Green Energy Forecast-Based Bi-Objective Scheduling of Tasks Across Distributed Clouds

Jing Bi, *Senior Member, IEEE*, Haitao Yuan, *Senior Member, IEEE*, Jia Zhang, *Senior Member, IEEE*, and MengChu Zhou, *Fellow, IEEE*

Abstract—Current large-scale green cloud data centers (GCDCs) tend to consume a huge amount of energy and generate enormous carbon emissions. Existing studies have tried to solve this problem by either realizing prediction of green energy, or optimizing task scheduling. In contrast, this work seamlessly combines green energy prediction and task scheduling to jointly optimize revenue and energy cost of GCDCs. Specifically, this work designs a prediction method, named Savitzky-Golay and Long Short-Term Memory network (SG-LSTM), to realize noise filtering and forecast green energy. Based on such prediction, a bi-objective optimization method, named Decomposition-based Multi-objective evolutionary algorithm with Gaussian mutation and Crowding distance (DMGC), is developed to optimize the revenue and energy cost of GCDCs. Its performance is demonstrated over real-life datasets including Google cluster traces, wind speeds, solar irradiance and prices of electricity. Experimental results show that SG-LSTM outperforms its two peers, back propagation neural network and gated recurrent unit, in terms of root mean square errors and mean absolute errors. In addition, DMGC surpasses its such peers as NSGA-II, SPEA2, and MOEA/D in terms of revenue, energy cost and average execution time. Particularly, DMGC's revenue is 18%, 20% and 13.1% higher, energy cost is 16%, 19.8% and 15.2% lower, and average execution time is 60.02%, 38.47% and 24.17% lower than those of NSGA-II, SPEA2, and MOEA/D, respectively.

Index Terms—Green clouds, intelligent optimization, machine learning, task scheduling, Savitzky-Golay filter, recurrent neural network, multi-objective optimization algorithms

1 Introduction

In recent years, various compute-intensive and data-intensive applications produce, analyze, and store a great amount of data in cloud data centers (CDCs) [1], [2]. The data processing in CDCs thus requires tremendous energy. If only thermal power is used to provide electricity, the energy cost will be extremely high and CDCs will produce a large amount of carbon dioxide to our environment. With the growing number of large-scale CDCs being deployed around the world, an imperative demand and challenge is to reduce high energy cost and environmental pollution [3], [4]. As an emerging solution, most CDCs gradually evolve to green CDCs (GCDCs) by adopting renewable energy facilities [5]. To reduce the energy cost of GCDCs, existing studies mainly adopt high-accuracy green energy prediction or energy-efficient task scheduling techniques.

Studies on the high-accuracy prediction methods of green energy are becoming essential, because they help to realize energy-efficient task scheduling for multiple GCDCs [6]. Various prediction models on green energy time series have been proposed in the literature, including back propagation neural network (BPNN) [7], artificial neural network (ANN) [8], gated recurrent unit (GRU), and support-vector regression (SVR) [9]. Anamika *et al*. [10] predict the global monthly average solar radiation with ANN. Jiang *et al*. [11] combine a traditional BPNN with a representative unit approach to forecast a short-term wind energy time series. Ahmed *et al*. [9] design an SVR-based method for predicting wind speed. However, these methods have limited feature learning abilities and may be easily trapped into local minima. In recent years, deep learning has been widely adopted in the green energy prediction field. Many researchers adopt a long short-term memory (LSTM) network, which is known for realizing high-accuracy prediction of large-scale time series [12]. For example, Dong *et al*. [13] design a prediction method for wind power time series with the LSTM. Its feasibility and effectiveness are verified, and the prediction accuracy of wind power is significantly improved compared with traditional methods. Afrasiabi *et al*. [14] adopt a convolutional neural network to increase spatial feature learning, and a GRU to capture temporal features of wind speed.

In addition, there are many studies on energy-efficient task scheduling for GCDCs. Similar to [15], [16], this work focuses on delay sensitive applications, e.g., network I/O apps, video conferences, and online games. In addition, in

- *Jing Bi is with the School of Software Engineering in Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China. E-mail: bijing@bjut.edu.cn.*
- *Haitao Yuan is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China. E-mail: yuan@buaa.edu.cn.*
- *Jia Zhang is with the Department of Computer Science, School of Engineering, Southern Methodist University, Dallas, TX 75205 USA. E-mail: jiazhang@smu.edu.*
- *MengChu Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA. E-mail: zhou@njit.edu.*

Manuscript received 10 May 2021; revised 21 Sept. 2021; accepted 30 Oct. 2021. Date of publication 4 Nov. 2021; date of current version 8 Sept. 2022.
This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62173013, 62073005 and 61802015, and in part by the Major Science and Technology Program for Water Pollution Control and Treatment of China under Grant 2018ZX07111005.
(Corresponding author: Haitao Yuan.)
Recommended for acceptance by P. Bouvry.
Digital Object Identifier no. 10.1109/TSUSC.2021.3124893

GCDCs, each application is usually deployed in separate virtual machines, and the types of its tasks from users can be identified from network packets. Thus, a task scheduler can directly determine different applications in GCDCs, and forward tasks to their corresponding virtual machines for further processing. The work in [15] proposes a profit maximization approach to well balance energy cost and revenue of a single GCDC. Specifically, a convex optimization problem is formulated for one GCDC, and solved by convex programming techniques, e.g., interior point methods. Shao *et al.* [16] consider load distribution for CDCs to minimize the total energy cost for multiple distributed CDCs in a market, where their electricity prices vary from location to location. It is shown that the total amount of available green energy in GCDCs is often insufficient, and it is highly affected by weather conditions. As a result, current GCDCs still mainly rely on power grid [17]. Therefore, this work adopts three types of energy sources, i.e., wind energy, solar energy, and power grid, to power GCDCs.

To date, there have been several studies focusing on either energy prediction [6], [9], [10], [11], [13] or resource optimization [15], [16], [17] in GCDCs. Our hypothesis is that, considering these two objectives together may lead to better results for both aspects simultaneously. This idea directly motivates this work, and our proposed methods well demonstrate the importance of prediction accuracy of green energy on the scheduling of tasks in GCDCs. Trying to tackle the two issues simultaneously can jointly increase the revenue of GCDCs and reduce their energy cost in much lower execution time. The combination of a time series prediction method and a bi-objective task scheduling one should help GCDC providers achieve more intelligent and energy-efficient task scheduling than those without such combination. Our work aims to fill the gap between energy prediction and resource optimization for better performance, which has not been done in existing studies of GCDC operations. It reveals the impact of prediction quality of green energy on bi-objective GCDC task scheduling performance.

The contributions of this work are summarized in four-fold:

1) A deep neural network called Savitzky-Golay (SG)-LSTM is designed to predict green energy in distributed GCDCs. It provides fundamental data support for intelligent task scheduling.
2) A comprehensive set of factors (including service level agreements (SLAs) [2] of tasks, task arriving rates, service ones and loss probability of tasks) are simulated to make our designed model close to real-life GCDCs powered by different energy sources, which yields realistic experimental results.
3) An improved bi-objective optimization algorithm is developed to make a good trade-off between revenue gain and energy cost of GCDCs. It is named Decomposition-based Multi-objective evolutionary method with Gaussian mutation and Crowding distance (DMGC).
4) This work seamlessly combines the power of the prediction and scheduling technologies to achieve the better results than other widely-known methods. An Savitzky-Golay LSGM (SG-LSTM) model is developed to predict the available renewable energy in each GCDC, while DMGC simultaneously optimizes the energy cost and revenue for multiple GCDCs.

The remainder of the article is organized as follows. Section 2 discusses the related work. Section 3 presents the motivation, and an overall architecture of GCDCs and formulates a bi-objective optimization problem. SG-LSTM and DMGC are described in Section 4. Section 5 presents the results of performance evaluation of designed methods over real-life traces and green energy data. Finally, Section 6 concludes the work.

2 Related Work

In this section, we compare our work with the literature in the aspects of task scheduling, green energy management in data centers, and green energy prediction.

2.1 Task Scheduling

Recently, scheduling of tasks in data centers has received an increasing amount of attention [4], [5], [19], [20], [21], [22]. Hu *et al.* [4] design an algorithm of task scheduling to reduce both cost of network, and total time of data-parallel applications across multiple data centers located in different sites. An integer linear program is formulated, and its solution achieves the improvement of both job completion and cost efficiency. Yuan *et al.* [5] design a spatio-temporal scheduling method to manage tasks within their response time limits in a cost-effective way. It considers temporal and spatial variations for energy cost minimization, which is achieved by combing particle swarm optimization, simulated annealing and genetic algorithm. Kanemitsu *et al.* [19] present a clustering-based algorithm to schedule tasks for the schedule length minimization in heterogeneous processors. It gives the lower limit of the whole running time in each single processor, by considering applications and characteristics of a system. Task assignment and clustering are realized to minimize the length of schedule. Varshney and Simmhan [20] propose several scheduling approaches for many tasks with delay bound limits on virtual machines in clouds to decrease the cost. The timely completion is guaranteed by determining price, the number of virtual machines, task placement, migration and checkpointing. Mashayekhy *et al.* [21] design a method for improving the energy efficiency of MapReduce applications, while meeting SLA. The energy-aware job scheduling is modeled as an integer program for minimizing the consumption of energy. Zhang *et al.* [22] present a two-stage approach to classify and match tasks with virtual machines, thereby balancing workload among cloud resources and minimizing user cost.

Different from all of the aforementioned methods, this work adopts an improved algorithm of bi-objective optimization to simultaneously reduce energy cost and increase revenue of GCDCs. To the best of our knowledge, we are the first one to form such a problem and provide a solution. Our approach jointly optimizes the energy cost and revenue of GCDCs for a good trade-off between them, by considering a holistic collection of parameters common to distributed GCDCs.

TABLE 1
Comparison of Our Work and its Related Studies (✓ Means Considered While × Means Not)

Study/Parameter	Green energy	Distributed CDCs	Bi-objective optimization	Revenue modeling	Long-term dependence	Data filtering
[4]	×	✓	✓	×	×	×
[5]	✓	✓	×	×	×	×
[13]	✓	×	×	×	✓	×
[19]	×	✓	×	×	×	×
[20]	×	✓	×	×	×	×
[21]	×	✓	×	×	×	×
[22]	×	✓	×	×	×	×
[23]	✓	×	✓	×	×	×
[24]	✓	✓	✓	×	×	×
[25]	✓	✓	×	×	×	×
[26]	×	✓	✓	×	×	×
[27]	✓	×	✓	×	×	×
[28]	✓	×	×	×	×	×
[29]	✓	×	×	×	×	×
[30]	✓	×	×	×	✓	✓
[31]	✓	×	×	×	×	×
[32]	✓	×	×	×	×	✓
Our work	✓	✓	✓	✓	✓	✓

2.2 Green Energy Management in Data Centers

More emerging studies have been conducted to achieve energy optimization with green energy in distributed GCDCs [23], [24], [25], [26], [27]. The work in [23] proposes a game-theoretic method to minimize the elasticity cost, by provisioning resources in a green mobile cloud environment. They present the modeling of overhead, processing time, and energy in opportunistic offloading for the mobile cloud. A utility-based elasticity cost and profit calculation method is also given for resource utilization maximization. Nan *et al.* [24] adopt hybrid renewable energy sources to support fog nodes that use solar power and grid power as primary and backup supplies. They propose an online algorithm by using Lyapunov optimization, to gain a trade-off between average cost and average response time. Pan and Chen [25] propose a transmission rate scheduling scheme for energy conservation in mobile cloud computing, through a Lyapunov optimization approach. A closed-form formula for the transmission rate is given. Chou *et al.* [26] propose an energy-saving resource allocation approach with an algorithm of particle swarm optimization. It considers the energy efficiency ratio of air conditioners, and energy consumption of virtual and physical machines. Higuera *et al.* [27] design a power model for time-limited servers in data centers, by considering the temperature dependency and static energy consumption. It is used to optimize workload assignment, and dynamic voltage and frequency scaling.

In contrast to the aforementioned work, we propose an LSTM-based time series approach to realize the prediction of solar irradiance and wind speed in multiple GCDCs in future time slots. We then integrate the prediction of green energy with the improved bi-objective task scheduling to better manage GCDCs.

2.3 Green Energy Prediction

Many studies have been conducted to predict green energy in GCDCs in recent years [28], [29], [30], [31], [32]. Park and Hur [28] design a spatial prediction method for green energy in power grids. Kriging methods and optimal spatial modeling are adopted to estimate wind and solar energy for long-term capacity planning of power grid. Li *et al.* [29] propose an algorithm based on multi-verse optimizer to realize the optimization of a support vector machine for predicting the amount of generated photovoltaic power. The chaotic sequences are adopted to realize the initialization of population and improve the algorithm convergence. Li *et al.* [30] propose a fuzzy approach for the prediction of wind speeds sampled in each hour. A structure of neural network is designed to integrate the advantages of both a fuzzy inference system and a neural network. A least square approach with optimized parameter learning is developed to minimize training errors. Zhang *et al.* [31] construct the fractional grey models with varying orders to analyze the uncertainty in wind speed. Then, a hybrid model of prediction designed with a neural network is proposed, and a method of support vector regression is designed to realize the scatter operation for wind speeds. Huang *et al.* [32] design a method for predicting combination probability of wind power with the correlation decision of area gray. The data of wind power is processed with energy-efficient decomposition of variational mode to decrease the data randomness. Then, ten covariance functions are used to establish several Gaussian process regression prediction models.

Different from those approaches, this work adopts a deep learning method to realize the prediction of time series of wind speed and solar radiation. It applies an SG filter method to the time series before LSTM-based prediction is conducted. It verifies the importance of prediction accuracy of green energy on the optimized scheduling of tasks in GCDCs, and combines data prediction and task scheduling optimization for energy-efficient and environment-friendly GCDCs. For clarity, Table 1 compares our work and its state-of-the-art studies in green energy, distributed CDCs, bi-objective optimization, revenue modeling, long-term dependence, and data filtering.

2.4 Comparison With Our Earlier Work

Compared with our prior work reported in [18], this article proposes an improved bi-objective optimization algorithm, DMGC, to realize high-performance task scheduling.

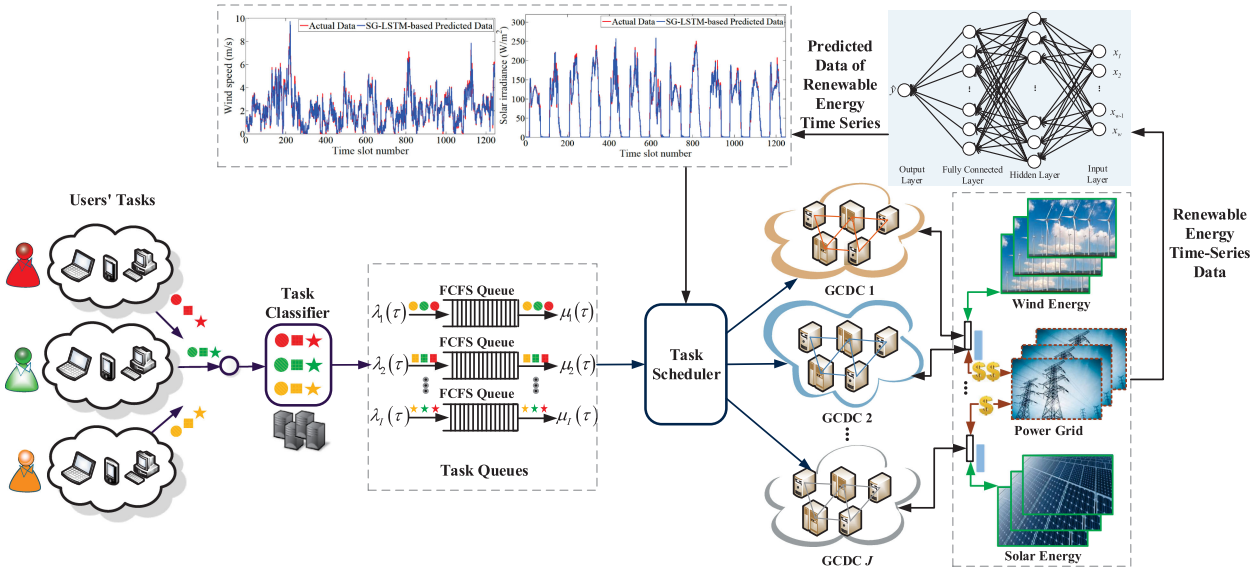


Fig. 1. System architecture of GCDCs.

Specifically, dynamic crossover mutation parameters, Gaussian mutation and crowding distance operations are newly introduced. Among them, dynamic parameters can overcome shortcomings of premature convergence and poor stability. In addition, the Gaussian mutation can introduce new genes into the population if individuals are trapped into local optima. Furthermore, the method of crowding degree evaluation based on crowding distance can make the distribution of the results more uniform, so as to obtain better solutions. By integrating the aforementioned new operations, our extensive experiments have demonstrated that a good trade-off between revenue and energy cost of GCDCs is achieved in this work.

3 Motivation and Problem Formulation

In this section, we will first illustrate a motivating example, then propose an overall architecture of GCDCs, and then introduce preliminaries, and finally formulate the problem. To reduce the emission of environment-polluting gas and the energy cost, present GCDCs usually consider the usage of green energy. In other words, a GCDC is typically supported by three types of supplies of power, i.e., solar, wind and grid power. To reduce the energy cost for the GCDC as much as possible, green renewable energy has the priority to be used.

Let us imagine one scenario in the context of simplified federated GCDCs, whose architecture is illustrated in Fig. 1. As shown in Fig. 1, three GCDCs are federated, and the configuration settings of the servers in each single GCDC follow those in [15]. Assume that based on some green energy prediction, at a certain time point t_0, the green power-based processing capacity of each GCDC is 3,000 due to limits of resources including wind and solar energy in each GCDC.

Assume that 9,000 tasks all arrive at t_0, closest to GCDC 1. If we only consider minimum communication delay, all tasks are sent to GCDC 1 for processing. Note that although GCDC 1 only possesses 3,000 processing capacity upon green power, the federation does carry a processing capacity to cover all 9,000 tasks using green power. Thus, an intelligent scheduler will choose to dispatch the incoming 9,000 tasks to three GCDCs, as shown in Fig. 1, in order for all available renewable energy in all three GCDCs to be fully utilized.

Meanwhile, how to dispatch the 9,000 over the three GCDCs require careful task scheduling. Tasks may feature different types and may require to be processed within certain time limits, i.e., Service Level Agreement (SLA). Note that each GCDC may have different computing power at the time. Therefore, optimized task scheduling may not only reduce delay and loss possibility of tasks, but also reduce cost and increase revenue of every GCDC.

This motivating example explains the necessity of green energy prediction and task scheduling. Fig. 1 also illustrates the overall system architecture of our proposed framework. To intelligently dispatch different types of tasks to GCDCs, this work first builds a prediction model based on deep neural networks by using historical data of solar radiation and wind speed at each GCDC. The purpose of this step aims to more accurately predict solar radiation and wind speed in future time slots; and green renewable energy in each GCDC is calculated. A centralized task scheduler of the federation can quickly and intelligently dispatch tasks based on the predicted green renewable energy data. The task scheduler is supported by a proposed algorithm, aiming to achieve both energy cost saving and gas emission reduction by efficiently and rationally using the renewable energy at each federated GCDC.

Next, we will start to introduce preliminaries in six aspects: SLAs, forecasting model, service rate, loss probability of tasks, solar and wind energy models, and objective functions. For clarity, all notations used throughout this article are summarized in Tables I and II in the Supplementary File, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TSUSC.2021.3124893.

3.1 SLAs

This work assumes that servers in each single GCDC are identical, i.e., they possess the same processing capacity for the same type of applications. In addition, we assume that the amount of renewable energy does not incur the cost after

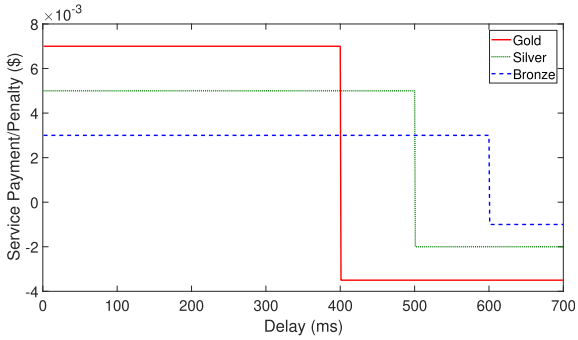


Fig. 2. Three types of SLAs.

facilities of green energy have been deployed and successfully run. Naturally, different GCDCs have different processing capabilities, and different loss probabilities of tasks for different applications, thereby leading to different ways to calculate penalties and revenues for GCDCs. Without losing generality, we consider three SLAs that are standard service protocols and are specified in terms of three parameters, i.e., α_i, β_i and D_i. Here, D_i is the specified delay bound of type i tasks, α_i is the fee paid by users if a task is finished within D_i, and β_i is the penalty fee paid by a GCDC provider if the completion time of a task is beyond D_i. Three types of SLAs are illustrated in Fig. 2: Gold where $D_1 = 300$ ms, $\alpha_1 = 7\times10^{-3}$ and $\beta_1 = 3.5\times10^{-3}$; Silver where $D_2 = 400$ ms, $\alpha_2 = 5\times10^{-3}$ and $\beta_2 = 2\times10^{-3}$; and Bronze where $D_3 = 500$ ms, $\alpha_3 = 3\times10^{-3}$, and $\beta_3 = 1\times10^{-3}$.

3.2 Forecasting Model

This work builds on top of units of LSTM to investigate long-term dependencies of data. As a variant of Recurrent Neural Networks (RNNs), LSTM adopts a block of memory to replace neurons in the hidden layer. Each memory block includes memory cells and three types of gates, i.e., *input*, *forget* and *output* ones, respectively. They are either closed or opened, to specify whether the memory state in the layer of output leads to a threshold value that is used in the current layer. The model of the memory cells in an SG-LSTM is shown in Fig. 3.

A *forget* gate is adopted to produce the output vectors of the input and the hidden layer in the last time slot. The output value of *forget* gate in current time slot t, f_t, is obtained as:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \tag{1}$$

where σ is a sigmoid activation function, h_{t-1} denotes an output of hidden state in time slot $t-1$, x_t denotes an input vector in time slot t, W_f denotes a weight vector for forget gate f, and b_f denotes a bias vector for forget gate f.

After *input* and *forget* gates, the state of current unit of time memory is delivered to the next time slot, i.e.,

$$a_t = \sigma(W_a[h_{t-1}, x_t] + b_a) \tag{2}$$

$$c_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{3}$$

$$c_t = f_t * c_{t-1} + a_t * c_t, \tag{4}$$

where a_t is the output result of the *input* gate in time slot t, W_a and W_c denote weight vectors for input gate a and cell state c, $tanh$ denotes the tangent function, c_t denotes the cell state in time slot t, c_{t-1} denotes the cell state in time slot $t-1$, and b_a and b_c denote bias vectors for input gate a and cell state c.

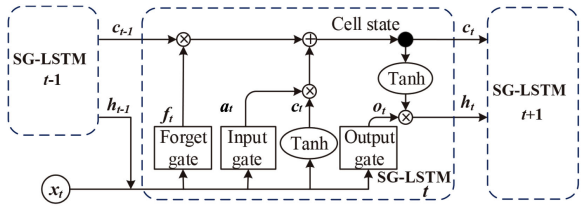


Fig. 3. Structure of memory cells in SG-LSTM.

The third component is the *output* gate. Its input and updated state of memory units are synthetically used to produce the output, i.e.,

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(c_t), \tag{6}$$

where o_t is the value of the *output* gate in time slot t, h_t is the output of hidden state in time slot t, W_o denotes a weight vector and b_o is a vector of bias for output gate o.

3.3 Service Rate

Let J denote the total number of GCDCs. Let I denote the number of task types. $\varpi_t^{i,j}$ is the number of switched-on servers of type i ($1\le i\le I$) in GCDC j ($1\le j\le J$) in time slot t, and $\varpi_t^{i,j}$ is an integer, i.e., $\varpi_t^{i,j}\in N^+$. κ_i is the number of type i tasks that each single server can process per second. $\mu_t^{i,j}$ denotes the task service rate of type i in GCDC j in time slot t. Then, i.e., $\mu_t^{i,j} = \kappa_i\varpi_t^{i,j}$. Υ^j is the total number of servers in GCDC j:

$$\sum_{i=1}^{I} \varpi_t^{i,j} \le \Upsilon^j. \tag{7}$$

3.4 Loss Probability of Tasks

Q_i is the queue capacity of type i tasks. Similar to [5], we adopt an $M/M/1/Q_i/\infty$ queueing system to model servers of type i in each GCDC. $q(\varpi_t^{i,j})$ is the task loss probability of type i in GCDC j in time slot t. If $\varpi_t^{i,j} = 0$, all tasks have to be rejected, i.e., $q(\varpi_t^{i,j}) = 1$. If $\lambda_t^{i,j} = \mu_t^{i,j}$, according to [33], $q(\varpi_t^{i,j}) = \frac{1}{Q_i+1}$. If $\lambda_t^{i,j} < \mu_t^{i,j}$, the queueing system of each GCDC can achieve a steady state, and $q(\varpi_t^{i,j})$ is obtained in (8) following [5]. Thus, $q(\varpi_t^{i,j})$ is obtained as:

$$q(\varpi_t^{i,j}) = \begin{cases} 1, \varpi_t^{i,j} = 0 \\ \frac{1}{Q_i+1}, \lambda_t^{i,j} = \mu_t^{i,j} \ne 0 \\ \frac{1-\frac{\lambda_t^{i,j}}{\kappa_i\varpi_t^{i,j}}}{1-\left(\frac{\lambda_t^{i,j}}{\kappa_i\varpi_t^{i,j}}\right)^{Q_i+1}}\left(\frac{\lambda_t^{i,j}}{\kappa_i\varpi_t^{i,j}}\right)^{Q_i}, \lambda_t^{i,j} < \mu_t^{i,j} \end{cases} \tag{8}$$

According to the $M/M/1/Q_i/\infty$ queueing system, the average response time of each task of type i in GCDC j is $\frac{1}{\kappa_i\varpi_t^{i,j}-\lambda_t^{i,j}}$, and it needs to be less than or equal to D_i, i.e.,

$$\frac{1}{\kappa_i\varpi_t^{i,j}-\lambda_t^{i,j}} \le D_i \tag{9}$$

In addition, $\lambda_t^{i,j}$ is the number of type i tasks scheduled to GCDC j, and λ_t^i is the arriving rate of type i tasks in time slot t. Thus, we have:

$$\lambda_t^i = \sum_{j=1}^{J} \lambda_t^{i,j}. \tag{10}$$

3.5 Solar and Wind Energy Models

3.5.1 Solar Energy Model

$E_t^{j,\odot}$ is GCDC j's current available solar energy in time slot t [34], [35].

$$E_t^{j,\odot} = \varphi^j \psi^j \upsilon_t^j L, \tag{11}$$

where φ_j denotes the converting rate of the solar irradiance to the electricity in GCDC j, ψ_j denotes the area of active solar irradiance in GCDC j, υ_t^j denotes the current solar irradiance in GCDC j in time slot t, and L is length of a time slot.

3.5.2 Wind Energy Model

$E_t^{j,\approx}$ denotes GCDC j's current available wind energy in time slot t [34], [35].

$$E_t^{j,\approx} = \frac{1}{2}\eta^j \xi^j \zeta^j (\nu_t^j)^3 L, \tag{12}$$

where η^j is the wind to electricity converting rate in GCDC j, ξ^j denotes the local air density in GCDC j, ζ^j denotes the wind turbine rotor area, and ν_t^j is GCDC j's wind speed in t.

3.6 Objective Functions

Two objective functions are designed in this work: energy cost modeling and revenue modeling.

3.6.1 Energy Cost

Within each time slot t, the number of type i tasks handled by each switched-on server in GCDC j is $\frac{L(1-q(\varpi_t^{i,j}))\lambda_t^{i,j}}{\varpi_t^{i,j}}$. Thus, the CPU busy time of each switched-on server of type i in GCDC j is $\frac{L(1-q(\varpi_t^{i,j}))\lambda_t^{i,j}}{\kappa_i \varpi_t^{i,j}}$ minutes. The CPU utilization for each switched-on server of type i in GCDC j in time slot t is denoted by $U_t^{i,j}$. By dividing the busy time of CPU by L, $U_t^{i,j}$ is obtained as $\frac{(1-q(\varpi_t^{i,j}))\lambda_t^{i,j}}{\kappa_i \varpi_t^{i,j}}$.

In addition, let $P_t^{i,j}$ denote the total power consumed by type i tasks in GCDC j in time slot t. Following [35], $P_t^{i,j}$ is obtained as:

$$P_t^{i,j} = \varpi_t^{i,j}\left[\check{P} + (\epsilon - 1)\hat{P} + (\hat{P} - \check{P})U_t^{i,j}\right], \tag{13}$$

where $\check{P}$ ($\hat{P}$) is the power consumed by each idle (active) server, and ϵ is the value of power usage effectiveness for each GCDC.

By replacing $U_t^{i,j}$ in (13), $P_t^{i,j}$ is obtained as:

$$P_t^{i,j} = \frac{\mathfrak{\lambda}_1 \kappa_i \varpi_t^{i,j} + \mathfrak{\lambda}_2 \lambda_t^{i,j}(1 - q(\varpi_t^{i,j}))}{\kappa_i} \tag{14}$$

where

$$\mathfrak{\lambda}_1 = \check{P} + (\epsilon - 1)\hat{P} \tag{15}$$

$$\mathfrak{\lambda}_2 = \hat{P} - \check{P} \tag{16}$$

$$[\cdot]^+ = max\{\cdot, 0\}, \tag{17}$$

$\mathbb{C}_t$ is the cost of energy consumed by multiple GCDCs in time slot t. Similar to [15], it is also assumed that the green energy is free if the renewable facilities have been installed. Then, $\mathbb{C}_t$ is obtained as:

$$\mathbb{C}_t = \sum_{j=1}^{J}\left(\Psi_t^j\left[\sum_{i=1}^{I} P_t^{i,j} L - E_t^{j,\odot} - E_t^{j,\approx}\right]^+\right), \tag{18}$$

where Ψ_t^j is the price of electricity in GCDC j in time slot t.

3.6.2 Revenue Modeling

The revenue brought to multiple GCDCs in time slot t [35] is:

$$\mathbb{R}_t = \sum_{j=1}^{J}\sum_{i=1}^{I}((1 - q(\varpi_t^{i,j}))\alpha_i \lambda_t^{i,j} L - q(\varpi_t^{i,j})\beta_i \lambda_t^{i,j} L), \tag{19}$$

where α_i denotes the fee within D_i of each task of type i, and β_i is the penalty corresponding to each task of type i finished beyond D_i.

3.7 Problem Definition

With all the preliminaries prepared, we are ready to formulate the problem into solving a bi-objective optimization problem to evaluate the prediction accuracy of renewable energy. According to the objective functions established in Section 3.6, a problem of bi-objective optimization is formulated to optimize F_1 and F_2, i.e.,

$$F_1 : \mathop{\mathbf{Max}}_{\boldsymbol{d}} \ \mathbb{R}_t(\boldsymbol{d}) \tag{20}$$

$$F_2 : \mathop{\mathbf{Min}}_{\boldsymbol{d}} \ \mathbb{C}_t(\boldsymbol{d}) \tag{21}$$

subject to

$$\sum_{i=1}^{I} \varpi_t^{i,j} \le \Upsilon^j \tag{22}$$

$$\frac{1}{\kappa_i \varpi_t^{i,j} - \lambda_t^{i,j}} \le D_i \tag{23}$$

$$\lambda_t^i = \sum_{j=1}^{J} \lambda_t^{i,j} \tag{24}$$

$$\varpi_t^{i,j} \in N^+, \ \lambda_t^{i,j} \ge 0. \tag{25}$$

It is worth noting that (20) and (21) are conflicting with each other. For example, the minimization of F_2 requires that the fewest servers are switched-on, i.e., $\varpi_t^{i,j} = 0$. However, this means that no tasks are executed, which brings no revenue to GCDCs, thereby failing to maximize F_2.

Here, $\boldsymbol{d}$ is a vector of decision variables, i.e., $\lambda_t^{i,j}$ and $\varpi_t^{i,j}$ (i=1, 2, ..., I, j=1, 2, ..., J). Thus, $\boldsymbol{d}$=[$\lambda_t^{1,1}$, $\lambda_t^{1,2}$, $\cdots$, $\lambda_t^{I,J}$, $\varpi_t^{1,1}$, $\varpi_t^{1,2}$, $\cdots$, $\varpi_t^{I,J}$]. $\boldsymbol{d}$ can be regarded as an encoding operation, which is composed of $\lambda_t^{i,j}$ and $\varpi_t^{i,j}$. Its dimension is $2IJ$. Its first IJ dimensions represent the number of tasks of three different types assigned to three GCDCs; while its last IJ

dimensions represent the number of switched-on servers in GCDCs for processing the tasks.

To evaluate the objective functions in the proposed model, we need to calculate them through $\boldsymbol{d}$. In this work, the transformation from $\boldsymbol{d}$ to objective functions $\mathbb{R}_t$ and $\mathbb{C}_t$ can be regarded as a decoding operation. A method of penalty function is adopted to transform a bi-objective constrained optimization problem into an unconstrained one given as:

$$\widetilde{F}_1 : \min_{\boldsymbol{d}} \{-\mathbb{R}_t(\boldsymbol{d}) + \varepsilon \beth(\boldsymbol{d})\} \tag{26}$$

$$\widetilde{F}_2 : \min_{\boldsymbol{d}} \{\mathbb{C}_t(\boldsymbol{d}) + \varepsilon \beth(\boldsymbol{d})\} \tag{27}$$

$$\beth(\boldsymbol{d}) = \sum_{\rho=1}^{\beth} [max\{0, -\Delta_\rho(\boldsymbol{d})\}]^2 + \sum_{\chi=1}^{\sqsubset} |n_\chi(\boldsymbol{d})|^2, \tag{28}$$

where $n_\chi(\boldsymbol{d})$ denotes an equality constraint χ, and $\Delta_\rho(\boldsymbol{d})$ denotes an inequality constraint ρ. $\beth$ is the total number of all inequality constraints, $\sqsubset$ is the total number of all equality constraints, and ε is a large positive constant.

Therefore, the unconstrained problem represents a constrained mixed integer non-linear programming problem (MINLP), and its solution complexity is NP-hard [37]. It is also demonstrated that MINLP has a problem of exponential explosion, and there are no polynomial-time approaches [38]. Thus, we developed an improved bi-objective metaheuristic optimization algorithm named DMGC to solve it.

4 SG-LSTM PLUS DMGC

In this section, we present our proposed solution, an SG-LSTM network to realize noise filtering and forecast green energy, and DMGC to realize the optimization of the revenue and energy cost of GCDCs.

4.1 SG-LSTM Network

Based on our earlier work [36], we design a network of Savitzky Golay and Long Short Term Memory (SG-LSTM) for predicting solar irradiance and wind speed in GCDCs. Savitzky Golay filter is adopted to remove interference of noise, and extreme points in the data. The network comprises four layers, as shown in Fig. 4: input, hidden, fully connected (FC) and output layers, respectively. The neuron numbers in the input layer and the hidden layer are 30 and 128, respectively. The neuron numbers in the fully connected layer and the output one are 64 and 1, respectively. The neurons in the input layer mean the input values, and all the other neurons are functional ones with activation functions. $x = (x_1, x_2, \cdots, x_\omega)$ is the input data, and ω represents its size.

4.2 Decomposition-Based Multi-Objective Evolutionary Algorithm With Gaussian Mutation and Crowding Distance (DMGC)

Based on our green energy prediction model, we further design an improved bi-objective optimization algorithm to jointly optimize F_1 and F_2 in the formulated problem defined in Section 3.7. The proposed algorithm named DMGC is derived by following a multiobjective evolutionary algorithm based on decomposition (MOEA/D) algorithm. It includes crossover, mutation and selection operations. Different from MOEA/D, our proposed DMGC is novel in the following three aspects. *First*, DMGC adopts dynamic crossover/mutation parameters for tackling the shortcomings of premature convergence and poor stability of MOEA/D. *Second*, DMGC exploits the Gaussian mutation to strengthen the efficiency of global search. It can keep better genes, and introduce high-quality ones into the population when they are trapped into local optima. *Third*, DMGC adopts a crowding degree evaluation approach based on crowding distance. In each iteration of DMGC, individuals with smaller crowding distances are removed from the external population.

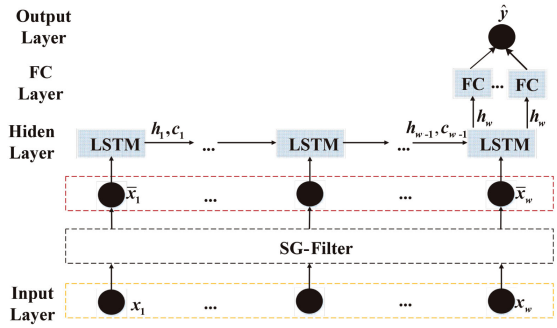


Fig. 4. Structure of SG-LSTM.

4.2.1 Dynamic Crossover/Mutation Parameters

At the initial stage, DMGC adopts larger crossover/mutation parameters, which strengthen the ability of global search, and speed up the convergence speed. In the later stage, crossover/mutation parameters are smaller for preventing excellent genes from being destroyed than those in the initial stage. In addition, new genes can be introduced into the population for avoiding trapping into local optima.

$$p_v = \hat{p}_v - (\hat{p}_v - \check{p}_v) * atan(\eth/\Gamma), \tag{29}$$

$$p_\iota = \hat{p}_\iota - (\hat{p}_\iota - \check{p}_\iota) * atan(\eth/\Gamma) \tag{30}$$

where $\hat{p}_v$ ($\check{p}_v$) is the maximum (minimum) possibility of crossover, $\hat{p}_\iota$ ($\check{p}_\iota$) is the maximum (minimum) possibility of mutation, $atan$ denotes an arctan function, $\eth$ is the iteration count, and Γ is the total number of iterations in DMGC.

4.2.2 Gaussian Mutation

The Gaussian mutation adopts a random vector that conforms to a Gaussian distribution for updating a new individual (X_{new}) produced after the crossover operation. In this way, it helps DMGC to escape from local optima and converge to global ones in a faster manner. κ denotes a vector of variance, which is obtained as:

$$\kappa = \frac{\hat{\vartheta} - \check{\vartheta}}{|\vartheta|}, \tag{31}$$

where $\hat{\vartheta}$ ($\check{\vartheta}$) is a vector of lower (higher) bounds of all decision variables, and $|\vartheta|$ is the dimension of each individual.

$\mathcal{V}$ denotes a vector of random values, which is obtained as:

$$\mathcal{V} = \mathrm{Min}(\mathrm{Max}(\mathcal{N}(X_{new}, \kappa), \check{\vartheta}), \hat{\vartheta}), \tag{32}$$

where $\mathcal{N}(\cdot)$ denotes a Gaussian distribution function.

Then, for each dimension $\lceil$ of X_{new}, if $\mathrm{rand}() < p_\iota$, $X^{\lceil}_{new} = \mathcal{V}^{\lceil}$, i.e.,

$$X^{\lceil}_{new} = \mathcal{V}^{\lceil}, \text{ if } \mathrm{rand}() < p_\iota, \tag{33}$$

where $\mathrm{rand}()$ is a standard function that returns a random number in (0,1) in Matlab.

4.2.3 Crowding Distance

Let $\Lambda(\vartheta)$ denote the crowding distance of each individual ϑ in the external elite solution set Ω. $\Lambda(\vartheta)$ is obtained as follows.

$$\Lambda(\vartheta) = \begin{cases} \infty, \widetilde{F}^{+}_{\varrho} - \widetilde{F}^{-}_{\varrho} = 0 \text{ or } \vartheta = 1 \text{ or } \vartheta = |\Omega| \\ \sum_{\varrho=1}^{\gamma} \frac{(next^{\varrho}_{\vartheta} - pre^{\varrho}_{\vartheta})}{(\widetilde{F}^{+}_{\varrho} - \widetilde{F}^{-}_{\varrho})}, \text{otherwise} \end{cases}, \tag{34}$$

where $|\Omega|$ denotes the number of individuals in Ω, $\widetilde{F}^{+}_{\varrho}$ and $\widetilde{F}^{-}_{\varrho}$ denote maximum and minimum values of objective function ϱ in Ω, respectively. $next^{\varrho}_{\vartheta}$ ($pre^{\varrho}_{\vartheta}$) denotes the function value of the next (previous) individual adjacent to individual ϑ sorted by values of each objective function ϱ.

Algorithm 1 shows the pseudo code of DMGC. Line 1 specifies the population size ($\widetilde{N}$), the number of vectors of weight (T), the set of vectors of weight ($\tilde{\lambda}$) and the maximum number of iterations (Γ). Line 2 updates the Euclidean distance between any two vectors of weight, and changes $B(\vartheta)$ for each solution ϑ. Line 4 initializes Z that includes currently best values for two objectives. Line 5 chooses non-dominated solutions from X, and keeps them into Ω. Line 8 changes the probability of crossover, p_v, and the probability of mutation, p_ι, by using (29) and (30). Line 11 selects two random solutions k and l from $B(\vartheta)$. Line 12 shows that if a random number r is within the range of the generated crossover parameters, two selected individuals are crossed to produce a new solution. Otherwise, a random individual is selected as a new one. Similar to MOEA/D, X_{new} is updated as:

$$X_{new} = \begin{cases} rX(k) + (1-r)X(l), \ r < p_v \\ X(m), \ \text{otherwise} \end{cases}, \tag{35}$$

where r ($r \in (0,1)$ is a random number and $X(m)$ denotes an individual randomly selected from population X.

Line 13 performs Gaussian mutation on individual X_{new} with probability p_ι, and checks the boundary value. Line 16 updates Z. For each $\varrho \in \{1,\ldots,\gamma\}$, if $Z_\varrho < \widetilde{F}_\varrho(X_{new})$, $Z_\varrho = \widetilde{F}_\varrho(X_{new})$. Line 19 updates $B(\vartheta)$ for each individual ϑ. Similar to operations of MOEA/D, for each $\vartheta' \in B(\vartheta)$, if $g^{te}(X_{new}|\tilde{\lambda}^\varrho, Z) \leq g^{te}(X(\vartheta')|\tilde{\lambda}^\varrho, Z)$, $X(\vartheta') = X_{new}$. $g^{te}(X(\vartheta')|\tilde{\lambda}^\varrho, Z)$ is obtained as follows.

$$g^{te}(X(\vartheta')|\tilde{\lambda}^\varrho, Z) = \underset{\varrho \in \{1,\cdots,\gamma\}}{\mathbf{Max}} \{\tilde{\lambda}^\varrho |\widetilde{F}_\varrho(X(\vartheta')) - Z_\varrho|\}, \tag{36}$$

where $X(\vartheta')$ denotes individual ϑ' in $B(\vartheta)$, and γ denotes the number of objective functions.

Line 20 updates Ω according to the crowding distance method. This step removes all the vectors dominated by X_{new} from Ω. If the size of Ω is less than $|\Omega|$, X_{new} is immediately added to Ω if X_{new} is not dominated by any vectors in Ω. If the current size of Ω exceeds $|\Omega|$, the method of crowding degree evaluation based on crowding distance $\Lambda(\vartheta)$ of individual ϑ is performed with (34). An individual with the smallest crowding distance in Ω is replaced. Finally, Line 12 outputs Ω.

Algorithm 1. DMGC

1: Initialize the size of population ($\widetilde{N}$), the number of vectors of weight (T), the set of weight vectors ($\tilde{\lambda}$), and the total number of iterations (Γ)
2: Update Euclidean distances between any two vectors of weight, and change $B(\vartheta)$ for each solution ϑ
3: Initialize X and Ω
4: Initialize Z that includes currently best values for two objectives
5: Keep non-dominated solutions in X into Ω
6: $\eth \leftarrow 1$
7: **while** $\eth \leq \Gamma$ **do**
8: Change the probability of crossover, p_v, and the probability of mutation, p_ι, by using (29) and (30)
9: $\vartheta \leftarrow 1$
10: **while** $\vartheta \leq \widetilde{N}$ **do**
11: Choose two random solutions k and l from $\overline{B}(\vartheta)$
12: Update X_{new} with (35)
13: Perform Gaussian mutation on X_{new} with the probability of p_ι, and check the boundary value
14: **for** $\varrho \leftarrow 1$ to γ **do**
15: **if** $Z_\varrho < \widetilde{F}_\varrho(X_{new})$ **then**
16: $Z_\varrho = \widetilde{F}_\varrho(X_{new})$
17: **end if**
18: **end for**
19: Update $B(\vartheta)$ for each solution ϑ
20: Update Ω with (34)
21: **end while**
22: **end while**
23: **return** Ω

We further analyze the computing complexity of Algorithm 1. In Algorithm 1, the computing overhead is mainly caused by the *while* loop, which terminates after Γ iterations. In each iteration of DMGC, the complexity of Lines 14–18 is $\mathcal{O}(\gamma)$. The complexity of Line 19 is $\mathcal{O}(\gamma^2)$. Line 20 includes the sorting of individuals based on the crowding distance, and the update of Ω. The time complexity of the former is $\mathcal{O}(|\Omega|\log|\Omega|)$ while that of the latter is $\mathcal{O}(|\Omega|)$. Thus, the time complexity of Line 20 is $\mathcal{O}(|\Omega|\log|\Omega|)$. In other words, the accumulated complexity of each iteration is $\mathcal{O}(\widetilde{N}(\gamma^2 + |\Omega|\log|\Omega|))$. Finally, the complexity of Algorithm 1 becomes $\mathcal{O}(\Gamma\widetilde{N}(\gamma^2 + |\Omega|\log|\Omega|))$.

5 Performance Evaluation

We adopted real-life data to evaluate the performance of our proposed algorithms. DMGC was coded with MATLAB r2017b, and executed on a server with an Intel (R) core (TM) i7-6700hq CPU at 2.59 GHz and 8GB memory. Three types of tasks were collected from the trace of Google cluster.[1] Each type of task was split into 96 time slots, each of which lasting for 15 minutes. In addition, three GCDCs were considered, and their prices of electricity,[2] wind speeds and solar irradiance[3] on the same day were collected.

1. https://github.com/google/cluster-data
2. https://www.nyiso.com/
3. https://midcdmz.nrel.gov/

TABLE 2
MAEs and RMSEs With Varying Sizes of Window and Three Methods of Smoothing

Smoothing methods	RMSE					MAE				
	$\omega=3$	$\omega=5$	$\omega=7$	$\omega=9$	$\omega=11$	$\omega=3$	$\omega=5$	$\omega=7$	$\omega=9$	$\omega=11$
Average filter (υ_t^1)	99.82	111.69	131.93	152.75	174.20	48.32	64.08	82.12	99.93	117.59
Median filter (υ_t^1)	76.89	76.87	78.86	80.92	84.00	33.98	33.03	33.27	34.27	36.04
SG filter (υ_t^1)	99.79	**47.73**	51.62	54.22	56.22	45.71	**21.24**	24.46	24.17	23.76
Average filter (υ_t^2)	97.69	109.14	129.12	150.49	170.67	47.57	62.80	80.17	98.08	114.62
Median filter (υ_t^2)	73.43	73.48	79.43	75.37	80.51	32.05	31.03	32.52	31.10	33.16
SG filter (υ_t^2)	106.54	**47.26**	52.34	52.01	55.47	47.01	**21.62**	23.46	22.97	23.69
Average filter (υ_t^3)	20.33	24.88	29.95	33.99	37.38	11.15	14.07	17.81	20.99	23.63
Median filter (υ_t^3)	19.02	17.23	17.16	17.10	17.59	10.12	9.26	9.51	9.49	9.44
SG filter (υ_t^3)	21.11	**9.30**	13.23	9.99	10.21	11.85	**5.15**	7.08	5.59	5.39
Average filter (ν_t^1)	0.75	0.83	0.90	0.96	1.02	0.54	0.61	0.68	0.73	0.78
Median filter (ν_t^1)	0.55	0.54	0.58	0.63	0.68	0.38	0.36	0.39	0.43	0.48
SG filter (ν_t^1)	0.75	**0.29**	0.38	0.39	0.42	0.55	**0.22**	0.28	0.29	0.31
Average filter (ν_t^2)	0.96	1.07	1.16	1.23	1.31	0.66	0.76	0.83	0.88	0.94
Median filter (ν_t^2)	0.77	0.72	0.73	0.81	0.87	0.48	0.46	0.49	0.54	0.58
SG filter (ν_t^2)	1.02	**0.40**	0.52	0.51	0.54	0.72	**0.28**	0.36	0.37	0.38
Average filter (ν_t^3)	1.15	1.29	1.41	1.51	1.60	0.83	0.96	1.05	1.13	1.20
Median filter (ν_t^3)	0.89	0.85	0.89	0.99	1.04	0.59	0.56	0.61	0.67	0.71
SG filter (ν_t^3)	1.14	**0.49**	0.66	0.62	0.66	0.82	**0.36**	0.49	0.45	0.48

5.1 Prediction of Solar and Wind Energy

We compared our proposed SG-LSTM against GRU, BPNN and Bi-LSTM on their performance of predicting wind speeds and solar irradiance. The parameters of SG-LSTM, GRU, BPNN and Bi-LSTM are listed as follows. The data sample count is 12,480; the input data size is 30; the percentage of the training set is 0.9; and the input feature and the output result are both one-dimensional. The neuron count in the hidden layer in SG-LSTM, GRU, BPNN and Bi-LSTM is 128, and that in their fully connected layer is 64.

Different sizes of window were considered and three types of methods of data smoothing, i.e., SG filter [39], median filter [40], and average filter [41] were adopted in SG-LSTM. The best one with the smallest error was selected to determine the optimal size of window. In this experiment, MAE and RMSE were used to measure the accuracy of prediction[42], i.e.,

$$MAE = \frac{1}{N}\sum_{\hbar=1}^{N}|y_\hbar - \widehat{y}_\hbar| \tag{37}$$

$$RMSE = \sqrt{\frac{\sum_{\hbar=1}^{N}(y_\hbar - \widehat{y}_\hbar)^2}{N}}, \tag{38}$$

where N is the data sample number, $y_\hbar$ and $\widehat{y}_\hbar$ are the original and the predicted data.

The experimental results are summarized in Table 2. As shown in Table 2, when window size is set to 5, the least mean absolute errors (MAEs) and root mean square errors (RMSEs) are achieved. The SG filter's window size is 5 and the order is 3.

RMSEs and MAEs of the four methods in a day are shown in Table 3. In other words, Table 3 shows the prediction errors of different algorithms for different energy. It can be seen from the results of RMSE that, the RMSE of SG-LSTM is significantly better than that of GRU, BPNN and Bi-LSTM methods in υ_t^1, υ_t^2 and υ_t^3, and slightly better than that of GRU, BPNN and Bi-LSTM methods in ν_t^1, ν_t^2 and ν_t^3. The results of MAE prove that SG-LSTM's MAE is much better than GRU, BPNN and Bi-LSTM methods in υ_t^1, υ_t^2 and υ_t^3, and slightly better than GRU, BPNN and Bi-LSTM methods in ν_t^1, ν_t^2 and ν_t^3. It can be concluded that SG-LSTM outperforms its three peers in terms of prediction accuracy. The reason may be that SG-LSTM adopts an SG filter to remove the interference of outliers and noises in the large-scale time series data of wind speeds and solar irradiance. Thus, highly fluctuating and non-linear problems of the data are well addressed.

TABLE 3
RMSEs and MAEs of SG-LSTM, GRU, BPNN and Bi-LSTM

Criteria	Methods	υ_t^1	υ_t^2	υ_t^3	ν_t^1	ν_t^2	ν_t^3
RMSE	SG-LSTM	**23.24**	**21.88**	**3.48**	**0.31**	**0.49**	**0.50**
	GRU	29.75	24.74	3.64	0.32	0.55	0.54
	BPNN	30.50	31.94	4.78	0.35	0.59	0.62
	Bi-LSTM	25.40	21.10	3.10	0.33	0.53	0.52
MAE	SG-LSTM	**11.03**	**10.75**	**2.39**	**0.22**	**0.36**	**0.38**
	GRU	14.09	11.89	2.45	0.24	0.40	0.40
	BPNN	16.04	17.91	3.37	0.25	0.41	0.48
	Bi-LSTM	11.80	10.10	2.43	0.24	0.38	0.41

5.2 Results of Bi-Objective Task Scheduling

To demonstrate the effect of the energy prediction accuracy on the bi-objective task scheduling, tasks of three types were scheduled to three GCDCs by using DMGC. The parameters of energy sources were set in Table 4; while κ_i and Q_i were given in Table 5. The number of servers in three GCDCs was all set to 2,000, i.e., $\Upsilon^1=\Upsilon^2=\Upsilon^3=2000$. The power consumption of each idle server was 500 (W), i.e., $\check{P}=500$ (W), the power consumption of each active server was 2,000 (W), i.e., $\hat{P}=2000$ (W), the power usage effectiveness value was 1.2, i.e., $\epsilon=1.2$.

To evaluate the performance of DMGC, we selected the Multi-Objective Evolutionary Algorithm based on

TABLE 4
Setting of Parameters of Wind and Solar Energy

$GCDC_j$	Wind energy			Solar energy	
	η^j	$\varsigma^j(m^2)$	$\xi^j(kg/m^3)$	$\varphi^j(m^2)$	ψ^j
j=1	0.3	2000	1.0	0.2	1400
j=2	0.35	2500	1.25	0.25	1600
j=3	0.4	3000	1.5	0.3	1800

TABLE 5
Setting of Parameters of Three GCDCs

$GCDC_j$	κ_i (tasks/sec.)			B_i		
	i=1	i=2	i=3	i=1	i=2	i=3
j=1	0.01	0.03	0.05	50	55	60
j=2	0.01	0.03	0.05	50	55	60
j=3	0.01	0.03	0.05	50	55	60

Decomposition (MOEA/D) [18], the Strength Pareto Evolutionary Algorithm II (SPEA2) [43], and the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [44] for performance comparison. The reasons why we selected them as state-of-the-art algorithms for comparison are three-fold. *First*, over the years, those algorithms have been improved in efficiency and accuracy, and they have been widely adopted in many different real-world areas. *Second*, there are some similarities between DMGC and those three algorithms. They all have operations of initialization, mutation and crossover to obtain new solutions. *Third*, they all have external elite solution sets. In our experiments, the main parameters of all algorithms under study were set as follows: the population size $\widetilde{N} = 100$, and the total number of iterations $\Gamma = 500$.

Figs. 5 and 6 illustrate task arriving rates of three types and prices of electricity in different GCDCs. Table 6 shows the comparison of revenue, profit and cost with different algorithms. Specifically, the revenue with SG-LSTM, GRU, Bi-LSTM, and BPNN-based predicted data reduces by

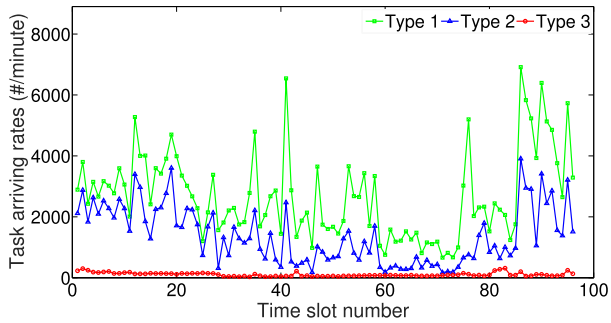


Fig. 5. Arriving rates of tasks of three applications.

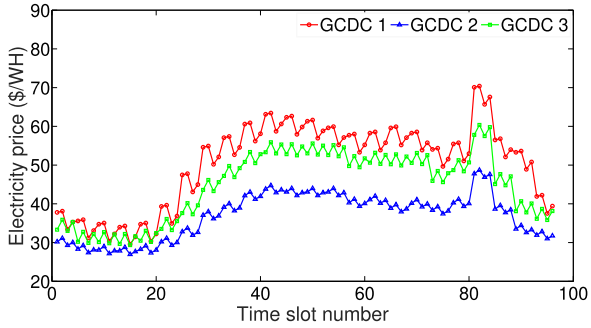


Fig. 6. Prices of electricity of three GCDCs.

0.0001%, 0.0038%, 0.0020%, and 0.0015% over that with the ideal case, respectively. The cost with SG-LSTM, GRU, Bi-LSTM, and BPNN-based predicted data increases by 2.04%, 4.50%, 4.13%, and 5.74% over that with the ideal case, respectively. The profit with SG-LSTM, GRU, Bi-LSTM, and BPNN-based predicted data reduces by 0.24%, 0.55%, 0.243%, and 0.71% over that with the ideal case, respectively.

Moreover, in Table 6, the optimization results with help of the SG-LSTM prediction algorithm are the closest to the ideal case', and the profit of DMGC is 1.20% and 1.93% higher than those of NSGA-II and SPEA2, respectively. The average execution time of DMGC is 59.36% and 36.27% lower than those of SPEA2 and NSGA-II, respectively. This means that DMGC achieves better performance with respect

TABLE 6
Optimization Results of Algorithms

Algorithms	Predicted data	Revenue ($)	Energy cost ($)	Profit ($)	Average execution time (Sec.)
DMGC	Ideal Case	26319.04	2743.95	23575.09	0.3556
	SG-LSTM	**26319.01**	**2801.19**	**23517.82**	**0.3560**
	GRU	26318.05	2873.35	23444.70	0.3603
	BPNN	26318.64	2910.92	23407.72	0.3986
	Bi-LSTM	26318.50	2862.25	23456.25	0.3590
NSGA-II	Ideal Case	26317.15	3035.64	23281.51	0.8747
	SG-LSTM	**26316.81**	**3081.63**	**23235.18**	**0.8760**
	GRU	26314.84	3091.16	23223.68	0.8870
	BPNN	26314.49	3124.13	23190.36	0.8987
	Bi-LSTM	26315.60	3086.45	23229.15	0.8812
SPEA2	Ideal Case	26316.01	3218.51	23097.50	0.5570
	SG-LSTM	**26315.74**	**3244.73**	**23071.01**	**0.5586**
	GRU	26314.73	3319.66	22995.07	0.5720
	BPNN	26314.57	3397.13	22917.44	0.5770
	Bi-LSTM	26315.10	3280.70	23034.40	0.5685

TABLE 7
Comparison of NSGA-II, SPEA2, MOEA/D and DMGC

Algorithms	Revenue ($)	Energy cost ($)	Execution time (Sec.)
NSGA-II	4.3250×10^4	3.2266×10^3	0.8650
SPEA2	4.2230×10^4	3.3370×10^3	0.5620
MOEA/D	4.5893×10^4	3.1981×10^3	0.4560
DMGC	$\mathbf{5.2750}\times10^4$	$\mathbf{2.7810}\times10^3$	**0.3458**

to the profit and execution time. The experimental data also shows that green energy forecast accuracy has significant influence on the task scheduling in GCDCs. The reason is that our proposed method solves the challenge of how to combine the green energy prediction and bi-objective optimization for high-performance GCDC operation. In addition, the improved DMGC integrates Gaussian mutation and crowding distance to achieve a good trade-off between energy cost and revenue of GCDCs.

Table 7 further shows the comparison of NSGA-II, SPEA2, MOEA/D and DMGC in terms of revenue, energy cost and execution time by using predicted data with SG-LSTM. In Table 7, we double α_i and β_i in Fig. 2. Table 7 shows that compared with NSGA-II, SPEA2, and MOEA/D, DMGC increases the revenue by 18%, 20% and 13.1%, reduces the energy cost by 16%, 19.8% and 15.2%, and lowers execution time by 60.02%, 38.47% and 24.17%, respectively. The reason is that DMGC adopts dynamic crossover/mutation parameters and the Gaussian mutation to escape from local optima. In addition, DMGC adopts the crowding distance to make the distribution of solutions more uniform, thereby achieving a good trade-off between revenue and energy cost of GCDCs.

6 Conclusion

Increasingly more enterprises migrate their business applications in cloud data centers, whose operations cause increasingly high energy consumption and large amount of carbon emission. To well address such concerns, renewable energy is increasingly used to create green cloud data centers (GCDCs). Nevertheless, it is often challenging to realize high-accuracy prediction of wind and solar energy, which plays a crucial role in optimal task scheduling. This work designs an intelligent task scheduling approach that integrates a hybrid green energy prediction method with an improved intelligent optimization algorithm. The prediction is realized by designing Savitzky-Golay filter and Long Short-Term Memory (SG-LSTM). The bi-objective optimization of the revenue and energy cost of GCDCs is achieved by proposing Decomposition-based Multi-objective evolutionary algorithm with Gaussian mutation and Crowding distance (DMGC). This work has verified the importance of renewable energy prediction accuracy in achieving a desired intelligent schedule. Experimental results over real-world data have proven that SG-LSTM achieves higher prediction accuracy for wind speed and solar irradiance, and increases the bi-objective scheduling of tasks in GCDCs. The experiments have demonstrated the importance of high-accuracy prediction of green energy on scheduling of tasks of different types in GCDCs, which was not reported in previous studies to our best knowledge.

In this work, we adopt a centralized task scheduler to schedule tasks to multiple GCDCs, and it might suffer from a single point of failure/bottleneck and some communication delays. Our future work aims to design federated task schedulers to perform distributed optimization and handle a great number of tasks in parallel. Besides, we also plan to consider capacity management in the combination of workload prediction and the task scheduling.

References

[1] Y. Mansouri, A. N. Toosi, and R. Buyya, "Cost optimization for dynamic replication and migration of data in cloud data centers," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 705–718, Sep. 2019.

[2] J. Bi *et al.*, "Application-aware dynamic fine-grained resource provisioning in a virtualized cloud data center," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1172–1184, Apr. 2017.

[3] H. Yuan, J. Bi, W. Tan, M. Zhou, B. H. Li, and J. Li, "TTSA: An effective scheduling approach for delay bounded tasks in hybrid clouds," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3658–3668, Nov. 2017.

[4] Z. Hu, B. Li, and J. Luo, "Time- and Cost- Efficient task scheduling across geo-distributed data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 3, pp. 705–718, Mar. 2018.

[5] H. Yuan, J. Bi, and M. Zhou, "Spatiotemporal task scheduling for heterogeneous delay-tolerant applications in distributed green data centers," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1686–1697, Oct. 2019.

[6] H. Yuan, J. Bi, W. Tan, and B. H. Li, "Temporal task scheduling with constrained service delay for profit maximization in hybrid clouds," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 337–348, Jan. 2017.

[7] Y. Lu, J. Panneerselvam, L. Liu, and Y. Wu, "RVLBPNN: A workload forecasting model for smart cloud computing," *Sci. Program.*, vol. 2016, pp. 1–9, Nov. 2016.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[9] A. G. Abo-Khalil and D. Lee, "MPPT control of wind generation systems based on estimated wind speed using SVR," *IEEE Trans. Ind. Electronics*, vol. 55, no. 3, pp. 1489–1490, Mar. 2008.

[10] N. K. Anamika and A. K. Akella, "Prediction and efficiency evaluation of solar energy resources by using mixed ANN and DEA approaches," in *Proc. IEEE PES General Meeting Conf. Expo.*, 2014, pp. 1–5.

[11] Y. Jiang, L. Liu, Y. Cui, and X. Su, "Ultra-short-term multistep prediction of wind power based on representative unit method," *Grid Technol.*, vol. 8, pp. 112–118, Oct. 2018.

[12] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[13] D. Dong, Z. Sheng, and T. Yang, "Wind power prediction based on recurrent neural network with long short-term memory units," in *Proc. Int. Conf. Renewable Energy Power Eng.*, 2018, pp. 34–38.

[14] M. Afrasiabi, M. Mohammadi, M. Rastegar, and S. Afrasiabi, "Advanced deep learning approach for probabilistic wind speed forecasting," *IEEE Trans. Ind. Inform.*, vol. 17, no. 1, pp. 720–727, Jan. 2021.

[15] M. Ghamkhari and H. Mohsenian-Rad, "Energy and performance management of green data centers: A profit maximization approach," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 1017–1025, Jun. 2013.

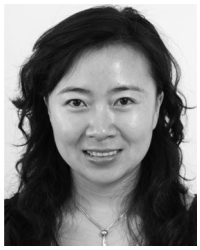
[16] H. Shao, L. Rao, Z. Wang, X. Liu, Z. Wang, and K. Ren, "Optimal load balancing and energy cost management for internet data centers in deregulated electricity markets," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 10, pp. 2659–2669, Oct. 2014.

[17] Y. Wang *et al.*, "Multi-objective workflow scheduling with deep-Q-network-based multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 39974–39982, Apr. 2019.

[18] H. Liu, J. Bi, H. Yuan, and M. Zhou, "Bi-objective intelligent task scheduling for green clouds with deep learning-based prediction," in *Proc. IEEE 17th Int. Conf. Netw. Sens. Control*, 2020, pp. 1–6.

[19] H. Kanemitsu, M. Hanada, and H. Nakazato, "Clustering-based task scheduling in a large number of heterogeneous processors," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 11, pp. 3144–3157, Nov. 2016.

[20] P. Varshney and Y. Simmhan, "AutoBoT: Resilient and cost-effective scheduling of a bag of tasks on spot VMs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 7, pp. 1512–1527, Jul. 2019.
[21] L. Mashayekhy, M. M. Nejad, D. Grosu, Q. Zhang, and W. Shi, "Energy-aware scheduling of mapreduce jobs for big data applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 10, pp. 2720–2733, Oct. 2015.
[22] P. Zhang and M. Zhou, "Dynamic cloud task scheduling based on a two-stage strategy," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 772–783, Apr. 2018.
[23] G. Skourletopoulos *et al.*, "Elasticity debt analytics exploitation for green mobile cloud computing: An equilibrium model," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 1, pp. 122–131, Mar. 2019.
[24] Y. Nan *et al.*, "Adaptive energy-aware computation offloading for cloud of things systems," *IEEE Access*, vol. 5, pp. 23947–23957, Oct. 2017.
[25] S. Pan and Y. Chen, "Energy-optimal scheduling of mobile cloud computing based on a modified lyapunov optimization method," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 1, pp. 227–235, Mar. 2019.
[26] L. Chou, H. Chen, F. Tseng, H. Chao, and Y. Chang, "DPRA: Dynamic power-saving resource allocation for cloud data center using particle swarm optimization," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1554–1565, Jun. 2018.
[27] M. T. Higuera-Toledano, J. L. Risco-Martín, P. Arroba and J. L. Ayala, "Green adaptation of real-time web services for industrial CPS within a cloud environment," *IEEE Trans. Ind. Inform.*, vol. 13, no. 3, pp. 1249–1256, Jun. 2017.
[28] B. Park and J. Hur, "Spatial prediction of renewable energy resources for reinforcing and expanding power grids," *Energy*, vol. 164, pp. 757–772, Sep. 2018.
[29] L. Li, S. Wen, M. Tseng and C. Wang, "Renewable energy prediction: A novel short-term prediction model of photovoltaic output power," *J. Cleaner Prod.*, vol. 228, pp. 359–375, Aug. 2019.
[30] C. D. Li, L. Wang, G. Q. Zhang, H. D. Wang and F. Shang, "Functional-type single-input-rule-modules connected neural fuzzy system for wind speed prediction," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 4, pp. 751–762, Oct. 2017.
[31] Y. Zhang, H. Sun, and Y. Guo, "Wind power prediction based on PSO-SVR and grey combination model," *IEEE Access*, vol. 7, pp. 136 254–136 267, Sep. 2019.
[32] N. Huang, Y. Wu, G. Lu, W. Wang, and X. Cao, "Combined probability prediction of wind power considering the conflict of evaluation indicators," *IEEE Access*, vol. 7, pp. 174 709–174 724, Nov. 2019.
[33] H. Yuan, J. Bi and M. Zhou, "Spatial task scheduling for cost minimization in distributed green cloud data centers," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 729–740, Apr. 2019.
[34] M. Hsieh, F. Hsu, and D. G. Dorrell, "Winding changeover permanent-magnet generators for renewable energy applications," *IEEE Trans. Magn.*, vol. 48, no. 11, pp. 4168–4171, Nov. 2012.
[35] H. Yuan, J. Bi, and M. Zhou, "Temporal task scheduling of multiple delay-constrained applications in green hybrid cloud," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 729–740, Apr. 2019.
[36] H. Yuan, J. Bi, M. Zhou, J. Zhang and W. Zhang, "Improved LSTM-based prediction method for highly variable workload and resources in clouds," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2020, pp. 1206–1211.
[37] F. Boukouvala, R. Misener, and C. A. Floudas, "Global optimization advances in mixed-integer nonlinear programming, MINLP, and constrained derivative-free optimization, CDFO," *Eur. J. Oper. Res.*, vol. 252, no. 3, pp. 701–727, Aug. 2016.
[38] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, "Mixed-integer nonlinear optimization," *Acta Numer.*, vol. 22, pp. 1–131, May 2013.
[39] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chem.*, vol. 36, no. 8, pp. 1627–1639, Jul. 1964.
[40] P. K. Sinha and Q. H. Hong, "An improved median filter," *IEEE Trans. Medical Imaging*, vol. 9, no. 3, pp. 345–346, Sep. 1990.
[41] S. Golestan, M. Ramezani, J. M. Guerrero, F. D. Freijedo and M. Monfared, "Moving average filter based phase-locked loops: Performance analysis and design guidelines," *IEEE Transactions on Power Electronics*, vol. 29, no. 6, pp. 2750–2763, Jun. 2014.
[42] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79–82, Dec. 2005.
[43] A. M. Adham, N. Mohd-Ghazali, and R. Ahmad, "Performance optimization of a microchannel heat sink using the improved strength pareto evolutionary algorithm (SPEA2)," *J. Eng. Thermophysics*, vol. 1, no. 24, pp. 86–100, Jan. 2015.
[44] S. H. R. Pasandideh, S. T. A. Niaki, and K. Asadi, "Bi-objective optimization of a multi-product multi-period three-echelon supply chain problem under uncertain environments: NSGA-II and NRGA," *Inf. Sci.*, vol. 292, pp. 57–74, Jan. 2015.



Jing Bi (Senior Member, IEEE) received the BS and PhD degrees in computer science from Northeastern University, Shenyang, China. She was a postdoctoral researcher with the Department of Automation, Tsinghua University, Beijing, China. She was a visiting research scholar with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ. She is currently an associate professor with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. Her research interests include cloud computing, large-scale data analysis, machine learning and performance optimization. She received the IBM Fellowship Award and the Best Paper Award in the 17th ICNSC.

Haitao Yuan (Senior Member, IEEE) received the PhD degree in modeling simulation theory and technology from Beihang University, Beijing, China, in 2016, and the PhD degree in computer engineering from the New Jersey Institute of Technology (NJIT), Newark, NJ, in 2020. He is currently an associate professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. His research interests include cloud computing, edge computing, data centers, big data and deep learning. He received the Chinese Government Award for Outstanding Self-financed Students Abroad and the Best Paper Award in the 17th ICNSC.

Jia Zhang (Senior Member, IEEE) received the PhD degree in computer science from the University of Illinois at Chicago. She is currently the Cruse C. and Marjorie F. Calahan Centennial Chair in engineering, Professor of Department of Computer Science, Lyle School of Engineering, Southern Methodist University. Her research interests emphasize the application of machine learning and information retrieval methods to tackle data science infrastructure and software engineering problems. She has published more than 180 refereed journal papers, book chapters, and conference papers.

MengChu Zhou (Fellow, IEEE) received the PhD degree from Rensselaer Polytechnic Institute, Troy, NY, in 1990, and then joined New Jersey Institute of Technology where he is currently a distinguished professor. His interests include petri nets, automation, Internet of Things, and big data. He has more than 900 publications including 12 books, more than 600 journal papers (more than 450 in IEEE Transactions), 28 patents and 29 book-chapters. He is Fellow of IFAC, AAAS, CAA and NAI.