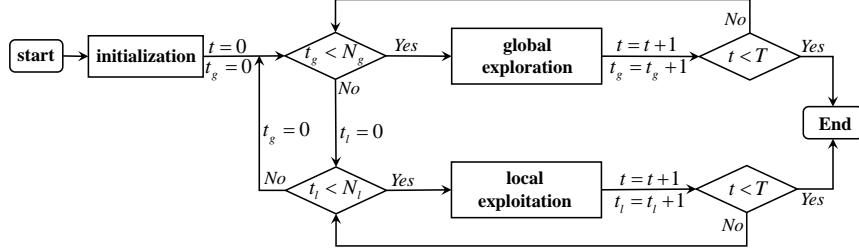


Fig. 1. Framework of DETEM.

Algorithm 2: Local Exploitation

```

1  $t_l \leftarrow 0$ ;
2 while  $t_l < N_l$  do
3   for  $i = 1 : m$  do
4     | Generate  $u_i$  for  $z_{best,i}$  according to Eq. (15);
5   end
6   for  $i = 1 : m$  do
7     | Replace a randomly selected dimension of  $z_{best}$  with  $u_i$  to generate
     | a new solution  $\hat{z}_{best}$ ;
8     | Evaluate  $\hat{z}_{best}$  according to Eq. (12);
9     | if  $\hat{z}_{best}$  is better than  $z_{best}$  then
10       | | Replace  $z_{best}$  with  $\hat{z}_{best}$ ;
11   end
12    $t_l \leftarrow t_l + 1$ ;
13 end
14 Output  $z_{best}$ ;

```

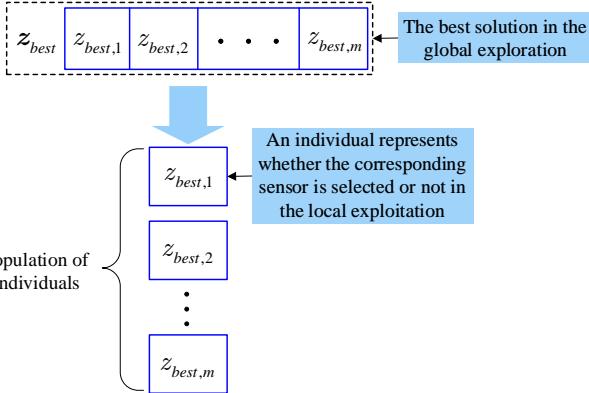


Fig. 3. The new encoding mechanism used for the local exploitation.

Since each individual has only one dimension, the crossover is omitted. Afterward, u_i is used to replace a randomly selected dimension of z_{best} ; thus, a new solution \hat{z}_{best} is obtained. If \hat{z}_{best} is better than z_{best} , it replaces z_{best} . The details of the local exploitation are described in Algorithm 2.

5) *Solution repair:* Note that the problem described in Eq. (12) has two constraints. To satisfy these two constraints, we repair a solution (i.e., z) as follows. Since each dimension of z is a value between 0 and 1, we use the rounding function to set it as the integer which is the closest to it. In this manner, the second constraint (i.e., $z_i \in \{0, 1\}$) can be satisfied. The other constraint is to guarantee that the number of 1 in z is equal to k . For convenience, we change the values of some randomly selected dimensions from 0 to 1, if the number of 1 is smaller than k . Inversely, we change the values of some randomly selected dimensions from 1 to 0. As a result, the first constraint can be satisfied.

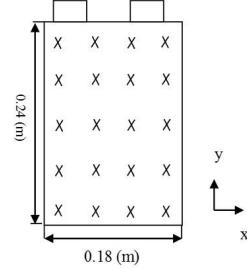


Fig. 4. Schematic of the battery.

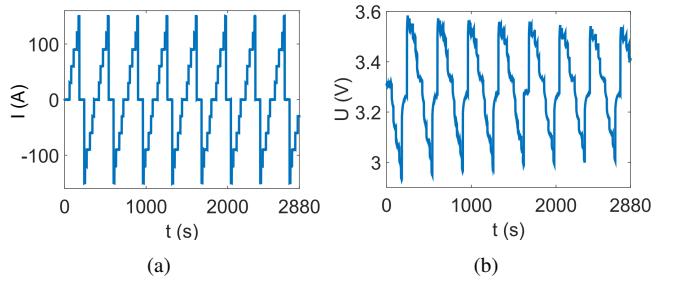


Fig. 5. Input current and output voltage: (a) input current, (b) output voltage.

6) *Main advantages of DETEM:* The main advantages of DETEM are summarized as follows:

- DETEM can be used to solve the sensor placement problem for spatiotemporal modeling of various types of DPSs because it does not require the objective function to be convex or submodular.
- Due to the one-to-one spawning and selection mechanism, DETEM would own more powerful search capability compared with the classical GA and some other EAs.
- Since DETEM does not add any complex operators, it is as efficient as the classical DE which has a compact structure and is easy to be implemented.
- In the new encoding mechanism, each individual contains one decision variable and the dimension of the search space is thus equal to one, regardless of the number of sensors. Thus, it can accelerate the optimization process significantly.
- Since two kinds of encoding mechanisms are combined compactly, DETEM can achieve a tradeoff between exploration and exploitation. Thus, it can find a satisfying solution effectively and quickly.

In next sections, we will verify the performance of DETEM on the thermal process of a lithium-ion battery.

TABLE I
RMSES OF MODELING THE THERMAL PROCESS OF A SIMULATED BATTERY BY GA, THE RL-BASED METHOD, AND DETEM.

$k \backslash$ method	GA	RL-based method	DETEM
6	0.0622	0.0635	0.0344
8	0.0553	0.0554	0.0294
10	0.0506	0.0507	0.0236

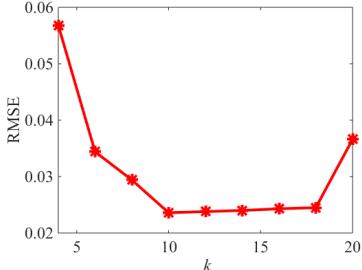


Fig. 6. RMSEs with respect to different values of k .

IV. SIMULATION STUDIES AND DISCUSSIONS

A. Simulation Setup

The thermal process of a lithium-ion battery is a typical kind of DPS. We applied the proposed method to place sensors for spatiotemporal modeling of the thermal process. First, we constructed an electrochemical-thermal model by a software package (i.e., COMSOL). The settings such as materials and parameters in the simulated model were the same as a real LiFePO₄/graphite battery. The schematic of the simulated battery is shown in Fig. 4. The popular KL decomposition method [7] was used for spatiotemporal modeling. As the same in [9], a multi-step input current was used to excite the thermal process adequately. The input current and the corresponding output voltage are described in Fig. 5. To sample data, a method was used to select k placements from 20 uniformly distributed locations to place sensors.

Specifically, 2880 snapshots were sampled where the first 1200 snapshots were used for modeling, the following 1200 ones were used for validation, and the remaining ones were used for testing. The sampling interval was 1 s. To verify the performance of the proposed method, we compared it with GA and the RL-based method in [11]. Similar to [11], the root of mean square error (RMSE) was used for comparisons:

$$\text{RMSE} = \sqrt{\frac{1}{kl} \sum_{i=1}^k \sum_{j=1}^l (y_{i,j} - \hat{y}_{i,j})^2} \quad (16)$$

where k is the number of sensors, l is the size of the testing set, $\{y_{i,j}\}_{i=1,j=1}^{i=k,j=l}$ is the testing set, and $\{\hat{y}_{i,j}\}_{i=1,j=1}^{i=k,j=l}$ includes the outputs of the analytical model. All methods were repeated 20 runs independently and the average modeling RMSE was used for comparison.

Some other parameters were set as follows. In the KL decomposition method, the number of DSBFs was set to 5. Both the maximal output and input lags were set to 1. In DETEM, the population size was set to 20. N_g and N_l were set to 3 and 10, respectively. T was set to 100. Both K and CR were set to 0.9 according to [27].

TABLE II
RMSES OF MODELING THE THERMAL PROCESS OF A SIMULATED BATTERY BY THE CLASSICAL DE, DEEM, AND DETEM.

$k \backslash$ method	classical DE	DEEM	DETEM
6	0.0570	0.0600	0.0344
8	0.0477	0.0482	0.0294
10	0.0403	0.0415	0.0236

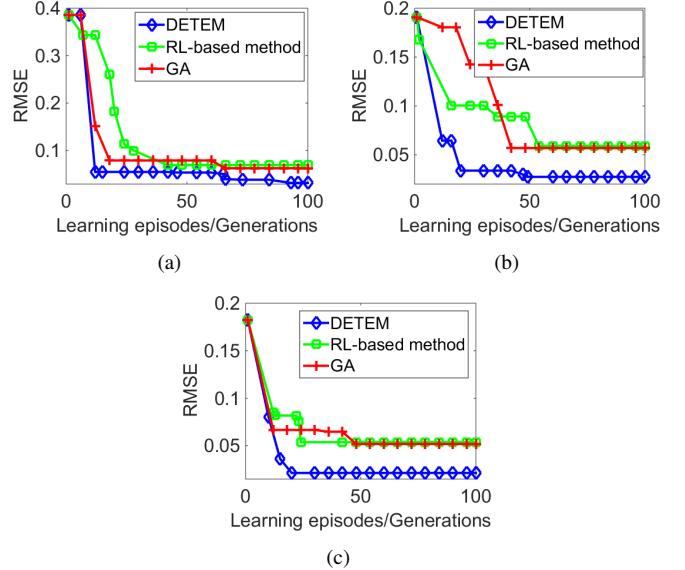


Fig. 7. Convergence graphs of DETEM, the RL-based method, and GA: (a) $k = 6$, (b) $k = 8$, (c) $k = 10$.

Remark: As described in [32], in practice, the best number of sensors (i.e., k) should be determined beforehand. A too small value of k would lose much process information; thus, the analytical model built based on the sampled data would not be accurate. Inversely, a too big value of k can easily lead to redundant configuration, thus wasting resources [33]. In order to decide the best value of k , we plotted a curve of modeling RMSEs with respect to different values of k . As shown in Fig. 6, $k = 10$ performs better than others. It reflects that $k = 10$ would be the best number of sensors. Thus, we compared our method with other methods in the case of $k = 10$. Besides, in many situations, only few sensors can be available. It is important to test the performance of a method when using a small number of sensors. Thus, we also compared our method with other methods in the case of $k < 10$. Specifically, $k = 6$ and $k = 8$ were considered in the simulations and experiments.

B. Comparisons and Discussions

1) *Performance comparison:* The modeling RMSEs of different methods are summarized in Table I. As shown in Table I, DETEM can obtain the best accuracy. In all cases, compared with the RL-based method and GA, DETEM reduces the modeling RMSE by nearly 50%. The convergence graphs of different methods are given in Fig. 7. As shown in Fig. 7, DETEM converges faster than the other two methods in all cases. The results verify that DETEM is effective and efficient. Additionally, we plotted the modeling RMSEs of

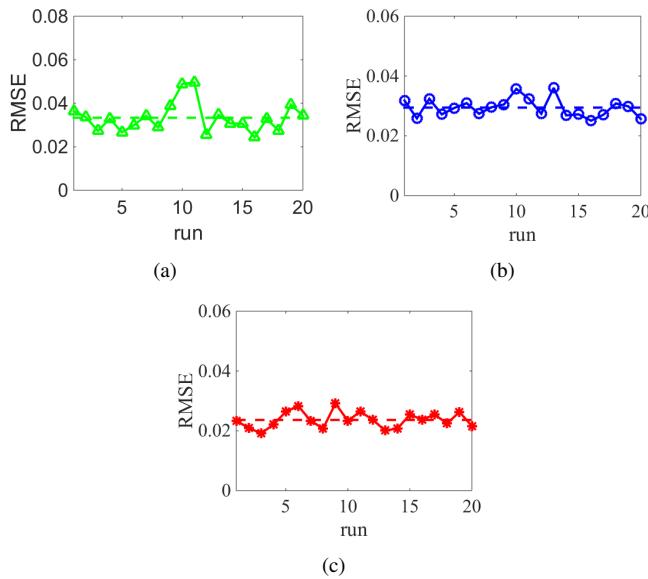


Fig. 8. RMSEs of DETEM with respect to 20 independent runs: (a) $k = 6$, (b) $k = 8$, (c) $k = 10$.

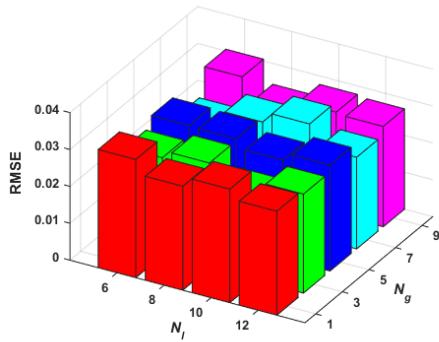


Fig. 9. Modeling RMSEs of different combinations of N_g and N_l in the case of $k = 10$.

DETEM with respect to 20 independent runs in Fig. 8. In each run, a different initial population was used. As shown in Fig. 8, in all cases (i.e., $k=6$, 8, and 10), all 20 RMSEs are close to the average value. It indicates that DETEM is not sensitive to the initial population.

2) *Effectiveness of different encoding mechanisms:* In order to investigate the effectiveness of different encoding mechanisms, we compared DETEM with the classical DE and DEEM [27]. In the classical DE, the conventional encoding mechanism was used. On the contrary, the new encoding mechanism was used in DEEM. Note that the scaling factor and the crossover control parameter were the same in these three methods. The modeling RMSEs of these three methods are summarized in Table II. As shown in Table II, DETEM performs better than the classical DE and DEEM in terms of modeling RMSE. The results reflect that both of these two kinds of encoding mechanisms are critical to DETEM.

3) *Effectiveness of the objective function:* As described in Section III-A, compared with the RL-based method, the proposed objective function (i.e., $F(\mathbf{z})$) takes the temporal error into consideration. To verify the effectiveness of $F(\mathbf{z})$,

TABLE III
SOME NOMINAL PARAMETERS OF THE BATTERY.

parameter	value
thickness	7.83 mm
capacity	60 Ah
length	240 mm
voltage	3.2 V
width	180 mm
temperature	20 °C
charge cut-off voltage	3.65 V
discharge cut-off voltage	2 V

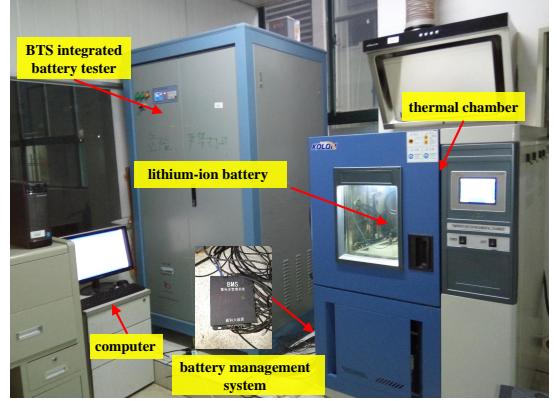


Fig. 10. Experiment platform.

we tested DETEM by using the spatial objective function (i.e., $F_s(\mathbf{z})$) only. The modeling RMSEs in the cases of $k = 6$, $k = 8$, and $k = 10$ are 0.0552, 0.0493, and 0.0358, respectively. It implies that the performance of the proposed method can be improved by considering the temporal error.

4) *Sensitivity of N_g and N_l :* In DETEM, two parameters (i.e., N_g and N_l) should be set manually. In order to investigate their sensitivity, we tested DETEM with different combinations of N_g and N_l . The modeling RMSEs of different parameters are described in Fig. 9. As shown in Fig. 9, DETEM with all combinations of N_g and N_l perform similarly. A proper value of N_l can be selected from {6, 8, 10, 12} while that of N_g can be selected from {1, 3, 5, 7, 9}. Additionally, the combination of $N_g = 3$ and $N_l = 10$ achieves the best performance. Thus, in the simulations and experiments, we set N_g and N_l to 3 and 10, respectively.

V. EXPERIMENT VALIDATION

A. Experiment Setup

To further verify the performance of the proposed method, we used it to place sensors for sampling data from a real battery. Afterward, the sampled data was used to model the thermal process. In the experiment, thermocouple sensors were used to sample data from a pouch-type LiFePO₄/graphite lithium-ion battery. Some nominal parameters of the battery are given in Table III. The platform used for the experiment is described in Fig. 10. As shown in Fig. 5 (a), the battery thermal system (BTS) integrated battery tester was used to generate the input current. The corresponding output voltage is described in Fig. 5 (b). The lithium-ion battery was placed in the thermal chamber and its ambient temperature was kept as 20 °C. k

TABLE IV
RMSEs OF MODELING THE THERMAL PROCESS OF A REAL BATTERY BY GA, THE RL-BASED METHOD, AND DETEM.

k	method	GA	RL-based method	DETEM
6		0.1360	0.1420	0.0790
8		0.0985	0.0888	0.0590
10		0.0916	0.0825	0.0565

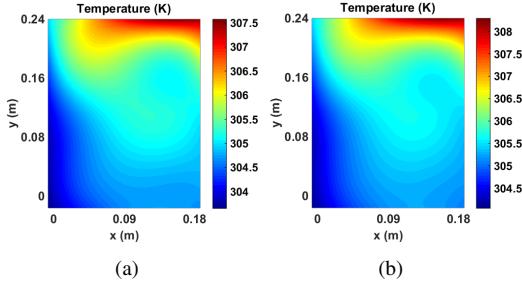


Fig. 11. Measured temperature distribution at 2560 s and 2880 s: (a) 2560 s, (b) 2880 s.

sensors were placed on the surface of the battery and the battery management system was used to collect temperature data. As the same in Section IV, 2880 snapshots were sampled. The performance of DETEM, GA, and the RL-based method were compared based on these snapshots.

B. Experiment Results

The modeling RMSEs of GA, the RL-based method, and DETEM are summarized in Table IV. As shown in Table IV, DETEM performs better than GA and the RL-based method. In all cases, DETEM improves the accuracy by nearly 50%. Additionally, the modeling RMSEs of all methods in the experiment are larger than that in the simulation. The reason may be that more disturbances are involved in the experiment. In summary, the results reflect that DETEM is effective to place sensors for spatiotemporal modeling of DPSs.

To further investigate the advantages of the proposed method, we compared it with the RL-based method in terms of the distributed prediction error. The measured temperature distributions at 2560 s and 2880 s are given in Fig. 11. The sensor layouts obtained by DETEM and the RL-based method over a typical run are described in Figs. 12-14 where a red circle denotes a sensor. The prediction errors of these two methods in the cases of $k = 6$, $k = 8$, and $k = 10$ are described in Figs. 15-17. The results show that the proposed method can obtain more accurate temperature distributions than the RL-based method in all cases. In summary, the proposed method is able to place sensors optimally for modeling the thermal process of a lithium-ion battery.

VI. CONCLUSIONS

In this paper, we made the first attempt to use the merits of DE to place sensors optimally for spatiotemporal modeling of the thermal process of a lithium-ion battery. First, an objective function was constructed by taking both the spatial error and the temporal error into consideration. Afterward, a

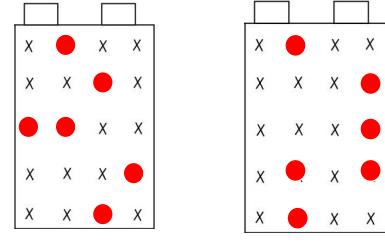


Fig. 12. Sensor layouts obtained by DETEM and the RL-based method over a typical run in the case of $k = 6$: (a) DETEM, (b) the RL-based method.

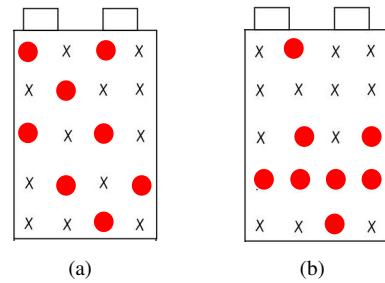


Fig. 13. Sensor layouts obtained by DETEM and the RL-based method over a typical run in the case of $k = 8$: (a) DETEM, (b) the RL-based method.

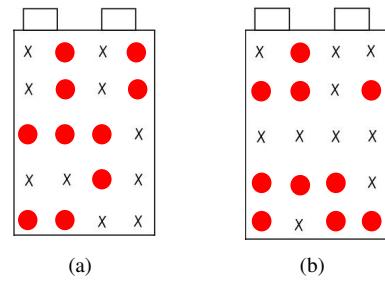


Fig. 14. Sensor layouts obtained by DETEM and the RL-based method over a typical run in the case of $k = 10$: (a) DETEM, (b) the RL-based method.

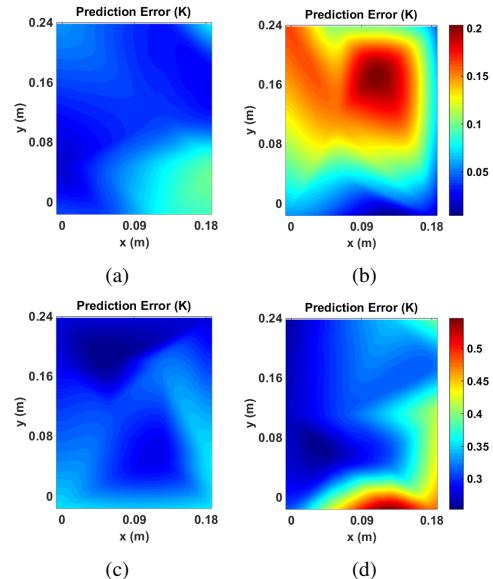


Fig. 15. Distributed prediction error in the case of $k = 6$: (a) DETEM at 2560 s, (b) the RL-based method at 2560 s, (c) DETEM at 2880 s, (d) the RL-based method at 2880 s.

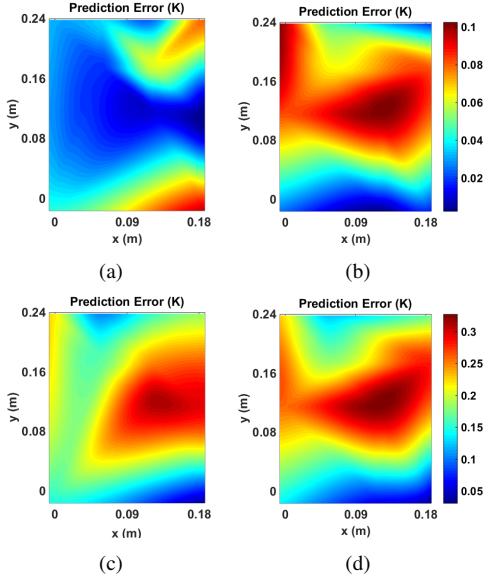


Fig. 16. Distributed prediction error in the case of $k = 8$: (a) DETEM at 2560 s, (b) the RL-based method at 2560 s, (c) DETEM at 2880 s, (d) the RL-based method at 2880 s.

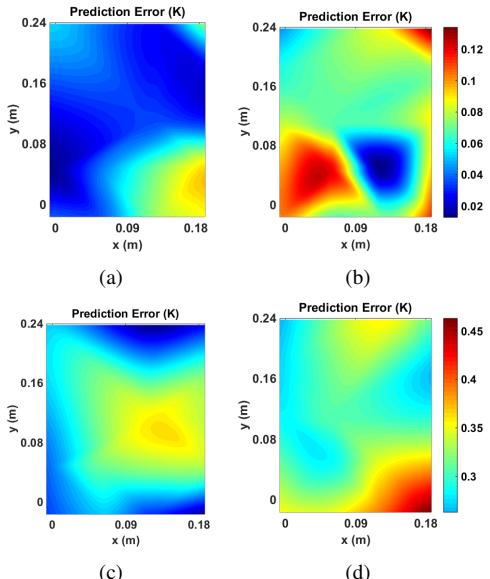


Fig. 17. Distributed prediction error in the case of $k = 10$: (a) DETEM at 2560 s, (b) the RL-based method at 2560 s, (c) DETEM at 2880 s, (d) the RL-based method at 2880 s.

novel DE algorithm with two kinds of encoding mechanisms was designed to optimize the objective function, where the conventional mechanism and a new mechanism focused on global exploration and local exploitation, respectively. The proposed method is easy to implement and does not impose any limitations on the objective function. Thus, it could be used in diverse areas. Simulations and experiments on the thermal process of a lithium-ion battery showed that the proposed method performs better than GA and a recently proposed RL-based method. In the future, we will focus on designing an efficient algorithm that can place a sensor anywhere not limited to the given locations.

REFERENCES

- [1] B.-C. Wang, H.-X. Li, and H.-D. Yang, "Spatial correlation-based incremental learning for spatiotemporal modeling of battery thermal process," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 2885–2893, 2019.
- [2] H. T. Banks and K. Kunisch, *Estimation techniques for distributed parameter systems*. Springer Science & Business Media, 2012.
- [3] Z. Wang and H.-X. Li, "Incremental spatiotemporal learning for online modeling of distributed parameter systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 12, pp. 2612–2622, 2018.
- [4] D. A. Kopriva, *Implementing spectral methods for partial differential equations: Algorithms for scientists and engineers*. Springer Science & Business Media, 2009.
- [5] H. Deng, H.-X. Li, and G. Chen, "Spectral-approximation-based intelligent modeling for distributed thermal processes," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 5, pp. 686–700, 2005.
- [6] C. Qi, H.-X. Li, S. Li, X. Zhao, and F. Gao, "Probabilistic pca-based spatiotemporal multimodeling for nonlinear distributed parameter processes," *Industrial & Engineering Chemistry Research*, vol. 51, no. 19, pp. 6811–6822, 2012.
- [7] H. Park and D. Cho, "The use of the karhunen-loeve decomposition for the modeling of distributed parameter systems," *Chemical Engineering Science*, vol. 51, no. 1, pp. 81–98, 1996.
- [8] X. Lu, F. Yin, C. Liu, and M. Huang, "Online spatiotemporal extreme learning machine for complex time-varying distributed parameter systems," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1753–1762, 2017.
- [9] K.-K. Xu, H.-X. Li, and Z. Liu, "Isomap-based spatiotemporal modeling for lithium-ion battery thermal process," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 569–577, 2017.
- [10] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1135–1146, 2014.
- [11] Z. Wang, H.-X. Li, and C. Chen, "Reinforcement learning-based optimal sensor placement for spatiotemporal modeling," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2861–2871, 2019.
- [12] H. Rowaihy, S. Eswaran, M. Johnson, D. Verma, A. Bar-Noy, T. Brown, and T. La Porta, "A survey of sensor selection schemes in wireless sensor networks," in *Unattended Ground, Sea, and Air Sensor Technologies and Applications IX*, vol. 6562. International Society for Optics and Photonics, 2007, p. 65621A.
- [13] S. T. Jawaaid and S. L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282–288, 2015.
- [14] X. Shen and P. K. Varshney, "Sensor selection based on generalized information gain for target tracking in large sensor networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 363–375, 2013.
- [15] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2008.
- [16] Y. Mo, R. Ambrosino, and B. Sinopoli, "Sensor selection strategies for state estimation in energy constrained wireless sensor networks," *Automatica*, vol. 47, no. 7, pp. 1330–1338, 2011.
- [17] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms," *Automatica*, vol. 78, pp. 202–210, 2017.
- [18] R. Semaan, "Optimal sensor placement using machine learning," *Computers & Fluids*, vol. 159, pp. 167–176, 2017.
- [19] M. Naeem, U. Pareek, and D. C. Lee, "Swarm intelligence for sensor selection problems," *IEEE Sensors Journal*, vol. 12, no. 8, pp. 2577–2585, 2012.
- [20] Y. Xu, O. Ding, R. Qu, and K. Li, "Hybrid multi-objective evolutionary algorithms based on decomposition for wireless sensor network coverage optimization," *Applied Soft Computing*, vol. 68, pp. 268–282, 2018.
- [21] J. Uber, R. Janke, R. Murray, and P. Meyer, "Greedy heuristic methods for locating water quality sensors in distribution systems," in *Critical Transitions in Water and Environmental Resources Management*, 2004, pp. 1–9.
- [22] R. Rajagopalan, R. Niu, C. K. Mohan, P. K. Varshney, and A. L. Drozd, "Sensor placement algorithms for target localization in sensor networks," in *2008 IEEE Radar Conference*. IEEE, 2008, pp. 1–6.
- [23] C. Chisari, L. Macorini, C. Amadio, and B. A. Izzuddin, "Optimal sensor placement for structural parameter identification," *Structural and Multidisciplinary Optimization*, vol. 55, no. 2, pp. 647–662, 2017.

- [24] G. F. Gomes, S. S. da Cunha, P. d. S. L. Alexandrino, B. S. de Sousa, and A. C. Acelotti, "Sensor placement optimization applied to laminated composite plates under vibration," *Structural and Multidisciplinary Optimization*, vol. 58, no. 5, pp. 2099–2118, 2018.
- [25] G. F. Gomes, F. A. de Almeida, P. d. S. L. Alexandrino, S. S. da Cunha, B. S. de Sousa, and A. C. Acelotti, "A multiobjective sensor placement optimization for shm systems considering fisher information matrix and mode shape interpolation," *Engineering with Computers*, vol. 35, no. 2, pp. 519–535, 2019.
- [26] G. F. Gomes and J. V. P. Pereira, "Sensor placement optimization and damage identification in a fuselage structure using inverse modal problem and firefly algorithm," *Evolutionary Intelligence*, vol. 13, no. 4, pp. 571–591, 2020.
- [27] Y. Wang, H. Liu, H. Long, Z. Zhang, and S. Yang, "Differential evolution with a new encoding mechanism for optimizing wind farm layout," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1040–1054, 2017.
- [28] P. Huang, Y. Wang, K. Wang, and K. Yang, "Differential evolution with a variable population size for deployment optimization in a uav-assisted iot data collection system," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 324–335, 2020.
- [29] C. Yang, "An adaptive sensor placement algorithm for structural health monitoring based on multi-objective iterative optimization using weight factor updating," *Mechanical Systems and Signal Processing*, vol. 151, p. 107363, 2021.
- [30] S. Das, S. S. Mullick, and P. Suganthan, "Recent advances in differential evolution—an updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.
- [31] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 107–125, 2008.
- [32] C. Yang, K. Liang, and X. Zhang, "Strategy for sensor number determination and placement optimization with incomplete information based on interval possibility model and clustering avoidance distribution index," *Computer Methods in Applied Mechanics and Engineering*, vol. 366, pp. 1–21, 2020.
- [33] C. Yang, K. Liang, X. Zhang, and X. Geng, "Sensor placement algorithm for structural health monitoring with redundancy elimination model based on sub-clustering strategy," *Mechanical Systems and Signal Processing*, vol. 124, pp. 369–387, 2019.



Bing-Chuan Wang received the B.Eng. degree in automation and the M.S. degree in control science and engineering from the Central South University, Changsha, China, in 2013 and 2016, respectively, and the Ph.D. degree in system engineering and engineering management from the City University of Hong Kong, Hong Kong, in 2019.

Dr. Wang is currently with the School of Automation, Central South University, Changsha, China. His current research interests include intelligent optimization, data-driven modeling, and their applications to the management of lithium-ion batteries.



Yong Wang (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the Central South University, Changsha, China, in 2011.

He is a Professor with the School of Automation, Central South University, Changsha, China. His current research interests include intelligent learning and optimization and their interdisciplinary applications.

Dr. Wang is an Associate Editor of the *IEEE Transactions on Evolutionary Computation* and the *Swarm and Evolutionary Computation*. He was a recipient of Cheung Kong Young Scholar by the Ministry of Education, China, in 2018, and a Web of Science highly cited researcher in Computer Science in 2017 and 2018.



Shi-Hui He received the B.S. degree in computer science and technology from the Central South University of Forestry and Technology, Changsha, China, in 2018. She is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Central South University, Changsha, China.

Her current research interests include computational intelligence, machine learning, and their applications to real-world problems.