Biobjective Task Scheduling for Distributed Green Data Centers

Haitao Yuan, *Member, IEEE*, Jing Bi, *Senior Member, IEEE*, MengChu Zhou, *Fellow, IEEE*, Qing Liu, and Ahmed Chiheb Ammari, *Senior Member, IEEE*

***Abstract*—The industry of data centers is the fifth largest energy consumer in the world. Distributed green data centers (DGDCs) consume 300 billion kWh per year to provide different types of heterogeneous services to global users. Users around the world bring revenue to DGDC providers according to actual quality of service (QoS) of their tasks. Their tasks are delivered to DGDCs through multiple Internet service providers (ISPs) with different bandwidth capacities and unit bandwidth price. In addition, prices of power grid, wind, and solar energy in different GDCs vary with their geographical locations. Therefore, it is highly challenging to schedule tasks among DGDCs in a high-profit and high-QoS way. This work designs a multiobjective optimization method for DGDCs to maximize the profit of DGDC providers and minimize the average task loss possibility of all applications by jointly determining the split of tasks among multiple ISPs and task service rates of each GDC. A problem is formulated and solved with a simulated-annealing-based biobjective differential evolution (SBDE) algorithm to obtain an approximate Pareto-optimal set. The method of minimum Manhattan distance is adopted to select a knee solution that specifies the Pareto-optimal task service rates and task split among ISPs for DGDCs in each time slot. Real-life data-based experiments demonstrate that the proposed method achieves lower task loss of all applications and larger profit than several existing scheduling algorithms.**

***Note to Practitioners*—This work aims to maximize the profit and minimize the task loss for DGDCs powered by renewable energy and smart grid by jointly determining the split of tasks among multiple ISPs. Existing task scheduling algorithms fail to jointly consider and optimize the profit of DGDC providers and QoS of tasks. Therefore, they fail to intelligently schedule tasks of heterogeneous applications and allocate infrastructure resources within their response time bounds. In this work, a new method that tackles drawbacks of existing algorithms is proposed. It is achieved by adopting the proposed SBDE algorithm that solves a multiobjective optimization problem. Simulation experiments demonstrate that compared with three typical task scheduling approaches, it increases profit and decreases task loss. It can be readily and easily integrated and implemented in real-life industrial DGDCs. The future work needs to investigate the real-time green energy prediction with historical data and further combine prediction and task scheduling together to achieve greener and even net-zero-energy data centers.**

***Index Terms*—Cloud data centers, green computing, multiobjective differential evolution (DE), quality of service (QoS), simulated annealing (SA), task scheduling.**

I. Introduction

DISTRIBUTED data centers usually consist of millions of high-performance servers, switches, and storage devices that provide services to global users. The dramatic growth in the number of arriving tasks significantly brings the energy cost of billions of dollars to their providers [1], [2]. In addition, the amount of their energy consumption also causes significant carbon emissions. Therefore, a growing number of enterprises install green energy infrastructure and develop and adopt various methods to make geographically distributed data centers green, resulting in a concept of distributed green data centers (DGDCs) [3], [9]. For example, disk management, dynamic voltage and frequency scaling (DVFS) and energy-efficient task scheduling [5] have been proposed to realize DGDCs.

Yet it remains an open problem for DGDC providers to not maximize their profit and strictly guarantee quality of service (QoS) requirements of tasks. For clarity, tasks considered in this work mean requests of typical delay-sensitive applications deployed in DGDCs. Delay-sensitive applications include web search, interactive online games, electronic business, and other applications requiring very short response time. According to the service-level agreements signed between users and providers, users' tasks bring revenue to DGDC providers. Therefore, the reduction in energy consumption might deteriorate QoS of tasks of users. In this case, users would pay less or zero revenue to DGDC providers, and this inevitably decreases their profit. Consequently, the profit of providers and QoS of tasks need to be jointly optimized by intelligently scheduling tasks and allocating infrastructure resources.

Manuscript received June 23, 2019; revised September 14, 2019; accepted December 7, 2019. This article was recommended for publication by Associate Editor A. Matta and Editor Y. Tang upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61802015 and Grant 61703011, in part by the Major Science and Technology Program for Water Pollution Control and Treatment of China under Grant 2018ZX07111005, in part by the National Defense Pre-Research Foundation of China under Grant 41401020401 and Grant 41401050102, and in part by Sultan Qaboos University through the Omantel Research Program under Grant EG/SQU-OT/19/04. *(Corresponding author: Jing Bi.)*

H. Yuan, M. Zhou, and Q. Liu are with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: haitao.yuan@njit.edu; zhou@njit.edu; qing.liu@njit.edu).

J. Bi is with the School of Software Engineering, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: bijing@bjut.edu.cn).

A. C. Ammari is with the Department of Electrical and Computer Engineering, College of Engineering, Sultan Qaboos University, Muscat 123, Oman (e-mail: ac.ammari@yahoo.fr).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2019.2958979

Users' tasks are delivered to DGDCs through multiple Internet service providers (ISPs). The bandwidth capacity and unit bandwidth price of each ISP are different from each other. Typically, each application is deployed in one of DGDCs located in multiple geographical locations. In each time slot, the prices of power grid in different GDCs vary with their locations. In addition, many factors about green energy, e.g., air density, wind turbine rotor area, wind speed, solar irradiance, and active irradiation area of solar panels, also vary with their locations. Thus, the wind and solar energy also show spatial diversity. In addition, this work considers multiple applications whose QoS constraints are heterogeneous. Besides, this work adopts an $M/M/1/\beta_n^c/\infty$ queueing system [6]–[8] to model the response time of servers of each application in each GDC. It considers the maximum number of available servers and the maximum amount of available energy in each GDC.

There are many metrics (e.g., scalability, robustness, availability, and reliability) that can be used as measures for QoS of each GDC. Among them, the average task loss possibility (ATLP) is critically important because it directly affects the profit of DGDC providers brought by all tasks. In addition, ATLP shows an explicit relation between the task arriving rates and the task service rates of servers in DGDCs. Based on this relation, servers in DGDCs can be dynamically and accurately configured to realize fine-grained resource provisioning. In this work, the ATLP of all applications in DGDCs is chosen as a measure for QoS. This work aims to maximize the profit of a DGDC provider and minimize the ATLP of all applications by properly splitting users' tasks among multiple ISPs, adjusting task service rates of each GDC. It formulates a multiobjective optimization problem. To solve it, this work proposes a simulated-annealing-based biobjective differential evolution (SBDE) algorithm to obtain an approximate Pareto-optimal set. Then, the minimal Manhattan distance is adopted to determine a knee solution that specifies Pareto-optimal task service rates and task split among ISPs for DGDCs in each time slot. In this way, a profit and QoS-optimized task scheduling (PQTS) method is proposed to trade-off the profit of the DGDC provider against the ATLP of all applications. The tradeoff is usually difficult to realize with existing single-objective methods. This work adopts real-life data, e.g., solar irradiance, prices of power grid, wind speed, and task arriving rates in Google cluster, to evaluate the performance of PQTS. Our experiments demonstrate that PQTS achieves lower ATLP of all applications and larger profit than a number of task scheduling algorithms.

The remainder of this article is shown as follows. Section II discusses the related work and shows the novelty of our work. Section III formulates a multiobjective optimization problem for a DGDC provider. Section IV gives the details of the proposed SBDE algorithm. Section V evaluates PQTS by using real-life data. Section VI concludes this article.

II. Related Work

A. Distributed Green Data Centers

An increasing number of studies have been proposed to apply green energy to GDCs [9]–[13]. Kiani and Ansari [10] aim to maximize the profit of DGDCs, which provide multiple types of services to global users. An optimization framework is formulated based on a G/D/1 queue. They divide the workload into brown and green workloads executed by brown and green energy resources, respectively. They consider service-level agreements between users and data centers and different electricity markets in multiple areas. Lei *et al.* [11] formulate a multiobjective problem for energy-efficient task scheduling in a GDC. The computing nodes in this data center are DVFS-enabled and partially powered by green energy. To solve this problem, an improved multiobjective coevolutionary algorithm is proposed, and it applies the preference-inspired co-evolutionary algorithm with goal (PICEA-g) algorithm with the generalized opposition-based learning to determine the optimal supply voltage and clock frequency for task execution. Cai *et al.* [12] propose a green energy-driven method that efficiently enables a data center to increase its computing performance by adopting computational sprinting. They also present four sprinting mechanisms to handle the time-varying nature and intermittent of green energy supply. Tripathi *et al.* [13] formulate a mixed integer linear programming problem to minimize the total cost of ownership (TCO) by suitable capacity planning for highly available DGDCs. The cost of power consumption and server deployment is minimized provided that the minimum amount of green energy is consumed.

Different from the above studies, this work jointly considers spatial variations of many factors, e.g., prices of power grid, wind turbine rotor areas, on-site air density, wind speed, solar panel irradiation areas, and solar radiation in different GDCs. Then, a profit and QoS-optimized task scheduling (PQTS) method is proposed to smartly schedule tasks of all applications among DGDCs by specifying the Pareto-optimal task service rates and task split among ISPs.

B. Energy-Aware Task Scheduling

In recent years, there are several studies on energy-aware task scheduling algorithms [14]–[19]. In [15], a scheduler for parallel batch tasks is proposed to execute in a data center, which is powered by a photovoltaic solar array and power grid. It predicts the available amount of solar energy in the near future and schedules tasks to maximize the consumption of green energy while satisfying deadlines of tasks. On the other hand, it selects time when the power grid price is low if power grid has to be utilized to avoid violations of deadlines. In [16], a framework named OpenStack Neat is proposed to achieve dynamic consolidation of virtual machines (VMs) in OpenStack clouds. It is configured to adopt VM consolidation algorithms and integrated with available OpenStack deployments without changing their configuration. In [17], a scheduling system is proposed to minimize the execution time, energy consumption, or both by efficiently executing task-based applications on distributed computing platforms. A dynamic polynomial-time online algorithm is presented to combine a resource allocation method and multiple heuristic rules to obtain satisfying solutions in affordable time. Katsaros *et al.* [18] design a method to estimate and predict the ecological efficiency of VMs in private clouds. They specifically consider the information including energy availability in a region, energy consumption and resource performance, and ecological efficiency of VMs. Then, a framework is implemented to assess and forecast the ecological efficiency of a VM. The work in [19] proposes a hybrid bacteria foraging algorithm that adopts crossover and mutation operations of genetic algorithm to jointly minimize energy consumption and makespan. Then, an efficient scheduling algorithm is proposed for heterogeneous clouds.

Different from the above studies, this work aims to jointly optimize the profit of DGDC providers and ATLP of all applications by comprehensively considering spatial variations in ISP bandwidth prices, availability of green energy, prices of power grid, and revenue.

C. Performance Modeling

Recently, many studies have been conducted to achieve the QoS modeling and guaranteeing for users' tasks [20]–[25]. In [20], the average response time at each server is calculated according to an $M/M/c$ queuing system, and then, the average response time of all users is obtained. A profit optimization problem is next formulated for a cloud provider. In [21], the average response time of each application's tasks that are evenly executed by multiple homogeneous VMs is given according to $M/G/1$ queues. Then, a resource provisioning problem is formulated as a generalized Nash equilibrium one for a cloud computing system. In [22], the service latency of tasks is modeled with a nondecreasing convex function in terms of the geographical distance between users' region and each data center. Then, a constrained stochastic optimization problem is formulated and solved with a Lyapunov optimization method to obtain an online algorithm. In this way, eco-aware load scheduling and power management are jointly achieved to minimize the energy cost for DGDCs. In [23], an approximate analytical model is developed to evaluate the performance of active VMs in Infrastructure as a Service (IaaS) clouds by applying queue theory. It provides the probability distribution of the number of jobs and several performance measures, e.g., the average number of jobs, the average response time, immediate service probability, and blocking probability. The work in [24] formulates mathematical models to obtain the amount of consumed energy and the scheduling makespan of multiple tasks. It solves a multiobjective optimization problem by NSGA-II to jointly minimize both makespan and energy consumption. The artificial neural network is adopted to predict future available resources according to features of tasks.

However, these studies handle QoS of tasks by using constraints in their formulated optimization problems. Furthermore, it is a huge challenge to achieve profit maximization for multiple DGDCs than a single GDC case since the former has to consider the QoS optimization as an objective than a constraint. This is clearly more advantageous as it can give more choices to decision makers. Hence, different from above studies, this work aims to jointly maximize the profit of DGDC providers and QoS of tasks, which was not seen before.

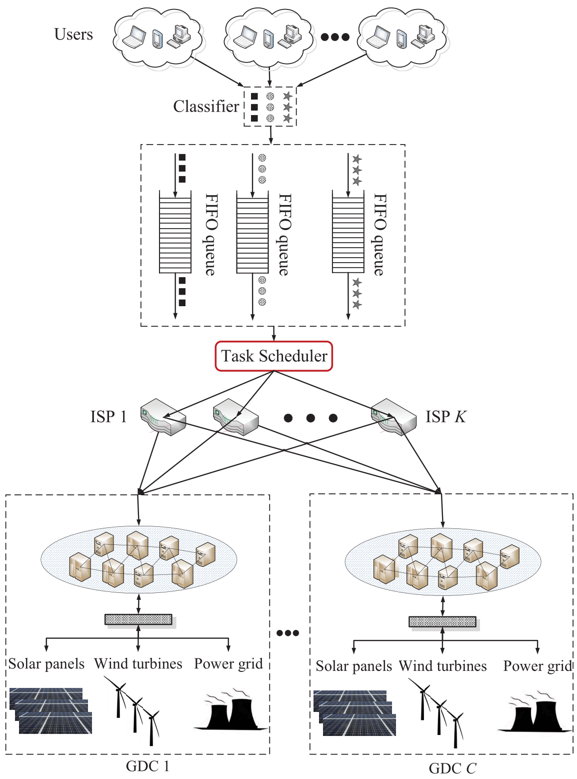


Fig. 1. Architecture of DGDCs.

III. Problem Formulation

This section formulates a multiobjective optimization problem for a DGDC provider. Fig. 1 illustrates the system architecture of DGDCs. Users send their tasks through different types of electronic devices. Then, these tasks are classified by *classifier* according to their types. Tasks of the same application are enqueued into their separate first-in-first-out (FIFO) queues. To guarantee the application availability and QoS of tasks, an application is deployed in multiple GDCs located in different locations. Then, these tasks are delivered to DGDCs through K ISPs. Each GDC manages a cluster of high-performance servers. We assume that it obtains energy from three types of sources including solar panels, wind turbines, and nonrenewable power grid. By exploiting the architecture of DGDCs, this work focuses on *task scheduler* that executes the proposed PQTS and specifies the Pareto-optimal task service rates and task split among ISPs for DGDCs in each time slot to jointly optimize profit and QoS. For clarity, we summarize parameters and decision variables in Table I in the supplementary file.

A. Delay Constraint

Let C denote the number of GDCs. Let N denote the number of applications. This work adopts the Google cluster trace[1] to evaluate PQTS. Similar to [26], it is assumed that the task service time conforms to the exponential distribution. It is proven that the Google cluster trace satisfies such assumption.

[1] https://github.com/google/cluster-data

Thus, according to the Google cluster trace, many existing studies [6]–[8] adopt the $M/M/1$ queueing model to evaluate the performance of their proposed methods with simulators or a realistic cluster of servers in data centers. Their results demonstrate that the Google trace data can effectively and well approximate the behaviors of real data centers. Therefore, similar to these studies, this work adopts the reliable $M/M/1/\beta_n^c/\infty$ queueing system to model the performance of servers of application n ($1 \le n \le N$) in GDC c ($1 \le c \le C$). Let β_n^c denote the capacity of a task queue corresponding to application n in GDC c. Let ξ_n denote the specified response time constraint of tasks of application n. Let $\lambda_\tau^{c,n}$ denote the arriving rate of tasks of application n in GDC c in time slot τ. Let $\mu_\tau^{c,n}$ denote the service rate of servers corresponding to application n in GDC c in time slot τ.

Let $\Gamma_\tau^{c,n}$ denote the average response time of tasks of application n in GDC c at time slot τ. According to [26], $\Gamma_\tau^{c,n}$ is obtained as

$$\Gamma_\tau^{c,n} = \frac{\Theta_\tau^{c,n}}{\mu_\tau^{c,n}\left(1 - Q_\tau^{c,n,0}\right)} \le \xi_n \tag{1}$$

where

$$\begin{aligned}
\Theta_\tau^{c,n} &= \frac{\rho_\tau^{c,n}}{1-\rho_\tau^{c,n}} - \frac{\left(\beta_n^c+1\right)\left(\rho_\tau^{c,n}\right)^{\beta_n^c+1}}{1-\left(\rho_\tau^{c,n}\right)^{\beta_n^c+1}}\\
Q_\tau^{c,n,0} &= \frac{1-\rho_\tau^{c,n}}{1-\left(\rho_\tau^{c,n}\right)^{\beta_n^c+1}}\\
\rho_\tau^{c,n} &= \frac{\lambda_\tau^{c,n}}{\mu_\tau^{c,n}}.
\end{aligned}$$

Then, $\Gamma_\tau^{c,n}$ must be less than or equal to ξ_n, that is,

$$\daleth_\tau^{c,n} \le \xi_n. \tag{2}$$

The service rate of tasks of application n means the rate at which tasks are scheduled to DGDCs and removed from their corresponding first-come-first-service (FCFS) queue. We have the task loss possibility of application n in GDC c in time slot τ

$$\delta\left(\lambda_\tau^{c,n}, \mu_\tau^{c,n}\right) = \begin{cases} \dfrac{1-\frac{\lambda_\tau^{c,n}}{\mu_\tau^{c,n}}}{1-\left(\frac{\lambda_\tau^{c,n}}{\mu_\tau^{c,n}}\right)^{\beta_n^c+1}}\left(\dfrac{\lambda_\tau^{c,n}}{\mu_\tau^{c,n}}\right)^{\beta_n^c}, & \mu_\tau^{c,n}>0\\ 1, & \mu_\tau^{c,n}=0. \end{cases} \tag{3}$$

Let $\lambda_\tau^{k,c,n}$ denote the arriving rate of tasks of application n in GDC c delivered through ISP k in time slot τ. To ensure the task queue stability of application n in GDC c, $\lambda_\tau^{c,n}$ must be less than $\mu_\tau^{c,n}$ in time slot τ. Then

$$\lambda_\tau^{c,n} = \sum_{k=1}^{K} \lambda_\tau^{k,c,n} < \mu_\tau^{c,n}. \tag{4}$$

Let λ_τ^n denote the task arriving rate of application n in time slot τ. In each time slot, the sum of arriving rates of tasks of application n delivered through all ISPs to all GDCs must equal λ_τ^n, that is,

$$\lambda_\tau^n = \sum_{c=1}^{C} \lambda_\tau^{c,n} = \sum_{c=1}^{C}\sum_{k=1}^{K} \lambda_\tau^{k,c,n}. \tag{5}$$

B. Energy Consumption Model

We now introduce an energy consumption model used in DGDCs. Similar to [26], it is assumed that the servers corresponding to the same application are homogeneous, and they consume the same energy. Let σ_n^c denote the number of application n's tasks executed by each corresponding powered-on server per minute in GDC c. Let $m_\tau^{c,n}$ denote the number of powered-on servers for application n in GDC c in time slot τ. Thus, $m_\tau^{c,n} = \mu_\tau^{c,n}/\sigma_n^c$. Let M_n^c denote the maximum number of servers corresponding to application n in GDC c. Then, $m_\tau^{c,n}$ must not exceed M_n^c, that is,

$$m_\tau^{c,n} \le M_n^c. \tag{6}$$

Then, the total energy consumed by DGDCs is obtained by summing up the energy consumption of all servers and the facilities including cooling and lighting in them. Let $\hat{P}_n^c$ and $\bar{P}_n^c$ denote the average peak and average idle power of each server of application n in GDC c, respectively. The power usage effectiveness value of GDC c is denoted by γ^c, and it is the ratio of the energy consumption of a GDC to that of its all servers. The power usage effectiveness is a significant metric to indicate the energy efficiency of a GDC, and its value is usually 1.2–2.0 for existing data centers.

Let L denote the length of each time slot. The number of tasks executed by each powered-on server corresponding to application n in time slot τ is calculated as

$$\frac{\left(1-\delta\left(\lambda_\tau^{c,n}, \mu_\tau^{c,n}\right)\right)\lambda_\tau^{c,n} L}{m_\tau^{c,n}}. \tag{7}$$

The busy period of each powered-on server of application n in GDC c in time slot τ is $((1-\delta(\lambda_\tau^{c,n}, \mu_\tau^{c,n}))\lambda_\tau^{c,n} L/\sigma_n^c m_\tau^{c,n})$ minutes. Let $u_\tau^{c,n}$ denote the average CPU utilization of servers corresponding to application n in GDC c in time slot τ. We have

$$u_\tau^n = \frac{\left(1-\delta\left(\lambda_\tau^{c,n}, \mu_\tau^{c,n}\right)\right)\lambda_\tau^{c,n}}{\sigma_n^c m_\tau^{c,n}}. \tag{8}$$

Thus, according to [10], the total power of GDC c in time slot τ is calculated as

$$P_\tau^c = \sum_{n=1}^{N}\left(m_\tau^{c,n}\left(\bar{P}_n^c+(\gamma^c-1)\hat{P}_n^c+\left(\hat{P}_n^c-\bar{P}_n^c\right)u_\tau^{c,n}\right)\right). \tag{9}$$

Let E_τ^c denote the total energy consumed by tasks of all applications in GDC c in time slot τ. According to (8) and (9), we have

$$E_\tau^c = \sum_{n=1}^{N}\left(\frac{g_n^c\mu_\tau^{c,n}+h_n^c\lambda_\tau^{c,n}\left(1-\delta\left(\lambda_\tau^{c,n}, \mu_\tau^{c,n}\right)\right)}{\sigma_n^c}L\right) \tag{10}$$

where

$$\begin{aligned}
g_n^c &\triangleq \bar{P}_n^c+(\gamma^c-1)\hat{P}_n^c\\
h_n^c &\triangleq \hat{P}_n^c-\bar{P}_n^c.
\end{aligned}$$

Let ϖ^c denote the available amount of the total energy in GDC c. Thus, E_τ^c must not exceed ϖ^c, that is,

$$E_\tau^c = \sum_{n=1}^{N}\left(\frac{g_n^c\mu_\tau^{c,n}+h_n^c\lambda_\tau^{c,n}\left(1-\delta\left(\lambda_\tau^{c,n}, \mu_\tau^{c,n}\right)\right)}{\sigma_n^c}L\right) \le \varpi^c. \tag{11}$$

C. Green Energy Model

The solar and wind energy models are introduced. Let $E_\tau^{c,s}$ denote the amount of solar energy generated in GDC c in time slot τ. Based on the solar energy model [27], $E_\tau^{c,s}$ is obtained as

$$E_\tau^{c,s} = \kappa^c \psi^c I_\tau^c L \tag{12}$$

where κ^c denotes the active irradiation area of solar panels, ψ^c is the efficiency of solar-to-electricity conversion, and I_τ^c is the solar irradiance in time slot τ.

Let $E_\tau^{c,w}$ denote the amount of wind energy generated in GDC c in time slot τ. Based on the wind energy model [28], we have

$$E_\tau^{c,w} = \frac{1}{2}\eta^c \alpha^c \zeta^c \left(\nu_\tau^c\right)^3 L \tag{13}$$

where η^c denotes the efficiency of wind-to-electricity conversion, α^c is the air density, ζ^c is the wind turbine rotor area, and ν_τ^c is the wind speed in time slot τ.

D. Constrained Optimization Problem

Let Ω_k denote the bandwidth limit of ISP k. Let s_n denote the average size of each task of application n. Then, the total bandwidth required by tasks of all applications transmitted through ISP k in time slot τ satisfies

$$\sum_{c=1}^{C}\sum_{n=1}^{N}\left(\lambda_\tau^{k,c,n} s_n\right) \le \Omega_k. \tag{14}$$

In real scenarios, users negotiate with current DGDC providers to establish a service-level agreement (SLA) [29]. SLA specifies the revenue brought by tasks if their performance requirements are fulfilled. In addition, the penalty cost paid by DGDC providers is also specified in SLA if its performance requirement of each task is unmet. Let θ_τ^n denote the revenue brought by the execution of each task of application n in time slot τ. Let ϑ_τ^n denote the penalty cost brought by each rejected task of application n in time slot τ. We have the total revenue brought by tasks of application n in all GDCs in time slot τ

$$\gamma_\tau^n = \sum_{c=1}^{C}\left(\left(1-\delta\left(\lambda_\tau^{c,n},\mu_\tau^{c,n}\right)\right)\lambda_\tau^{c,n}\theta_\tau^n - \delta\left(\lambda_\tau^{c,n},\mu_\tau^{c,n}\right)\lambda_\tau^{c,n}\vartheta_\tau^n\right)L. \tag{15}$$

Then, we obtain the total revenue brought by tasks of all applications in all GDCs in time slot τ

$$\Phi_\tau = \sum_{n=1}^{N}\gamma_\tau^n. \tag{16}$$

The total cost of the DGDC provider in time slot τ, denoted as Ψ_τ, consists of energy cost and ISP bandwidth cost of all tasks. Let Ψ_τ^1 denote the energy cost due to the execution of all tasks in all GDCs in time slot τ. Let Ψ_τ^2 denote the ISP bandwidth cost caused by the transmission of tasks among users and DGDCs in time slot τ

$$\Psi_\tau = \Psi_\tau^1 + \Psi_\tau^2. \tag{17}$$

It is assumed that the solar and wind energy are cost-free once they are purchased and installed in DGDCs. The amount of power grid energy consumed by all tasks is $\max(E_\tau^c - E_\tau^{c,s} - E_\tau^{c,w}, 0)$. Let p_τ^c denote the price of power grid of GDC c in time slot τ. Thus, Ψ_τ^1 is calculated as

$$\Psi_\tau^1 = \sum_{c=1}^{C}\left(p_\tau^c\left(\max\left(E_\tau^c - E_\tau^{c,s} - E_\tau^{c,w}, 0\right)\right)\right). \tag{18}$$

Let b_τ^k denote the unit bandwidth price of ISP k in time slot τ. We have

$$\Psi_\tau^2 = \sum_{k=1}^{K}\left(b_\tau^k\left(\sum_{c=1}^{C}\sum_{n=1}^{N}\left(\lambda_\tau^{k,c,n} s_n\right)\right)\right). \tag{19}$$

Let Θ_τ denote the profit of DGDC providers brought by the execution of all tasks. Θ_τ is calculated as

$$\Theta_\tau = \Phi_\tau - \Psi_\tau. \tag{20}$$

The first objective is to maximize Θ_τ, that is,

$$\max_{\lambda_\tau^{k,c,n},\mu_\tau^{c,n}} \{\Theta_\tau\}. \tag{21}$$

The ATLP is

$$\Delta_\tau = \frac{\sum_{c=1}^{C}\sum_{n=1}^{N}\left(\delta\left(\lambda_\tau^{k,c,n},\mu_\tau^{c,n}\right)\right)}{C}. \tag{22}$$

The second objective is to minimize Δ_τ, that is,

$$\min_{\lambda_\tau^{k,c,n},\mu_\tau^{c,n}} \{\Delta_\tau\}. \tag{23}$$

Then, this work formulates a biobjective optimization problem (***P***) as

$$\max_{\lambda_\tau^{k,c,n},\mu_\tau^{c,n}} \{\Theta_\tau\}$$
$$\min_{\lambda_\tau^{k,c,n},\mu_\tau^{c,n}} \{\Delta_\tau\}$$

subject to (2), (4)–(6), (11), (14) and

$$\lambda_\tau^{k,c,n} \ge 0, \quad \mu_\tau^{c,n} \ge 0. \tag{24}$$

Constraint (24) specifies ranges of decision variables, i.e., $\lambda_\tau^{k,c,n}$ and $\mu_\tau^{c,n}$. There are many prediction algorithms, e.g., radial basis function neural networks and support vector regression [30], to well predict time-related parameters, e.g., p_τ^c, b_τ^k, λ_τ^n, I_τ^c, and ν_τ^c. Therefore, similar to [10], we assume that they are already known at the start of each time slot τ. Then, the algorithm adopted to solve ***P*** is developed next.

IV. Simulated-Annealing-Based Biobjective Differential Evolution

It is worth noting that Θ_τ and Δ_τ in **P** are nonlinear in terms of $\lambda_\tau^{k,c,n}$ and $\mu_\tau^{c,n}$. Consequently, **P** is a constrained nonlinear biobjective optimization problem. Constraints (2), (4)–(6), (11), and (14) are nonlinear and complex. Therefore, to well handle them, this work adopts a penalty function method to transform **P** into its corresponding unconstrained optimization problem **P**′

$$\begin{aligned}&\min_{\lambda_\tau^{k,c,n},\mu_\tau^{c,n}} \{\hat{f}_1 = \varphi\mho - \Theta_\tau\}\\&\min_{\lambda_\tau^{k,c,n},\mu_\tau^{c,n}} \{\hat{f}_2 = \varphi\mho + \Delta_\tau\}\end{aligned} \tag{25}$$

where $\hat{f}_1$ and $\hat{f}_2$ denote the new objective functions, φ denotes a large positive integer, and $\mho$ denotes the total penalty corresponding to all constraints. Let $\vec{\hbar}$ denote a vector, including $\lambda_\tau^{k,c,n}$ and $\mu_\tau^{c,n}$. $\mho$ is calculated as

$$\mho = \sum_{z=1}^{Z}(\max\{0, -u_z(\vec{\hbar})\})^2 + \sum_{y=1}^{Y}|w_y(\vec{\hbar})|^2. \tag{26}$$

An inequality constraint z $(1 \le z \le Z)$ is first transformed into $u_z(\vec{\hbar}) \ge 0$. If it is met, its penalty is 0; otherwise, its penalty is $(-u_z(\vec{\hbar}))^2$. Similarly, an equality constraint y $(1 \le y \le Y)$ is first transformed into $w_y(\vec{\hbar}) = 0$. If it is met, its penalty is 0; otherwise, its penalty is $|w_y(\vec{\hbar})|^2$. Then, **P**′ is obtained.

Currently, there are many conventional multiobjective optimization methods that convert many objectives into a single objective by linear combination with weights, e.g., weighted sum method and weighted metric method, to solve it. However, each method has its own disadvantages. For example, it is difficult for the weighted sum method to specify the objective weight vectors to produce a Pareto-optimal solution in a selected objective space. It cannot locate certain Pareto-optimal solutions in a nonconvex objective space. In addition, it only produces one candidate solution in each run and it needs a series of different runs to obtain Pareto-optimal solutions. In reality, it is more desired to produce a set of candidate solutions in each run.

Several multiobjective evolutionary optimization algorithms are proposed to simultaneously optimize these objectives and obtain a set of Pareto-optimal solutions for decision makers, e.g., ϵ-constraint method [31], strength Pareto-evolutionary algorithm (SPEA) [32], nondominated sorting genetic algorithm II (NSGA-II) [33], and multiobjective particle swarm optimization (MOPSO) [34]. Each algorithm has its own pros and cons. For example, an ϵ-constraint method is applicable to convex and nonconvex optimization problems. However, the ϵ vector should be carefully set because it has to be in the valid range of each objective function. One major drawback of MOPSO is that it converges to a single solution at a very fast speed and causes a premature convergence problem. Thus, it often results in false Pareto-optimal solutions when it is applied to solve multi-objective optimization problems.

Differential evolution (DE) is a simple yet effective population-based direct search algorithm for typical optimization problems over continuous spaces. Compared with other evolutionary algorithms, DE is easier to implement and its evolutionary convergence speed is higher [35]. Therefore, DE has been commonly adopted to solve single-objective optimization problems because of its high robustness, simple optimization structure, and quick convergence speed. Due to its high efficiency for solving single-objective problems, DE has evolved into multiobjective DE (MODE) to solve multiobjective problems (MOPs) with complex constraints in continuous spaces. In addition, only a few parameters need to be adjusted to obtain final solutions. MODE obtains a set of desirable results with its efficient population update strategy and Pareto-based fitness ranking. Thus, it has been validated by many real-life applications, e.g., power flow optimization and facial expression recognition. However, MODE faces the problem of premature convergence.

Consequently, this work adopts an improved MODE algorithm, i.e., SBDE to solve **P**′. It is a population-based algorithm for obtaining a limited set of Pareto-optimal solutions. The Pareto dominance in iterations is enhanced, and therefore, the quality of solutions is improved. In SBDE, evolutionary operations, e.g., adaptive mutation, simulated annealing (SA)-based crossover, and SA-based selection, are performed to improve its convergence speed and accuracy. The adaptive elitist archive update mechanism is used to maintain the diversity of solutions, and an approximate well-located Pareto front is thus achieved. Typically, the Pareto-optimal set and Pareto front include numerous points. Numerically and practically, the Pareto front is approximated with a limited number of points, from which a final representative solution named a knee is selected among them. The knee is the most acceptable because it means a strategy where two objectives are traded-off but the best overall performance is achieved. The minimum Manhattan distance [36] is adopted to specify the knee from an approximate Pareto-optimal set. It has several pros including matrix computation-enabled efficient implementation, simple selection of the knee, and geometrical representation. Then, according to the knee, the final task scheduling strategy is determined to specify the optimal task service rates and task split among ISPs for DGDCs in each time slot.

A. Individual Encoding Structure

In **P**′, this work aims to specify the optimal task service rates and task split among ISPs for DGDCs in each time slot τ. Each individual i in SBDE consists of $\lambda_\tau^{k,c,n}$ and $\mu_\tau^{c,n}$. Then, it is encoded as follows:

$$x_i = \left[\mu_\tau^{1,1}, \ldots, \mu_\tau^{C,N}, \lambda_\tau^{1,1,1}, \ldots, \lambda_\tau^{K,C,N}\right]. \tag{27}$$

In (27), the first $C{*}N$ elements of x_i store $\mu_\tau^{c,n}$ $(1 \le c \le C, 1 \le n \le N)$. The next $K{*}C{*}N$ elements of x_i store $\lambda_\tau^{k,c,n}$ $(1 \le k \le K, 1 \le c \le C, 1 \le n \le N)$. Let J denote the total number of decision variables. Therefore, $J = (K{+}1){*}C{*}N$. In this way, the approach of transferring between the population and the solution of **P**′ is obtained.

B. Population Initialization

The population is randomly initialized according to a uniform distribution within the feasible search space of decision variables. Let $x_{i,j}^0$ denote the value of decision variable j of individual i $(i \in \{1, 2, \ldots, \chi\})$ in the first generation, and χ denotes the size of the population. The population is initialized as follows:

$$x_{i,j}^0 = x_j^{\min} + r_i{*}\left(x_j^{\max} - x_j^{\min}\right) \tag{28}$$

where $x_j^{\max}$ and $x_j^{\min}$ denote the upper and lower bounds of decision variable j, respectively, and r_i is a random number for individual i uniformly distributed in (0,1).

C. Adaptive Mutation

Let G denote the total number of generations. In the traditional MODE, in each generation g ($g \in \{1, 2, \ldots, G\}$), one widely used mutation operation in practice is known as DE/best/1 due to its fast convergence. It produces a new mutant individual $x_i^{g'}$ for x_i^g by perturbing the best individual x_{best}^g (best$\neq r1 \neq r2 \neq i$) searched in current generation g, with the difference of two other randomly chosen individuals in the population, $x_{r_1}^g$ and $x_{r_2}^g$. It is described as

$$x_i^{g'} = x_{\text{best}}^g + F\left(x_{r_1}^g - x_{r_2}^g\right), \quad \text{best} \neq r1 \neq r2 \neq i. \tag{29}$$

Here, F is a scaling factor that controls the perturbation and improves the convergence. However, it is worth noting that F in (29) is a constant. It fails to dynamically adjust to the evolution of population because it controls the search process with a constant convergence rate. Thus, this work designs F as a decreasing function shown as

$$F = e^{-a\frac{g}{G}} \tag{30}$$

where a is a positive constant. It is worth noting that F decreases as g increases. Then, as shown in (29), the evolution of population can explore the solution space widely at the start, and quickly at the end. Nevertheless, if the population traps into local optima, it cannot escape from them. As shown in (29), the best individual, x_{best}^g, is critically important for a mutation operation. It is the one with the least value of an objective function for a single-objective optimization problem. However, the best individual means a group of Pareto-optimal solutions for a multiobjective optimization problem. It is contradictive to guide individuals to a single solution because our goal is to obtain a set of Pareto solutions. Therefore, this work proposes an adaptive mutation operation to guide individuals to the Pareto set, and increase the global searching efficiency.

This work adopts an external archive (EA) denoted by $\digamma$ to keep the Pareto-optimal solutions searched so far. The best individual of current population is chosen from $\digamma$. Then, this work checks whether individual i is in $\digamma$. If so, i is chosen as x_{best}^g. Otherwise, the elitist individual in $\digamma$ with the minimum Euclidean distance from individual i is chosen as x_{best}^g. Let $\daleth$ denote the maximum size of $\digamma$. $d_{i,m}$ denotes the Euclidean distance between individual i and elitist individual m in $\digamma$. Based on the above discussion, the adaptive mutation is implemented as

$$x_i^{g'} = \begin{cases} x_i^g + e^{-a\frac{g}{G}}\left(x_{r_1}^g - x_{r_2}^g\right), & \text{if } x_i^g \in \digamma \\ x_{\text{best}}^g + e^{-a\frac{g}{G}}\left(x_{r_1}^g - x_{r_2}^g\right), & \text{otherwise} \end{cases} \tag{31}$$

$$\text{best} = \arg_m \min_{m \in \{1,2,\ldots,\daleth\}} (d_{i,m})$$

$$d_{i,m} = \sqrt{\sum_j^J \left(x_{i,j}^g - x_{m,j}^g\right)^2}, \quad m \in \{1, 2, \ldots, \daleth\}. \tag{32}$$

D. SA-Based Crossover

After an adaptive mutation operation, a crossover operation is implemented to produce a trial individual $x_{i,j}^{g''}$ for individual i. It is shown that the population evolution often traps into local optima, and it also causes a premature problem. To solve it, this work adopts the Metropolis acceptance criterion of SA [37] to improve the diversity of individuals in population evolution and avoid its too early convergence. SA is able to escape from local optima by conditionally accepting some worse individuals, and finally converge to global optima with high probability. Let Υ denote the number of objective functions, and $\Delta(x_i^g, x_i^{g'})$ the difference between x_i^g and $x_i^{g'}$. We have

$$\Delta\left(x_i^g, x_i^{g'}\right) = \sum_{\ell=1}^{\Upsilon} \left|\hat{f}_\ell\left(x_i^g\right) - \hat{f}_\ell\left(x_i^{g'}\right)\right|. \tag{33}$$

To improve the population diversity and avoid premature convergence, we have an SA-based crossover operation as

$$x_{i,j}^{g''} = \begin{cases} x_{i,j}^{g'}, & \text{if } e^{-\frac{\Delta\left(x_i^g, x_i^{g'}\right)}{t_g}} > \aleph \\ x_{i,j}^g, & \text{otherwise} \end{cases} \tag{34}$$

where $\aleph$ is a random number uniformly distributed in (0,1), and t_g denotes the current temperature in iteration g ($g \in \{1, 2, \ldots, G\}$).

E. SA-Based Selection

In each generation, after mutation and crossover operations, an SA-selection is implemented on each individual to check whether the newly produced trial individual $x_{i,j}^{g''}$ or x_i^g is selected as x_i^{g+1} in the next generation. Specifically, if $x_i^{g''}$ dominates x_i^g, $x_i^{g''}$ is accepted. If $e(\Delta(x_i^g, x_i^{g''})/t_g) > \aleph$, $x_i^{g''}$; otherwise, x_i^g is selected

$$x_i^{g+1} = \begin{cases} x_i^{g''}, & \text{if } x_i^{g''} \text{ dominates } x_i^g \\ x_i^{g''}, & \text{if } e^{-\frac{\Delta\left(x_i^g, x_i^{g''}\right)}{t_g}} > \aleph \\ x_i^g, & \text{otherwise.} \end{cases} \tag{35}$$

F. Adaptive Elitist Archive Update Mechanism

At the start of the population evolution, the individual size of $\digamma$ is zero. Then, a current individual is added into $\digamma$ directly. If the current individual is dominated by each elitist individual in $\digamma$, it is rejected. Otherwise, if the current individual dominates an elitist individual in $\digamma$, it is added into $\digamma$ and this dominated elitist individual is removed from $\digamma$. Besides, if a current individual and elitist individuals in $\digamma$ do not have any dominance relations, it is added into $\digamma$ and when the number of elitist individuals in $\digamma$ reaches $\daleth$, the following adaptive elitist archive update mechanism is adopted to update elitist individuals in $\digamma$. This mechanism is adopted to keep well-diversified Pareto-optimal individuals in each generation, and help the convergence of population. Let $\hat{f}_\ell^{\ddagger}$ and $\hat{f}_\ell^{\dagger}$ denote the maximum and minimum values of the ℓ th ($\ell \in \{1, 2, \ldots, \Upsilon\}$) objective function, that is,

$$\begin{cases} \hat{f}_\ell^{\dagger} = \min\limits_{i \in \{1,2,\ldots,\daleth\}} \left(\hat{f}_\ell\left(x_i^g\right)\right), & \ell \in \{1, 2, \ldots, \Upsilon\} \\ \hat{f}_\ell^{\ddagger} = \max\limits_{i \in \{1,2,\ldots,\daleth\}} \left(\hat{f}_\ell\left(x_i^g\right)\right), & \ell \in \{1, 2, \ldots, \Upsilon\}. \end{cases} \tag{36}$$

Algorithm 1 SBDE

```
1: Perform the initialization with (28)
2: $g \leftarrow 1$
3: $\digamma \leftarrow \emptyset$
4: while $g \leq G$ do
5:   for $i \leftarrow 1$ to $\chi$ do
6:     Perform the adaptive mutation with (31)
7:     Perform the SA-based crossover with (34)
8:     Perform the SA-based selection with (35)
9:   end for
10:  Change $\digamma$ with the adaptive elitist archive update mechanism
11:  $g \leftarrow g+1$
12:  $t_g \leftarrow t_g * c_r$
13: end while
14: Select the knee solution:
    $x^* = \arg_x \underset{x \in \digamma}{\mathrm{Min}} \| [\frac{\hat{f}_1(x) - \hat{f}_{1\min}}{\hat{f}_{1\max} - \hat{f}_{1\min}}, \frac{\hat{f}_2(x) - \hat{f}_{2\min}}{\hat{f}_{2\max} - \hat{f}_{2\min}}] - [1, 0] \|_1$
15: Output $x^*$ and its two objectives $[\hat{f}_1(x^*),\ \hat{f}_2(x^*)]$
```

Let R_ℓ denote the range of the ℓ th objective function, i.e., $R_\ell = \hat{f}_\ell^{\ddagger} - \hat{f}_\ell^{\dagger}$. Let $\underline{b}_\ell$ and $\bar{b}_\ell$ denote the lower and higher bounds of the grid region for the ℓth objective function, that is,

$$\begin{cases} \underline{b}_\ell = \hat{f}_\ell^{\dagger} - \frac{1}{2\beth} R_\ell \\ \bar{b}_\ell = \hat{f}_\ell^{\ddagger} + \frac{1}{2\beth} R_\ell. \end{cases} \tag{37}$$

The objective space specified by individuals in current $\digamma$ is separated into $\beth^2$ hypercubes of equal size, and $\beth$ denotes the grid number in any dimension of the space. The number of elitist individuals in each hypercube is calculated, and the hypercube with the most individuals is recorded as the most congested one. Then, when the number of elitist individuals in current $\digamma$ reaches $\daleth$, the following adaptive elitist archive update mechanism is invoked.

1) If an individual is within its current objective space, it is put into $\digamma$ and added to its corresponding hypercube. Besides, an elitist individual is removed randomly from the most congested grid.
2) If an individual is outside the current objective space, it is also put into $\digamma$. The new objective space is then determined, and hypercubes are separated again. The number of elitist individuals in each hypercube is also obtained again. Then, an elitist individual is removed randomly from the most congested grid.

It is worth noting that the extreme elitist individuals are reserved and not removed from $\digamma$. Then, the size of $\digamma$ is fixed and the individual diversity of $\digamma$ is increased.

Algorithm 1 presents the pseudo code of SBDE. Line 1 initializes the population with (28). The EA, $\digamma$, is initialized with $\emptyset$ in Line 3. In each generation g, Line 6 performs the adaptive mutation for each individual i with (31). Line 7 performs the SA-based crossover for each individual i with (34). Then, Line 8 performs the SA-based selection for each individual i with (35). Line 10 changes $\digamma$ with an adaptive elitist archive update mechanism. Let c_r denote the cooling rate of temperature. Line 12 decreases the temperature (t_g) in generation g by c_r. Line 14 selects the knee solution x^*. Finally, Line 15 outputs the knee solution x^* and its two objectives $[\hat{f}_1(x^*),\ \hat{f}_2(x^*)]$.

Furthermore, this work discusses the assumptions of the proposed PQTS as regards their validity in real-time situations. We assume that the future data (e.g., arriving rates of tasks, solar irradiance, wind speed, prices of power grid, and unit bandwidth prices) are already known at the start of each time slot. The emerging deep learning algorithms and the wide deployment of high-performance servers in DGDCs make it possible that the prediction of above information can be performed in a real-time manner.

TABLE I

PARAMETER SETTING OF POWER GRID, WIND, AND SOLAR ENERGY

	Power grid		Wind energy			Solar energy	
	γ^c	ϖ^c(WH)	η^c	ζ^c(m^2)	α^c	κ^c(m^2)	ψ^c
c=1	1.2	1.7×10^8	0.3	25000	1.225	15000	0.2
c=2	1.4	2.25×10^8	0.375	31250	1.5313	18750	0.25
c=3	1.6	1×10^8	0.45	37500	1.8375	22500	0.3

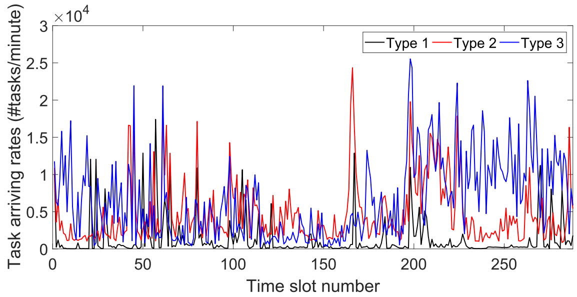


Fig. 2. Task arriving rates of three applications.

V. Performance Evaluation

This section evaluates PQTS by using realistic data including task arriving rates of three applications sampled every 5 min in Google cluster, prices of power grid,[2] bandwidth prices of ISPs, solar irridiance,[3] wind speed,[4] and so on for 24 h on May 10, 2011, as shown in Table I.

A. Parameter Setting

Similar to the realistic setting [38], [39], this work considers three ISPs and three GDCs, i.e., $K = 3$ and $C = 3$. Fig. 2 illustrates task arriving rates of three applications ($N = 3$). Their parameters including σ_n^c, $\bar{P}_n^c$, $\hat{P}_n^c$, β_n^c, and M_n^c are set in Table II. According to [40], $\Omega_1 = 4\times10^6$ (Mb/s), $\Omega_2 = 5\times10^6$ (Mb/s), and $\Omega_3 = 6\times10^6$ (Mb/s). Besides, $s_1 = 8$ (Mb), $s_2 = 5$ (Mb), $s_3 = 2$ (Mb), $\xi_1 = 0.15$ (s), $\xi_2 = 0.2$ (s), and $\xi_3 = 0.25$ (s). According to [41], the parameters of SBDE are set as follows. $\chi = 40$, $G = 100$, $\daleth = 20$, $t_1 = 10^{12}$, $c_r = 0.975$, $a = 0.2$, $\beth = 80$, and $\varphi = 10^{20}$.

B. Experimental Results

Fig. 3 illustrates the convergence analysis in terms of profit and temporary knee distance of SBDE in time slot 100.

[2] http://www.nyiso.com/public/index.jsp
[3] http://www.nrel.gov/midc/srrl_bms/
[4] http://www.nrel.gov/midc/nwtc_m2/

TABLE II
PARAMETER SETTING OF THREE GDCS

	σ_n^c (tasks/sec.)			$\bar{P}_n^c$ (W)			$\hat{P}_n^c$ (W)			β_n^c			M_n^c		
	n=1	n=2	n=3	n=1	n=2	n=3	n=1	n=2	n=3	n=1	n=2	n=3	n=1	n=2	n=3
c=1	0.15	0.3	0.6	200	100	50	400	200	100	50	55	60	1200	1500	1800
c=2	0.15	0.3	0.6	250	125	62.5	500	250	125	55	60	65	1000	1250	1500
c=3	0.15	0.3	0.6	300	150	75	600	300	150	65	70	75	1200	1500	1800

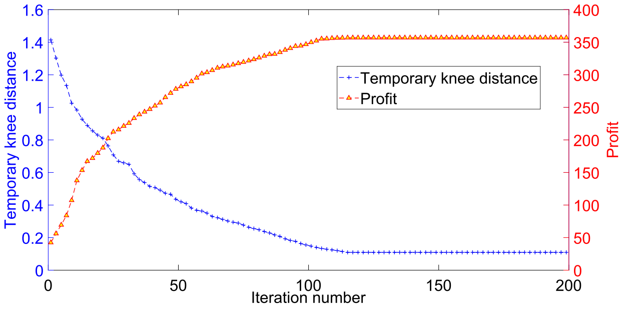


Fig. 3. Convergence analysis of SBDE.

TABLE III
CONVERGENCE TIME WITH DIFFERENT SCALES OF TASKS

Scale	$\frac{1}{4}$	$\frac{1}{2}$	1	2	4
Convergence time (s)	2.21	3.99	6.77	13.13	25.08

Here, the temporary knee distance in each iteration is calculated as follows. In Algorithm 1, each iteration runs Lines 5–12. $\mathcal{F}$ is updated with an adaptive elitist archive update mechanism. Therefore, in each iteration, a temporary knee solution with the minimum knee distance is selected according to the rule of knee solution selection in Line 14. The temporary knee distance in each iteration is obtained accordingly, and shown in Fig. 3. It is observed that it reduces as the number of iterations increases. Besides, it is also shown that the profit increases with the evolution of SBDE. Fig. 3 illustrates that the stable values are approached and demonstrates the convergence of SBDE.

Table III shows the convergence time with different scales of tasks. For example, if scale is 2, the task arriving rate of application n in time slot τ is doubled and it becomes $2\lambda_\tau^n$. Similarly, if scale is (1/2), it becomes $(1/2)\lambda_\tau^n$. Consequently, it is observed that the convergence time increases with the increase of the scales of tasks. This work considers the ATLP optimization as a second objective rather than a constraint, and this gives more advantageous options to decision makers. The adding QoS as an objective increases the execution time of SBDE in comparison with single-objective constrained optimization algorithms. However, this work adopts the operations of adaptive mutation, SA-based crossover and selection, knee solution selection, and adaptive elitist archive update to improve the convergence speed and accuracy of SBDE. It is also worth noting that similar to [22], the length of each time slot is 5 min, and tasks are scheduled based on the knee solution x^* obtained by SBDE from one time slot to another. Therefore, compared to length of each time slot, the convergence time is negligible and does not affect the scheduling of tasks. This means that $\boldsymbol{P}$ is solved in a real-time way.

Fig. 4 illustrates the number of powered-on servers in three GDCs, respectively. It is shown that the number of powered-on servers for each application in three GDCs is less than its corresponding limit. Besides, it is observed that the number of powered-on servers for each application in three GDCs varies. For example, the number of powered-on servers for application 2 in GDC 3 is much smaller than those in GDCs 1 and 2. The reason is that prices of power grid at GDCs 1 and 2 are less than that at GDC 3 in each time slot, and more tasks are therefore scheduled to GDCs 1 and 2. Accordingly, this yields a lower number of powered-on servers in GDC 3.

C. Comparison Results

To demonstrate the performance of SBDE, this work first compares it with several state-of-the-art multiobjective evolutionary algorithms, including NSGA-II [42], SPEA-II [32], and MODE [41]. It is worth noting that each algorithm obtains its own Pareto front, which consists of a set of different solutions. Then, the aforementioned minimum Manhattan distance method is adopted to select a knee solution from the Pareto front of each algorithm. Four knee solutions are further compared with each other in terms of profit and ATLP. Table IV shows that SBDE performs better than its three peers in terms of both profit and ATLP. Compared with SBDE, NSGA-II has 3.53% reduction in profit and 37.50% increase in ATLP, SPEA-II has 7.06% reduction in profit and 75.00% increase in ATLP, and MODE has 17.65% reduction in profit and 88.64% increase in ATLP. Furthermore, the convergence time of the Pareto-optimal front obtained by SBDE is 6.77 s, which is negligible because the length of each time slot is 5 min. The results are explained as follows. NSGA-II, SPEA-II, and MODE all converge to worse Pareto fronts with low diversity of populations. Different from NSGA-II, SPEA-II, and MODE, SBDE adopts adaptive mutation, SA-based crossover, SA-based selection, adaptive elitist archive update mechanism, and the minimum Manhattan distance to improve the diversity of solutions, convergence speed, and accuracy. Thus, the Pareto-optimal front obtained by SBDE is distributed more evenly and diversely than other three algorithms. This means that SBDE provides more comprehensive candidate solutions. Consequently, the profit of SBDE is the largest, while its ATLP is the lowest among four algorithms.

To prove the performance of PQTS, it is further compared with three typical task scheduling approaches [43]–[45] in terms of profit and ATLP.

1) M1 [43] is electricity-cost-aware distributed task scheduling.
2) M2 [44] is profit maximization approach (PMA).
3) M3 [45] is green task balancing (GTB).

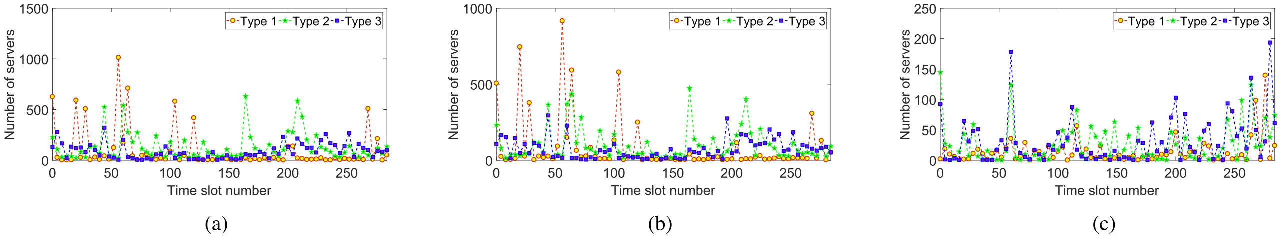


Fig. 4. Number of powered-on servers in GDCs (a) 1, (b) 2, and (c) 3.

TABLE IV
COMPARISON AMONG OPTIMIZATION ALGORITHMS

Algorithms	Knee solution	
	Profit ($)	ATLP
NSGA-II [44]	3316.2	0.004
SPEA-II [33]	3194.9	0.010
MODE [42]	2830.9	0.022
SBDE	3437.5	0.003

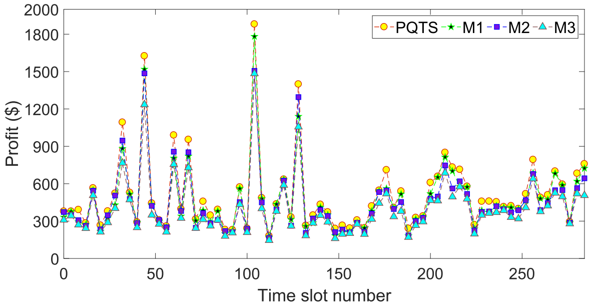


Fig. 5. Profit comparison of PQTS and M1–M3.

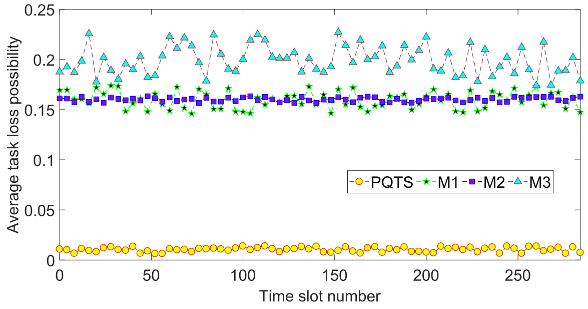


Fig. 6. ATLP comparison of PQTS and M1–M3.

Figs. 5 and 6 show the comparison of profit and ATLP of PQTS and M1–M3 on average. The ATLP of PQTS is the lowest and its profit is the highest compared with M1–M3. Its profit is 8.40% higher and its ATLP is 93.47% lower than M1's; 11.72% higher and its ATLP is 93.50% lower than M2's; and 20.66% higher and its ATLP is 94.75% lower than M3's. The reasons are explained as follows. The profit of M3 is the least because it aims to achieve the energy cost minimization rather than the profit maximization. In M3, the number of powered-on servers is minimized provided that it is sufficient to execute all tasks. Therefore, M3 achieves larger ATLP and smaller profit. Similarly, M1 aims to minimize the energy cost of distributed data centers with intelligent load balancing. M2 aims to maximize the profit of a GDC with convex optimization for two cases, with and without behind-the-meter green generators. However, they both do not consider the bandwidth cost and capacity limits of ISPs, and the availability of wind and solar energy in DGDCs. Therefore, the profits of M1 and M2 are lower than that of PQTS. M1 and M2 both model QoS as a constraint in its optimization problem, and therefore the minimum QoS satisfaction is achieved. Different from them, PQTS jointly maximizes the profit of DGDC providers, and minimizes the ATLP of all applications. Therefore, it achieves an excellent balance between two objectives and realizes higher profit and better QoS for DGDC providers than M1–M3.

VI. CONCLUSION

The number of tasks of global users significantly increases the energy cost of DGDCs while promoting the revenue of DGDC providers. The latter is closely related to tasks' QoS. Besides, many factors in different GDCs, e.g., prices of power grid, wind, and solar energy, exhibit spatial variations. Consequently, it becomes a big challenge to jointly maximize the profit of DGDC providers and minimize the ATLP of all customer applications. This work proposes a PQTS method to achieve a beneficial tradeoff between these two objectives for both users and DGDCs. Specifically, this work formulates a biobjective optimization problem and solves it with an SBDE algorithm. The minimum Manhattan distance is used to specify a knee solution that determines proper task split among ISPs and task service rates at DGDCs in each time slot. Real-life data-based simulations reveal that the proposed method achieves significantly higher profit and lower ATLP of all applications than several existing task scheduling algorithms. In the future, we will further extend our current work to use more advanced queueing models, e.g., Pareto queues with the tailed distribution assumption [46] to analyze the performance of DGDCs.

REFERENCES

[1] E. Baccarelli, N. Cordeschi, A. Mei, M. Panella, M. Shojafar, and J. Stefa, "Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: Review, challenges, and a case study," *IEEE Netw.*, vol. 30, no. 2, pp. 54–61, Mar. 2016.

[2] X. Xiao, W. Zheng, Y. Xia, X. Sun, Q. Peng, and Y. Guo, "A workload-aware VM consolidation method based on coalitional game for energy-saving in cloud," *IEEE Access*, vol. 7, pp. 80421–80430, 2019.

[3] J. Shuja *et al.*, "Survey of techniques and architectures for designing energy-efficient data centers," *IEEE Syst. J.*, vol. 10, no. 2, pp. 507–519, Jun. 2016.

[4] Y. Xia, M. Zhou, X. Luo, Q. Zhu, J. Li, and Y. Huang, "Stochastic modeling and quality evaluation of infrastructure-as-a-service clouds," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 162–170, Jan. 2015.

[5] L. Gu, D. Zeng, A. Barnawi, S. Guo, and I. Stojmenovic, "Optimal task placement with QoS constraints in geo-distributed data centers using DVFS," *IEEE Trans. Comput.*, vol. 64, no. 7, pp. 2049–2059, Jul. 2015.

[6] M. Hähnel, J. Martinovic, G. Scheithauer, A. Fischer, A. Schill, and W. Dargie, "Extending the cutting stock problem for consolidating services with stochastic workloads," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 11, pp. 2478–2488, Nov. 2018.

[7] X. Ma, S. Wang, S. Zhang, P. Yang, C. Lin, and X. S. Shen, "Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2019.2903240.

[8] M. S. Yoon, A. E. Kamal, and Z. Zhu, "Adaptive data center activation with user request prediction," *Comput. Netw.*, vol. 122, pp. 191–204, Jul. 2017.

[9] Y. Xia, M. Zhou, X. Luo, S. Pang, and Q. Zhu, "A stochastic approach to analysis of energy-aware DVS-enabled cloud datacenters," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 73–83, Jan. 2015.

[10] A. Kiani and N. Ansari, "Profit maximization for geographically dispersed green data centers," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 703–711, Mar. 2018.

[11] H. Lei, R. Wang, T. Zhang, Y. Liu, and Y. Zha, "A multi-objective co-evolutionary algorithm for energy-efficient scheduling on a green data center," *Comput. Oper. Res.*, vol. 75, pp. 103–117, Nov. 2016.

[12] H. Cai *et al.*, "GreenSprint: Effective computational sprinting in green data centers," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2018, pp. 690–699.

[13] R. Tripathi, S. Vignesh, and V. Tamarapalli, "Optimizing green energy, cost, and availability in distributed data centers," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 500–503, Mar. 2017.

[14] W. Li, K. Liao, Q. He, and Y. Xia, "Performance-aware cost-effective resource provisioning for future grid IoT-cloud system," *J. Energy Eng.*, vol. 145, no. 5, pp. 1–15, Jul. 2019.

[15] Í. Goiri *et al.*, "Matching renewable energy supply and demand in green datacenters," *Ad Hoc Netw.*, vol. 25, pp. 520–534, Feb. 2015.

[16] A. Beloglazov and R. Buyya, "OpenStack Neat: A framework for dynamic and energy efficient consolidation of virtual machines in openstack clouds," *Concurr. Comput., Pract. Exper.*, vol. 27, no. 5, pp. 1310–1333, Apr. 2015.

[17] F. Juarez, J. Ejarque, R. Badia, S. González Rocha, and O. Esquivel-Flores, "Energy-aware scheduler for HPC parallel task base applications in cloud computing," *Int. J. Combinat. Optim. Probl. Informat.*, vol. 9, no. 1, pp. 54–61, Feb. 2018.

[18] G. Katsaros, P. Stichler, J. Subirats, and J. Guitart, "Estimation and forecasting of ecological efficiency of virtual machines," *Future Gener. Comput. Syst.*, vol. 55, pp. 480–494, Feb. 2016.

[19] S. Srichandan, T. A. Kumar, and S. Bibhudatta, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm," *Future Comput. Inf. J.*, vol. 3, no. 2, pp. 210–230, Dec. 2018.

[20] C. Liu, K. Li, K. Li, and R. Buyya, "A new cloud service mechanism for profit optimizations of a cloud provider and its users," *IEEE Trans. Cloud Comput.*, to be published.

[21] D. Ardagna, M. Ciavotta, and M. Passacantando, "Generalized Nash equilibria for the service provisioning problem in multi-cloud systems," *IEEE Trans. Services Comput.*, vol. 10, no. 3, pp. 381–395, May 2017.

[22] X. Deng, D. Wu, J. Shen, and J. He, "Eco-aware online power management and load scheduling for green cloud datacenters," *IEEE Syst. J.*, vol. 10, no. 1, pp. 78–87, Mar. 2016.

[23] X. Chang, B. Wang, J. K. Muppala, and J. Liu, "Modeling active virtual machines on IaaS clouds using anM/G/m/m+KQueue," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 408–420, Jun. 2016.

[24] A. S. Sofia and P. GaneshKumar, "Multi-objective task scheduling to minimize energy consumption and makespan of cloud computing using NSGA-II," *J. Netw. Syst. Manage.*, vol. 26, no. 2, pp. 463–485, Sep. 2017.

[25] W. B. Zheng *et al.*, "Percentile performance estimation of unreliable IaaS clouds and their cost-optimal capacity decision," *IEEE Access*, vol. 5, pp. 2808–2818, 2017.

[26] J. Yao, H. Guan, J. Luo, L. Rao, and X. Liu, "Adaptive power management through thermal aware workload balancing in Internet data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 9, pp. 2400–2409, Sep. 2015.

[27] F. Valencia, J. Collado, D. Sáez, and L. G. Marín, "Robust energy management system for a microgrid based on a fuzzy prediction interval model," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1486–1494, May 2016.

[28] H. H. Abdeltawab and Y. A.-R. I. Mohamed, "Robust energy management of a hybrid wind and flywheel energy storage system considering flywheel power losses minimization and grid-code constraints," *IEEE Trans. Ind. Electron.*, vol. 63, no. 7, pp. 4242–4254, Jul. 2016.

[29] U. Franke and M. Buschle, "Experimental evidence on decision-making in availability service level agreements," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 1, pp. 58–70, Mar. 2016.

[30] A. Rubio-Solis, P. Melin, U. Martinez-Hernandez, and G. Panoutsos, "General type-2 radial basis function neural network: A data-driven fuzzy model," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 2, pp. 333–347, Feb. 2019.

[31] Z. S. Morabi, M. S. Owlia, M. Bashiri, and M. H. Doroudyan, "Multi-objective design of $\overline{X}$ control charts with fuzzy process parameters using the hybrid epsilon constraint PSO," *Appl. Soft Comput.*, vol. 30, pp. 390–399, May 2015.

[32] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 329–346, Jun. 2017.

[33] W. K. Mashwani, A. Salhi, O. Yeniay, H. Hussian, and M. A. Jan, "Hybrid non-dominated sorting genetic algorithm with adaptive operators selection," *Appl. Soft Comput.*, vol. 56, pp. 1–18, Jul. 2017.

[34] Y. Zhang, D. W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 64–75, Jan. 2017.

[35] N. Lynn, M. Z. Ali, and P. N. Suganthan, "Population topologies for particle swarm optimization and differential evolution," *Swarm Evol. Comput.*, vol. 39, no. 4, pp. 24–35, Apr. 2018.

[36] W.-Y. Chiu, G. G. Yen, and T.-K. Juan, "Minimum manhattan distance approach to multiple criteria decision making in multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 972–985, Dec. 2016.

[37] S. Lyden and M. E. Haque, "A simulated annealing global maximum power point tracking approach for PV modules under partial shading conditions," *IEEE Trans. Power Electron.*, vol. 31, no. 6, pp. 4171–4181, Jun. 2016.

[38] J. Luo, L. Rao, and X. Liu, "Spatio-temporal load balancing for energy cost optimization in distributed Internet data centers," *IEEE Trans. Cloud Comput.*, vol. 3, no. 3, pp. 387–397, Sep. 2015.

[39] J. Yao, H. Zhou, J. Luo, X. Liu, and H. Guan, "COMIC: Cost optimization for Internet content multihoming," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 7, pp. 1851–1860, Jul. 2015.

[40] H. Yuan, J. Bi, W. Tan, and B. H. Li, "CAWSAC: Cost-aware workload scheduling and admission control for distributed cloud data centers," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 976–985, Apr. 2016.

[41] N. Taran, D. M. Ionel, and D. G. Dorrell, "Two-level surrogate-assisted differential evolution multi-objective optimization of electric machines using 3-D FEA," *IEEE Trans. Magn.*, vol. 54, no. 11, pp. 1–5, Nov. 2018.

[42] U. Singh and S. N. Singh, "Optimal feature selection via NSGA-II for power quality disturbances classification," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 2994–3002, Jul. 2018.

[43] A. K. Kiani and N. Ansari, "On the fundamental energy trade-offs of geographical load balancing," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 170–175, May 2017.

[44] J. Mei, K. Li, Z. Tong, Q. Li, and K. Li, "Profit maximization for cloud brokers in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 1, pp. 190–203, Jan. 2019.

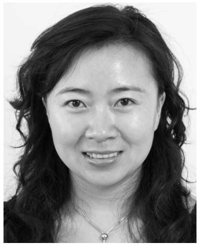
[45] K. Wang, H. Li, S. Maharjan, Y. Zhang, and S. Guo, "Green energy scheduling for demand side management in the smart grid," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 2, pp. 596–611, Jun. 2018.

[46] Y. Wang, H. Liu, W. Zheng, Y. Xia, Y. Li, P. Chen, K. Guo, and H. Xie, "Multi-objective workflow scheduling with deep-q-network-based multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 39974–39982, 2019.

Haitao Yuan (S'15–M'17) received the B.S. and M.S. degrees in software engineering from Northeastern University, Shenyang, China, in 2010 and 2012, respectively, and the Ph.D. degree in control science and engineering from Beihang University, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology (NJIT), Newark, NJ, USA.

He was a Ph.D. student with the Department of Computer Science, The City University of Hong Kong, Hong Kong, from 2013 to 2014. He was also a Visiting Doctoral Student with NJIT in 2015. He has authored or coauthored over 30 publications in international journals and conference proceedings, including the IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Services Computing, IEEE Transactions on Industrial Informatics, and IEEE Transactions on Cybernetics. His research interests include cloud computing, data center, big data, machine learning, optimization algorithms, and software-defined networking.

Dr. Yuan was the recipient of the 2011 Google Excellence Scholarship and the Best Paper Award-Finalist in IEEE ICNSC'19.



Jing Bi (SM'16) received the B.S. and Ph.D. degrees in computer science from Northeastern University, Shenyang, China, in 2003 and 2011, respectively.

She was a Post-Doctoral Researcher with the Department of Automation, Tsinghua University, Beijing, China. She was a Research Scientist with the Beijing Research Institute of Electronic Engineering Technology, Beijing. She was a Research Assistant and participated in research on cloud computing at IBM Research, Beijing. She was a Visiting Research Scholar with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. She is currently an Associate Professor with the Faculty of Information Technology, School of Software Engineering, Beijing University of Technology, Beijing. She has authored or coauthored over 80 publications including journal and conference articles. Her research interests are in distributed computing, cloud computing, large-scale data analytics, machine learning, and performance optimization.

Dr. Bi was the recipient of the IBM Fellowship Award and the Best Paper Award-Finalist in the 16th IEEE International Conference on Networking, Sensing and Control. She is currently an Associate Editor of IEEE Access.

MengChu Zhou (S'88–M'90–SM'93–F'03) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

In 1990, he joined the New Jersey Institute of Technology (NJIT), Newark, NJ, USA, where he is currently a Distinguished Professor of electrical and computer engineering. He has authored or coauthored over 800 publications, including 12 books, 460+ journal articles (360+ in IEEE Transactions), 11 patents, and 28 book chapters. His research interests are in Petri nets, intelligent automation, Internet of Things, big data, Web services, and intelligent transportation.

Dr. Zhou is a Life Member of the Chinese Association for Science and Technology—USA and served as its President in 1999. He is a fellow of the International Federation of Automatic Control (IFAC), American Association for the Advancement of Science (AAAS), and Chinese Association of Automation (CAA). He is a recipient of Humboldt Research Award for U.S. Senior Scientists from Alexander von Humboldt Foundation, Franklin V. Taylor Memorial Award, and the Norbert Wiener Award from IEEE Systems, Man and Cybernetics Society. He is the founding Editor of the IEEE Press Book Series on Systems Science and Engineering and the Editor-in-Chief of the IEEE/CAA Journal of Automatica Sinica.

Qing Liu received the B.S. and M.S. degrees from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer engineering from The University of New Mexico, Albuquerque, NM, USA, in 2008.

He was a Staff Scientist with the Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA, for seven years. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. He has authored or coauthored over 70 peer-reviewed journal and conferences, including the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Computers, ACM-TOS, IEEE Network, IEEE Communications Letters, Proceedings of SIGMETRICS, SC, IPDPS, HPDC, ICDCS, CLUSTER, and so on.

Dr. Liu was nominated for the best paper for IPDPS'18 and ICCCN'08 and won the best paper for NAS'19. He received the 100 research and development awards as a Principal Investigator for his contributions to scientific data management.

Ahmed Chiheb Ammari (M'12–SM'15) received the B.S. degree in electrical engineering from the Ecole Nationale des Ingenieurs de Monastir, Monastir, Tunisia, in 1993, and the M.Sc. and Ph.D. degrees in electrical engineering from the Institut National Polytechnique de Grenoble, Grenoble, France, in 1993 and 1996, respectively.

Since 1997, he has been a Faculty Member with the Institut National des Sciences Appliques et de Technologies, Carthage University, Tunis, Tunisia. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, College of Engineering, Sultan Qaboos University, Muscat, Oman. His main research interests include multicore and multiprocessor system on-chip, embedded computer vision, energy-efficient computing, inductive data, and power transfer for implantable medical devices, hybrid electric energy storage systems, and system-level modeling and optimization for smart grid and data centers.