IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 May 2023, accepted 24 May 2023, date of publication 29 May 2023, date of current version 6 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3280930

RESEARCH ARTICLE

An Integrated Optimization-Based Algorithm for Energy Efficiency and Resource Allocation in Heterogeneous Cloud Computing Centers

KUANG-YEN TAI[1], FRANK YEONG-SUNG LIN[1], AND CHIU-HAN HSIAO[2], (Member, IEEE)
[1]Department of Information Management, National Taiwan University, Taipei 106, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan

Corresponding author: Kuang-Yen Tai (davidking53211@gmail.com)

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant 110-2221-E-002-078-MY2; and in part by the Academia Sinica, Taiwan.

ABSTRACT At a significant moment in the rapid development of cloud technology, large-scale cloud computing centers have emerged. With the emergence of the internet and artificial intelligence, enormous computing resources are required to process data and train machine learning models. The architecture of cloud computing centers involves millions of computing resources, and improper management of these resources can increase operating costs and exert tremendous pressure on the environment. This study proposes an optimized computing resource and energy management algorithm for computing centers with heterogeneous computing resources from the perspective of Green IT. Specifically, this study models the energy consumption at each point in time and the relationship between tasks and also considers the calculation of data backup. This approach will be expanded to optimize decisions for all computing tasks in computing centers based on the sequence of tasks and energy consumption while considering heterogeneous computing resources, energy efficiency, task scheduling, and execution time. By modeling this issue as a highly nonlinear optimization problem and utilizing mathematical programming and Lagrangian relaxation, we propose an optimized energy management algorithm to effectively manage computing resources and create cloud computing centers with high performance and low energy consumption.

INDEX TERMS
Energy management, heterogeneous computing resources, green IT, optimization, Lagrangian relaxation.

I. INTRODUCTION

Today's representative academic research institutions and enterprises have their own cloud computing centers for operations such as training artificial intelligence models [1]. A cloud computing center consists of multiple tasks that work together to provide convenient and dynamic computing resources. Its architecture includes various computer systems and thousands to millions of heterogeneous computing resource servers. If these computing resources cannot be adequately controlled and scheduled, it causes considerable energy waste. Study pointed out that from 2011 to 2035, the energy demand on computing centers will increase by more than 66%, and the energy consumption of idle server accounts for 70% of the maximum energy consumption [2]. This energy consumption will also exert tremendous pressure on the environment.

Building an efficient and energy-saving cloud computing center with optimized computing resources has become an important issue. Most computing centers must process various types of work tasks in parallel. Reducing the energy consumption of cloud computing centers with heterogeneous computing resources is a significant challenge. These computing centers need to follow the different interpretations of their applications [3], including consistently supporting processing, overlapping processes, and workloads that need immediate support for immediate processing types. Consistently supporting workloads such as web browsing, online

The associate editor coordinating the review of this manuscript and approving it for publication was Cesar Vargas-Rosales.

games, data queries, and other work items is pertinent. Supplementary processing workloads such as data backup, log analysis, image processing, scientific applications, and financial data analysis can be planned so that they are completed before a set deadline time [4].

Energy-saving algorithms used in computing centers include static power management (SPM) and dynamic power management (DPM) technology. In the past, most studies used static energy consumption as a guideline for workload management. The static energy consumption in a cloud computing center environment includes a continuous power supply to servers, storage devices, cooling equipment, and other essential equipment. However, external factors such as time and climate affect the computing center's energy consumption [5].

Dynamic power management is adjusted according to the balance between the computing center's basic structure and the actual workload. The standard DPM method shuts down idle servers and restarts them as needed [6]. However, it is challenging to calculate the time sequence of processing tasks [7], external environmental changes, and the impact of heterogeneous computing resources on energy consumption at the same time [8], [9], [10].

The purpose of this study is to propose an optimized solution based on the energy and computing resource management architecture of a heterogeneous computing cloud computing center. Heterogeneous computing resources, energy consumption, task scheduling and execution time are comprehensively considered [11]. In a computing center with heterogeneous computing resources, the proposed algorithm can effectively reduce energy consumption to reduce the computing center's carbon emissions, quantify the central processing unit (CPU) or graphics processing unit (GPU) core allocation, and provide appropriate task scheduling according to the work content and execution time. In terms of energy consumption control methods, we set the upper limit of energy consumption and appropriately control the on and off time for each server so that we can set the upper power limit through dynamic voltage and frequency scaling to achieve lower energy consumption and improve computing effectiveness. This study proposes a mathematical model that describes the interrelationships among tasks in the cloud computing center, the allocation of computing resources and energy consumption, and the use of variables and parameters to simulate and seek an optimized algorithm. The detailed objectives are as follows:

(1) Computing resources: The proposed algorithm considers various computing resources, including the CPU, GPU, multi-processor system, computer clusters, and what type of equipment and number of operations are needed at all time points.

(2) Computing environment: The proposed algorithm considers the energy consumption of the overall computing center, the power consumption of each device, the opening and closing time, and whether there is a specific energy usage plan.

(3) Task scheduling: The proposed algorithm considers whether all tasks required by the overall cloud computing center need to be processed immediately or can be planned so that they are complete before their deadline.

(4) Optimal solution: We modeled the energy consumption of each end of time and refer to current energy management solutions such as SPM and DPM [12], [13] to compensate for the proposed algorithm's deficiencies and develop new solutions.

This paper is structured as follows. Following the introduction in Section I, Section II presents related research topics and practices. Section III describes the mathematical definition of the proposed model, which takes into consideration continuous process time and energy consumption. In Section IV, the LR-based optimal solution approach used in this study is introduced. Section V presents the experimental results in different scenarios and comparisons to the proposed methods. Finally, Section VI concludes this study and suggests future work.

II. RELATED WORKS

A. CLOUD COMPUTING CENTERS WITH HETEROGENEOUS COMPUTING RESOURCES

There are different computing resource servers in cloud computing centers with a heterogeneous structure [5], which are divided into CPUs and GPUs according to the type of arithmetic logic unit. Each CPU core has relative temporary storage and alternate logical operation units, and many tasks accelerate discrete judgments and even more complex logical decisions [14]. The internal integrated circuit architecture of the GPU contains more cores than the CPU, and the temporary storage memory possessed by each core is small. The two processors have different applicable processing tasks due to their structural differences.

The CPU has excellent handling of long-term processing tasks with complex calculation steps such as mathematical calculations, data compression, physical simulation, and more complex logical operations. Due to their large number of cores, GPUs are suitable for parallel processing of the exact instructions on multiple cores. It is more appropriate to handle many repeated operations such as graphics operations, artificial intelligence model training, numerical analysis, and large amounts of data operations [15]. The heterogeneous computing structure composed of different computing units in a computing center can provide extensive rules and efficient computing services [16]. However, the additional processing tasks of various heterogeneous computing units, if the appropriate computing resource server type can be used for relevant tasks, it can effectively reduce the overall power consumption of the cloud computing center [17]. Resource-aware dynamic task scheduling approaches were developed in recent researches [18].

Cloud service providers establish high-performance data centers to meet user demands. Users prioritize response time. Task scheduling for user applications in cloud computing has

gained attention. Numerous heuristics have been proposed, but finding optimal scheduling remains difficult due to the NP-hard nature of the problem [19], [20], [21].

B. COMPUTING RESOURCES SIMULATION OF CLOUD COMPUTING CENTERS

Due to the low feasibility of using formal computing centers as a test, researchers often used simulations to model execution time, energy consumption, and temperature. These metrics are used in many applications in various situations and in multiple combinations. Simulating a computing center's computing environment avoids the high cost of configuring the actual test environment. Environmental simulations can also evaluate the feasibility of a mechanism that requires a significant investment; for example, they can be used to analyze the costs and benefits of increasing and managing solar power generation [22].

The least median squares regression was used by Zhang et al. to analyze CPU utilization data, reducing time complexity from the perspective of industry manufacturing [23]. RAFL proposed by Thakur et al. focus on the load balance in the cloud computing environment and try to minimize energy consumption in physical machines [24]. Related studies have also highlighted the importance of computing resource allocation in energy consumption efficiency [25], [26], [27].

Biran et al., proposed a solution for Cloud Federation in a simulated public cloud server. Cloud Federation quantifies the core allocation of the CPU and GPU to reduce computing center carbon emissions [28]. Bilal proposed a simulation research model for the Power-aware Job Scheduler (PAJS) used in high performance computing computer clusters to study how flexible adjustments to the threshold voltage can achieve energy efficiency and minimize the reaction time by adjusting the power supply voltage [29].

C. ALGORITHMS RELATED TO REDUCTION OF ENERGY CONSUMPTION

This study refers to the energy distribution mechanism used by researchers in the past. In terms of power management, Okamura and others have considered optimizing energy management solutions from the computer. They used the Markov decision process to simulate dynamic energy management problems and optimize the unit energy efficiency [13]. Recent study pointed out that most current approaches prioritize minimizing active physical machines but fail to adequately consider the simultaneous challenges of load fluctuation and energy efficiency in virtual machine provisions [30], [31].

Luo's research considered the concentration of a virtual computer based on the server's memory and computing resources. According to the needs of a work task on a cloud platform where the physical node is powered on, both horizontal expansion (number of nodes) and vertical expansion (distributed to the virtual machine computing resources) are used, and then the task is closed according to the computing requirements [11]. Research by Lu et al. simulates a computing power center whose primary power source is wind power generation. The paper does not consider the computing center's internal computing resource related issues but simulates the use of available wind energy to minimize energy costs [1]. Cheng et al. developed a cloud workload distribution and migration method called sCloud that can distribute cloud workloads to different computing centers according to their time varying renewable power availability [5].

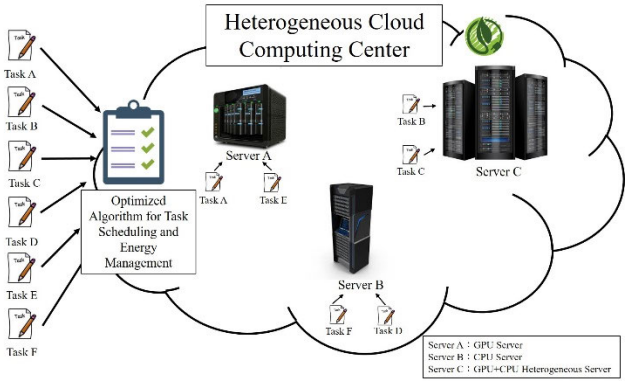

FIGURE 1. System architecture of heterogeneous cloud computing center.

One the other hand, the workflow scheduling is also attracted the interest of many studies to invested in enhancing scheduling performance in cloud computing through the allocation of time and resources [32], [33]. Take Sue & Xiong's research as an example; the proposed algorithm explores an agile response optimization model, considering task failure rate [34]. It investigates the probability density function of task request queue overflow and implements timeout requests to prevent network congestion [35].

The research mentioned above regarding energy allocation provides many relevant topics worthy of reference such as server power consumption, task priority, computing resources, and other factors. The method used to conduct energy allocation research in this research is described in detail. This study comprehensively considers the aspects of each test in the literature and extends them to construct an optimization algorithm.

III. RESEARCH METHOD

A. MATHEMATICAL MODEL

This study uses mathematical models to explore the optimal algorithm for heterogeneous computing resources, reducing energy consumption, and providing task scheduling and execution time in heterogeneous cloud computing centers. In a cloud computing center with a large amount of computing and various other resources, each job contains multiple tasks. The model calculates its optimal computing resource allocation and scheduling based on the sequence of tasks and energy consumption, and it expands these considerations into the optimization decisions for all computing work in the computing center. The system architecture is shown in FIGURE 1.

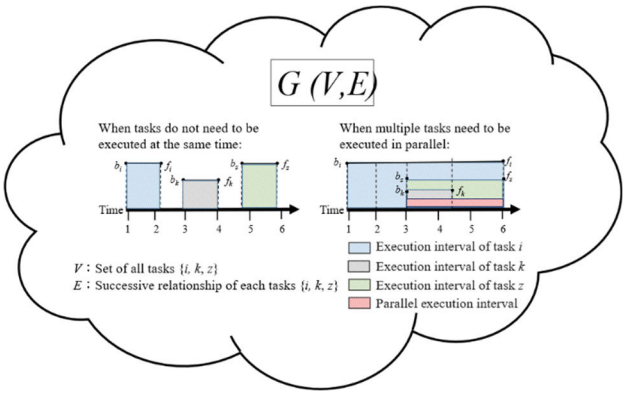

FIGURE 2. Relationship between tasks.

TABLE 1. Given parameters.

V	Set of all tasks in a job {1, 2, 3, ...,v}
M	Set of all heterogeneous computing VM servers (Virtual Machine, including CPU and GPU) resource types, {1,2,3,….m}
W_{ij}	The total execution time required when task $i \in V$ is assigned to VM $j \in M$
N_j	The sum of all tasks currently assigned to the VM $j \in M$ (upper limit)
U_{ij}	Power consumption rate if task $i \in V$ is assigned to VM $j \in M$
T	Deadline of the job
P	Maximum allowable power consumption rate
I_i	Set of tasks that are immediate ancestors of task $i \in V$
O_i	Set of tasks that are immediate descendants of task $i \in V$
d_{ij}	Set of the concurrent duplicates of task $i \in V$ that need to be executed simultaneously in parallel on the same type of VM $j \in M$
B	An arbitrarily large positive integer number

The model defines the relationship between different tasks in the predecessor graph $G\ (V,\ E)$, where V is the task set, and E is the sequence of the relationship between the tasks i, $k \in V$. The tasks i, k and z may be executed separately or in parallel, as shown as FIGURE 2.

The model targets both latency critical applications together with batch applications to minimize the processing energy consumption of tasks. The given parameters and decision variables were introduced in TABLE 1 and TABLE 2.

Where β_{ik}, γ_{ik} and δ_{ik} are defined to check whether other tasks k are also being executing at the starting time of each task i so that the maximum power consumption of the computing process can be calculated.

TABLE 2. Decision variables.

b_i	Beginning time of task $i \in V$
f_i	Finishing time of task $i \in V$
t_i	Execute time of task $i \in V$
p_i	Power allocated task $i \in V$, in watts
e_i	Total energy consumed when executing task $i \in V$
a_{ik}	1 if task $i \in V$ is executing at time b_i and 0 otherwise
β_{ik}	Defined as $b_i * b_k$, where $i,k \in V$
γ_{ik}	Defined as $b_i * f_k$, where $i,k \in V$
δ_{ik}	Defined as $a_{ik} * p_k$, where $i,k \in V$
x_{ij}	d_{ij} when task $i \in V$ is assigned to VM type $j \in M$, otherwise 0
θ_{ikj}	Defined as $a_{ik} * x_{kj}$, where $i,k \in V,\ j \in M$

The objective function is:

$$\min \sum_{i \in V} e_i \tag{IP}$$

The following constraints were developed to describe the mathematical model more precisely:

$$e_i = t_i \times p_i \quad \forall i \in V \tag{1.1}$$

$$t_i = \sum_{j \in M} x_{ij} W_{ij} \quad \forall i \in V \tag{1.2}$$

$$p_i = \sum_{j \in M} x_{ij} U_{ij} \quad \forall i \in V \tag{1.3}$$

$$t_i = f_i - b_i \quad \forall i \in V \tag{1.4}$$

$$\sum_{j \in M} x_{ij} \geq \min_j d_{ij} \quad \forall i \in V \tag{1.5}$$

$$t_i \geq 0 \quad \forall i \in V \tag{1.6}$$

$$f_i \geq 0 \quad \forall i \in V \tag{1.7}$$

$$b_i \geq 0 \quad \forall i \in V \tag{1.8}$$

$$b_i \geq f_k \quad \forall k \in I_i, i, k \in V, i \neq k \tag{1.9}$$

$$b_k \geq f_i \quad \forall k \in Q_i, i, k \in V, i \neq k \tag{1.10}$$

$$T \geq f_i \quad \forall i \in V \tag{1.11}$$

$$x_{ij} = d_{ij} \text{or} 0 \quad \forall i \in V \tag{1.12}$$

$$a_{ik} \geq \frac{\left[(b_k - f_k)^2 - (b_k - b_i)^2 - (f_k - b_i)^2 + \varepsilon\right]}{B} \quad \forall i, k \in V \tag{1.13}$$

$$(-2\gamma_{kk} + 2\beta_{ik} - 2\beta_{ii} + 2\gamma_{ik}) + \varepsilon \leq B a_{ik} \quad \forall i, k \in V \tag{1.14}$$

$$a_{ik} \in \{\varepsilon, 1\} \quad \forall i,\ k \in V \tag{1.15}$$

$$\sum_{k \in V} a_{ik} p_k \leq P \quad \forall i \in V \tag{1.16}$$

$$a_i k p_k = \delta_{ik} \quad \forall i,\ k \in V \tag{1.17}$$

$$b_i b_k = \beta_{ik} \quad \forall i,\ k \in V \tag{1.18}$$

$$b_i f_k = \gamma_{ik} \quad \forall i,\ k \in V \tag{1.19}$$

$$a_{ik} x_{kj} = \theta_{ikj} \quad \forall i,\ k \in V, j \in M \tag{1.20}$$

$$\sum_{k \in V} a_{ik} x_{kj} \leq N_j \quad \forall i \in V,\ j \in M. \tag{1.21}$$

The constraints are described as follows:

(1.1) The energy consumption of task i is multiplied by its execution time and power consumption.

(1.2) The total execution time of task i is the sum of the execution time on VM server j.

(1.3) The power consumed when task $i$is executed on VM server j.

(1.4) The task execution time is equal to the task end time minus the task start time.

(1.5) Any assigned task must be executed at least once and can be execute multiple time for backups.

(1.6) The execution time of task must be greater than or equal to 0.

(1.7) The end time of task must be greater than or equal to 0.

(1.8) The start time of task must be greater than or equal to 0.

(1.9) If task $k$is the immediate ancestor of task i, then the start time of task i must be later than the completion time of task k.

(1.10) If task i is an immediate descendant of task k, the start time of task i must be later than the completion time of task k.

(1.11) All tasks must be completed within the deadline.

1.12) The number of executions of the same task can be multiple backups. In addition, the following restriction is added to the model so that the total power consumption does not violate the upper limit P of the total power consumption of all types of available VM servers at any time. For each task i, the system checks the total power consumption at the beginning time of the task:

(1.13) If the square value minus the start time of task k and the start time of task i plus the square value minus the end time of task k and the start time of task i is less than or equal to the square value of the negative start and end times of task k, then the start time of i is between the start and end time of task k.

(1.14) This constraint is obtained by multiplying and simplifying the quadratic equation in (1.13).

(1.15) The introduction of (an extremely small number) that is replaced when its value can be set to 0 for LR-based reformulation.

(1.16) This constraint ensures that the system's total power consumption does not exceed the given upper limit P when the task is started.

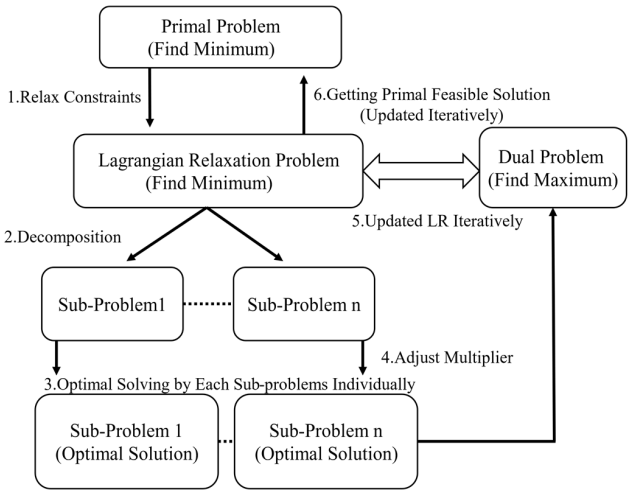

FIGURE 3. Flow chart of lagrangian relaxation.

(1.17), (1.18), (1.19) and (1.20) β_{ik}, γ_{ik}, δ_{ik} and θ_{ikj} subsidiary decision variable to make the constraints can be decompose in further optimization process.

(1.21) This constraint ensures that each type of server cannot exceed the maximum number of servers at any time.

IV. OPTIMIZATION SOLUTION APPROACH

The Lagrangian relaxation (LR) is chosen as the optimization method [36] for this study due to the high mathematical complexity characteristics of the developed mathematical model. The central concept of the LR method is that it relaxes the initially complicated constraints to its objective function, and each relaxation produces a corresponding Lagrangian Multiplier. The original problem is thus transformed into an LR problem. Next, the LR problem is divided into several subproblems. Each subproblem is decomposed using an optimization-based algorithm. Since the complex constraints are relaxed, the complexity and difficulty of the original problem are also relaxed. For a minimization problem, the LR problem is the lower bound of the original problem. This strategy narrows the gap between the original problem and the LR problem.

The resulting dual-model problem can help us easily observe the degree of optimization and has the following properties: (1) It minimizes the solution for the problem with a given set of non-negative Lagrange multipliers. (2) The optimal objective function value of the LR problem is the lower limit of the optimal solution to the original problem's objective function. The weak Lagrangian duality theorem can be compared with the feasible solution to the original problem we obtained as an evaluation index to derive the best solution. The flow of the LR process is shown in FIGURE. 3.

A. SIMULATION OF CLOUD COMPUTING CENTERS

To solve the complex problem regarding the multiplication of [0, 1] decision variables, this study input these variables into

the logarithm function, and then place these logarithms into the formula, the model is reformulated as following:

$$\min \sum_{i \in V} e_i \tag{IP2}$$

Subject to:

$$\log e_i = \log t_i + \log p_i \quad \forall i \in V \tag{2.1}$$
$$t_i = \sum_{j \in M} x_{ij} W_{ij} \quad \forall i \in V \tag{2.2}$$
$$p_i = \sum_{j \in M} x_{ij} U_{ij} \quad \forall i \in V \tag{2.3}$$
$$t_i = f_i - b_i \quad \forall i \in V \tag{2.4}$$
$$\min_j d_{ij} \le \sum_{j \in M} x_{ij} \quad \forall i \in V \tag{2.5}$$
$$\varepsilon \le t_i \quad \forall i \in V \tag{2.6}$$
$$\varepsilon \le f_i \quad \forall i \in V \tag{2.7}$$
$$\varepsilon \le b_i \quad \forall i \in V \tag{2.8}$$
$$f_k \le b_i \quad \forall k \in I_i, i, k \in V, i \ne k \tag{2.9}$$
$$f_i \le b_k \quad \forall k \in Q_i, i, k \in V, i \ne k \tag{2.10}$$
$$x_{ij} = d_{ij} \text{or} 0 \quad \forall i \in V \tag{2.11}$$
$$(-2\gamma_{kk} + 2\beta_{ik} - 2\beta_{ii} + 2\gamma_{ik}) + \varepsilon \le B a_{ik} \quad \forall i, k \in V \tag{2.12}$$
$$a_{ik} \in \{\varepsilon, 1\} \quad \forall i, k \in V \tag{2.13}$$
$$\sum_{k \in V} \delta_{ik} \le P \quad \forall i \in V \tag{2.14}$$
$$\log a_i k + \log p_k = \log \delta_{ik} \quad \forall i, k \in V \tag{2.15}$$
$$\log b_i + \log b_k = \log \beta_{ik} \quad \forall i, k \in V \tag{2.16}$$
$$\log b_i + \log f_k = \log \gamma_{ik} \quad \forall i, k \in V \tag{2.17}$$
$$\log a_{ik} + \log x_{kj} = \log \theta_{ikj} \quad \forall i, k \in V, j \in M \tag{2.18}$$
$$\sum_{k \in V} \theta_{ikj} \le N_j \quad \forall i \in V, j \in M. \tag{2.19}$$
$$\varepsilon^2 \le \delta_{ik} \quad \forall i, k \in V \tag{2.20}$$
$$\varepsilon^2 \le \beta_{ik} \quad \forall i, k \in V \tag{2.21}$$
$$\varepsilon^2 \le \gamma_{ik} \quad \forall i, k \in V \tag{2.22}$$
$$\theta_{ikj} \in \left\{\varepsilon^2, \varepsilon, 1\right\} \quad \forall i, k \in V, j \in M \tag{2.23}$$
$$x_{ij} \in \left\{\varepsilon, d_{ij}\right\} \quad \forall i \in V, j \in M \tag{2.24}$$
$$f_i \le T \quad \forall i \in V \tag{2.25}$$
$$b_i \le T \quad \forall i \in V \tag{2.26}$$
$$\varepsilon \le e_i \quad \forall i \in V \tag{2.27}$$

B. SOLUTION APPROACH FOR LR PROBLEM

After reformulation, this study substitute (2.1), (2.2), (2.3), (2.4), (2.5), (2.9), (2.10), (2.12), (2.14), (2.15), (2.16), (2.17), (2.18) and (2.19) into the relaxation to obtain the optimal solution:

$$\begin{aligned}
&\min \sum_{ieV} e_i \\
&+ \sum_{i \in V} \mu_i^1 (\log e_i - \log t_i - \log p_i) \\
&+ \sum_{i \in V} \mu_i^2 \left(\sum_{k \in V} \delta_{ik} - P \right) \\
&+ \sum_{i \in V} \sum_{j \in M} \mu_{ij}^3 \left(\sum_{k \in V} \theta_{ilj} - N_j \right) + \sum_{i \in V} \mu_i^4 \left(t_i - \sum_{j \in \mathbb{M}} x_{ij} W_{ij} \right) \\
&+ \sum_{i \in V} \mu_i^5 \left(p_i - \sum_{j \in M} x_{ij} U_{ij} \right) + \sum_{i \in V} \mu_i^6 \left(\min_j d_{ij} - \sum_{j \in M} x_{ij} \right) \\
&+ \sum_{ieV} \mu_i^7 (t_i - f_i + b_i) + \sum_{ieV} \mu_{ii}^8 [-2\gamma_{ii} - 2\beta_{ii}] \\
&+ \sum_{i \in V} \sum_{\substack{keV \\ i \ne k}} \mu_{ik}^8 2 (\gamma_{ik} + \beta_{ik}) + \sum_{i \in V} \sum_{k \in V} (\varepsilon - B a_{ik}) \mu_{ik}^8 \\
&+ \sum_{ieV} \sum_{k \in V} \mu_{ik}^9 (\log a_{ik} + \log p_k - \log \delta_{ik}) \\
&+ \sum_{i \in V} \sum_{k \in V} \mu_{ik}^{10} (\log b_i + \log b_k - \log \beta_{ik}) \\
&+ \sum_{i \in V} \sum_{k \in V} \mu_{ik}^{11} (\log b_i + \log f_k - \log \gamma_{ik}) \\
&+ \sum_{i \in V} \sum_{k \in V} \sum_{j \in M} \mu_{ikj}^{12} \left(\log a_{ik} + \log x_{kj} - \log \theta_{ikj}\right) \\
&+ \sum_{i \in V} \sum_{\substack{k \in V \\ k \in l_i \\ i \ne k}} \mu_{ik}^{13} (f_k - b_i) + \sum_{i \in V} \sum_{\substack{k \in V \\ k \in O_i \\ i \ne k}} \mu_{ik}^{14} (f_i - b_k)
\end{aligned} \tag{LR}$$

The LR problem can be decomposed into several independent subproblems with related decision variables. Following the development of the LR problem, it was broken down into solvable subproblems based on different decision variables. The decision variables were transformed by taking their logarithm, which allowed multiplication into addition. The range of feasible solutions was then incorporated into each subproblem. For instance, if the decision variable had a continuous range between 1 and ε, its original formula was differentiated once to obtain its extreme value. The resulting value and the two endpoints were input into the original subproblem to determine the minimum value. With knowledge of the range and concavity or convexity, each subproblem was easily solved using algorithms or heuristics. A comprehensive lower bound (LB) was developed to evaluate solution quality for the LR problems. Heuristic methods were designed to tune decision variables and fulfill all primal constraints to

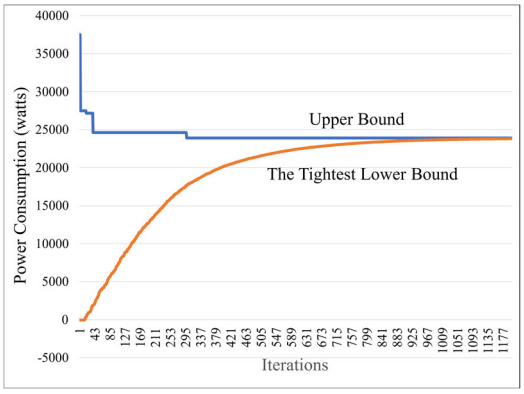

FIGURE 4. Obtaining the primal feasible solution and tightest lower bound.

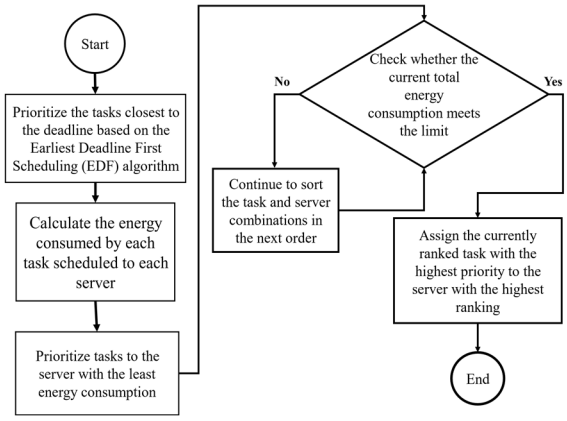

FIGURE 5. Flow chart of the proposed drop-and-add method.

obtain a feasible solution. The details of subproblems were not included in the article due to word count limitations.

C. DUAL PROBLEM AND THE PRIMAL FEASIBLE SOLUTION

The primal feasible solution developed by this study is to determine the optimal solution for the given problem of minimizing energy consumption. The algorithm iteratively approaches optimality by obtaining feasible solutions until it reaches the optimal solution. The algorithm's quality can be assessed by comparing the results of the primal feasible solution with those of the dual problem (LR results). After solving each subproblem, a set of decision variables was obtained and checked for feasibility. If the decision variables were feasible, an upper bound (UB) was calculated using the objective value of the primal problem. However, if the decision variables were not feasible, heuristic methods were utilized to adjust the decision variables and obtain feasible solutions. Determining the theoretical LB value based on the primal feasible solution requires selecting the key information from that feasible solution. The feasible region of the mathematical programming problem defined by the solution must meet all restrictions. The quality of the proposed algorithm is described by the target value gap between the algorithm and the LR problem (denoted by GAP):

$$\text{GAP} = \frac{\left|V_{theproposedmethod} - V_{LR}\right|}{\max\left|(V_{theproposed\ method}, V_{LR})\right|}$$

FIGURE 4 shows an experimental case in which this method was used. The blue line represents the process of iteratively obtaining the primal feasible solution. The goal is to determine the minimum value of primal problem. Then, the gradient descent method is used to determine the LB and iteratively obtain the closest LB. This process is represented by the orange line.

In this study, the resource management method used to obtain the primal feasible solution is named the Drop-and-Add algorithm. The process is shown in FIGURE 5, and it is based on the Earliest Deadline First (EDF) scheduling method and the energy consumption allocated to the server.

The best solutions are combinations with minor energy consumption. The initial solution uses First Come, First Served (FCFS) as a benchmark to evaluate the solution's quality.

V. COMPUTATIONAL EXPERIMENTS

This study simulates the actual operation mode of a cloud computing center while considering the lower bound, initial solution, and proposed Drop-and-Add method. In addition, to increase the experiment's effectiveness, this study also involves two cloud computing scheduling algorithms that are often used in practice. i.e., Round-Robin Scheduling (RR) [37] and Multilevel Queue Scheduling [38]. The performance evaluation uses several simulated cases to analyze performance.

A. DIFFERENT NUMBER OF TASKS

Different numbers of tasks may enter the cloud computing center at any time. This experiment verifies whether different numbers of tasks can be minimized when entering a cloud computing center with various computing resources. The proposed algorithm assigns tasks to the most suitable server in a way that consumes the least energy and does not exceed the execution deadline as shown in FIGURE 4 and TABLE 3.

Experimental observations reveal that an increase in the number of tasks directly corresponds to an increase in energy consumption. Initially, when the number of tasks remains small (less than 10), the energy consumption across all algorithms, except for the FCFS algorithm, exhibits negligible differences. This phenomenon can be attributed to the fixed computing resources available within the cloud computing center, which results in a non-linear growth pattern in energy consumption. However, as the number of tasks surpasses a certain threshold (more than 20), the energy consumption more than doubles. It is posited that this occurrence stems from a disparity between the number of tasks to be processed and the quantity of VM servers available. Consequently, this

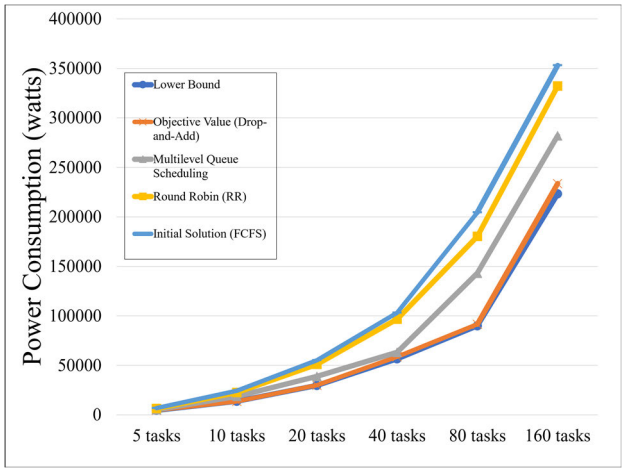

FIGURE 6. Energy consumption trends for different number of tasks.

TABLE 3. Energy consumption trends for different number of tasks.

	5 tasks	10 tasks	20 tasks	40 tasks	80 tasks
LB	4625.07	13661.55	29642.33	56586.5	89943.12
Drop-and-Add	4628.56	13714.33	29917.77	58608.29	91521.8
MQS	5233.43	18672.31	38790.13	63194.07	143125.3
RR	6433.02	22561.3	51003.11	96731.2	180284.3
FCFS	6847.28	24367.67	55127.9	103214.07	204523.5
GAP	0.07%	0.38%	0.92%	3.45%	1.72%

discrepancy impacts the accuracy of each algorithm in task scheduling and contributes to the overall energy consumption at all specific time points.

The proposed algorithm is capable for allocating tasks to VM servers with the lowest consumption within the computing center at any given time. By doing so, the computing center can exercise better control over energy consumption during peak and off-peak periods. It is believed that this proposed approach can enhance the overall efficiency of energy management within the computing center, thereby enabling effective utilization of resources during varying operational demands.

B. DIFFERENT TYPE OF VM SERVERS

A heterogeneous cloud computing center contains many different servers. Arranging tasks on various type of servers causes in different energy consumption and execution times. This study has completed all tasks within the deadline and minimized energy consumption. In other words, the proposed algorithm schedules tasks on a server type that can ensure completion before the deadline with the lowest energy consumption. In this experiment, we explore the difference between a single type of server, 10 different kinds (3 CPU servers, 3 GPU servers and 4 CPU+GPU servers) of servers

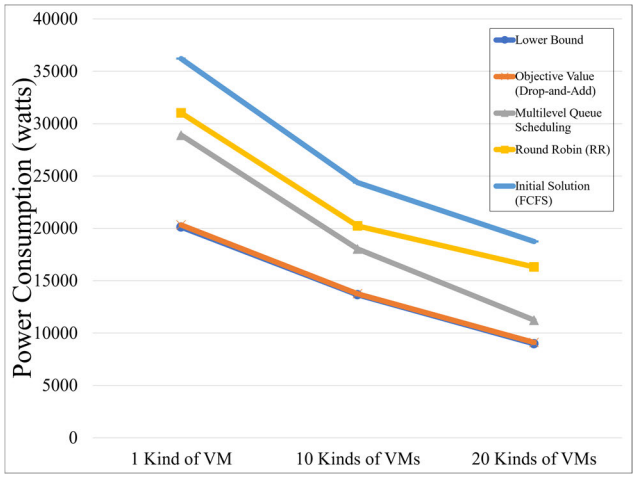

FIGURE 7. Energy consumption trends for different type of VM servers.

TABLE 4. Energy consumption trends for different type of VM servers.

	1 Kind of VM	10 Kinds of VMs	20 Kinds of VMs
LB	20123.4	13661.55	8976.3
Drop-and-Add	20332.1	13754.33	9103.2
MQS	28903.6	18032.4	11234.02
RR	31023.2	20234.5	16320.4
FCFS	36203.4	24367.67	18739.3
GAP	1.02%	0.67%	1.39%

and 20 different kinds (6 CPU servers, 6 GPU servers and 8 CPU+GPU servers) of servers. The experimental results are shown in FIGURE 7 and TABLE 4.

In the experiment examining different types of VM servers, a constant number of tasks was maintained. The obtained experimental findings corroborated the anticipated outcome, namely that when confronted with a fixed number of tasks to be processed, employing diverse types and quantities of servers enables the realization of task division and cooperation, leading to an overall reduction in energy consumption. The experimental results further demonstrated an exceedingly minimal GAP, providing compelling evidence that the algorithm introduced in this study relies on a comprehensive amalgamation of Lagrangian Multiplier and EDF parameters, thereby enabling the identification of the most accurate and optimal solution while adhering to the constraint of limiting the completion time for all recognition tasks.

C. DIFFERENT BACKUP TIMES

To ensure the security and integrity of computing tasks in a cloud computing center, this study innovatively considers the energy consumption related to task backup times. A decision variable d_{ij} is designed to represent a situation in which tasks are backed up. In the experiment, the same number of tasks were executed. During execution, the difference between no backups, random backups less than 5 times per job and ran-

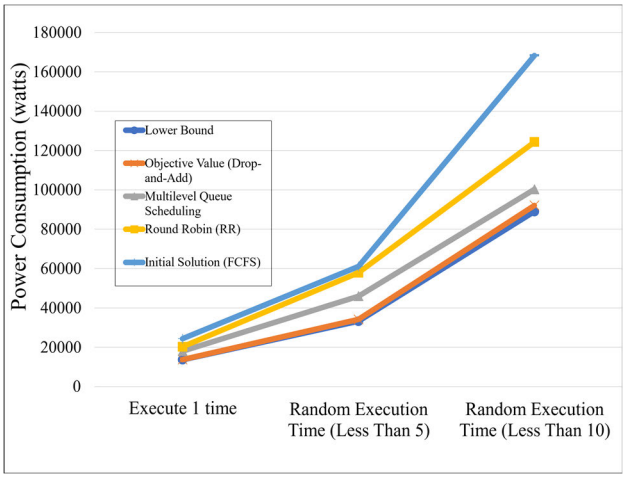

FIGURE 8. Energy Consumption Trends for Different Backup Times.

TABLE 5. Energy consumption trends for different backup times.

	Execute 1 time	Random Execution Time (less than 5)	Random Execution Time (less than 10)
LB	13661.55	33256.4	88983.21
Drop-and-Add	13754.33	34132.03	91112.31
MQS	18032.4	45980.02	100293.8
RR	20234.5	57893.5	124322.39
FCFS	24367.67	61092.32	168293.2
GAP	0.67%	2.57%	3.39%

dom backups less than 10 times is shown in FIGURE 8 and TABLE 5.

In order to explore the applicability of the experiment to practical scenarios, we conducted an investigation involving various backup time intervals. The outcomes of the experiment revealed noteworthy variations in results corresponding to the task quantity. Notably, as the number of task repetitions reached 10, all algorithms exhibited an escalation in energy consumption, attributable to the substantial surge in task volume. This observation further validates the superior performance of the algorithm proposed in this study, even under more demanding computational circumstances.

It can be found out that the proposed algorithm performs well regardless of the number of tasks, VM types, or backup times that meets the initial assumptions including computing resources and environment of the work. Specifically, the proposed Drop-and-Add algorithm with constraints that describe energy limitation at all points in time makes the task scheduling more efficiency. Also, the GAP between the algorithm proposed in this research and the LB is very small from the experimental data, which is sufficient to verify the algorithm's optimality.

VI. CONCLUSION

This study aimed to develop a resource allocation algorithm for cloud computing centers with heterogeneous computing resources, utilizing a mathematical model. The algorithm was specifically designed to address various factors that had been overlooked in prior research, including different types and quantities of computing equipment, multi-processor systems, and CPU and GPU utilization. By considering the energy consumption of the entire computing center, power consumption of individual devices, specific energy usage plans, all tasks handled by the comprehensive cloud computing center, and the temporal immediacy of tasks, the proposed algorithm sought to optimize resource allocation.

An LR-based algorithm was put forward, capable of identifying the optimal energy distribution combination upon completion of all computing tasks. The effectiveness of the proposed algorithm in addressing power consumption issues was revealed through experimental analysis. The minimal discrepancy observed between the upper and lower limits in the experimental data served as evidence supporting the high quality of the solution provided by the Drop-and-Add algorithm. Moreover, the resource allocation algorithm, determined using Lagrangian multipliers, demonstrated task prioritization and was compared against existing solutions based on experimental outcomes, exhibiting notable effectiveness and efficiency. The substantial impact of this study lies in its presentation of a novel perspective within the realm of cloud computing resource allocation research. Departing from the conventional approach of merely assigning tasks to servers with the minimum energy consumption required for task execution, our study explores the optimal allocation of computing resources across diverse application scenarios. This encompasses factors such as setting maximum energy consumption thresholds during peak computing periods and the imperative of meeting task deadlines. By considering these nuanced aspects, our research offers a heightened level of flexibility in devising comprehensive energy-saving strategies. In doing so, we contribute to the advancement of resource allocation methodologies in cloud computing, bringing forth innovative insights and paving the way for more effective and efficient energy management practices. Ultimately, the proposed resource allocation algorithm empowered service providers to make informed decisions and minimize energy consumption within their cloud computing centers.

The strengths of the optimization method presented in this study showcased potential applicability to computing tasks based on containers, as an alternative to VMs. Additionally, the proposed optimization framework could be readily expanded to encompass multiple jobs. The authors anticipate that the proposed optimization approach will greatly assist cloud service providers in achieving energy savings. However, the weaknesses would be the current mathematical model under consideration solely addresses resource allocation and energy consumption within a sin-

gular computing center, neglecting the potential for task transmission, collaboration, and interaction between distinct computing units. Such transmission processes encompass queuing, communication bandwidth, and data transfer, necessitating the exploration of distributed cloud computing systems that exhibit varying levels of task allocation across computational resources. Consequently, an expanded research scope is required to incorporate these multifaceted dynamics, enabling a more comprehensive understanding and analysis of resource allocation strategies within distributed cloud computing environments.

In future research endeavors, the authors aim to facilitate more synergistic cloud computing energy management across multiple devices, such as edge-to-edge, edge-to-cloud, or cloud-to-cloud configurations. Furthermore, they intend to consider the time and energy requirements associated with data transmission. The inclusion of a wider range of application types is recommended to enhance the scope and applicability of this research.

In summary, this study developed a resource allocation algorithm for cloud computing centers with heterogeneous computing resources. By considering various crucial factors and leveraging an LR-based algorithm, the proposed solution demonstrated effectiveness in mitigating power consumption concerns. The optimization method exhibited flexibility in accommodating different computing paradigms and the potential for extension to incorporate multiple jobs. The authors envision that the proposed algorithm will serve as a valuable tool for cloud service providers in minimizing energy consumption. Future research directions include enhancing cloud computing energy management across multiple devices, considering transmission time and energy, and expanding the scope of application types. of applications could be incorporated into the research.

REFERENCES

[1] X. Lu, D. Jiang, G. He, and H. Yu, "GreenBDT: Renewable-aware scheduling of bulk data transfers for geo-distributed sustainable datacenters," *Sustain. Comput., Informat. Syst.*, vol. 20, pp. 120–129, Dec. 2018.

[2] S. Vakilinia, "Energy efficient temporal load aware resource allocation in cloud computing datacenters," *J. Cloud Comput.*, vol. 7, pp. 1–24, Dec. 2018.

[3] E. W. Wächter, C. de Bellefroid, K. R. Basireddy, A. K. Singh, B. M. Al-Hashimi, and G. Merrett, "Predictive thermal management for energy-efficient execution of concurrent applications on heterogeneous multicores," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 6, pp. 1404–1415, Jun. 2019.

[4] Y. Li, X. Wang, P. Luo, and Q. Pan, "Thermal-aware hybrid workload management in a green datacenter towards renewable energy utilization," *Energies*, vol. 12, no. 8, p. 1494, Apr. 2019.

[5] D. Cheng, X. Zhou, Z. Ding, Y. Wang, and M. Ji, "Heterogeneity aware workload management in distributed sustainable datacenters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 2, pp. 375–387, Feb. 2019.

[6] C. de Alfonso, M. Caballer, A. Calatrava, G. Moltó, and I. Blanquer, "Multi-elastic datacenters: Auto-scaled virtual clusters on energy-aware physical infrastructures," *J. Grid Comput.*, vol. 17, no. 1, pp. 191–204, Mar. 2019.

[7] L. Zhang, K. Li, Y. Xu, J. Mei, F. Zhang, and K. Li, "Maximizing reliability with energy conservation for parallel task scheduling in a heterogeneous cluster," *Inf. Sci.*, vol. 319, pp. 113–131, Oct. 2015.

[8] L. Liu, H. Sun, C. Li, Y. Hu, T. Li, and N. Zheng, "Exploring customizable heterogeneous power distribution and management for datacenter," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 12, pp. 2798–2813, Dec. 2018.

[9] J. Yang, W. Xiao, C. Jiang, M. S. Hossain, G. Muhammad, and S. U. Amin, "AI-powered green cloud and data center," *IEEE Access*, vol. 7, pp. 4195–4203, 2019.

[10] G. Zhang, "Managing and scheduling approximate applications to utilize renewable energy in cloud computing datacenters," *Appl. Ecology Environ. Res.*, vol. 15, no. 3, pp. 307–321, 2017.

[11] R. F. Mansour, H. Alhumyani, S. A. Khalek, R. A. Saeed, and D. Gupta, "Design of cultural emperor penguin optimizer for energy-efficient resource scheduling in green cloud computing environment," *Cluster Comput.*, vol. 26, no. 1, pp. 575–586, Feb. 2023.

[12] P. Luo, X. Wang, H. Jin, Y. Li, and X. Yang, "Smart-grid-aware load regulation of multiple datacenters towards the variable generation of renewable energy," *Appl. Sci.*, vol. 9, no. 3, p. 518, Feb. 2019, doi: 10.3390/app9030518.

[13] B. Aksanli and J. Venkatesh, "Using datacenter simulation to evaluate green energy integration," *Computer*, vol. 45, no. 9, pp. 56–64, Sep. 2012.

[14] O. Beaumont, B. A. Becker, A. DeFlumere, L. Eyraud-Dubois, T. Lambert, and A. Lastovetsky, "Recent advances in matrix partitioning for parallel computing on heterogeneous platforms," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 1, pp. 218–229, Jan. 2019.

[15] Y. Chen, G. Xie, and R. Li, "Reducing energy consumption with cost budget using available budget preassignment in heterogeneous cloud computing systems," *IEEE Access*, vol. 6, pp. 20572–20583, 2018.

[16] A. Hazra, M. Adhikari, T. Amgoth, and S. N. Srirama, "A comprehensive survey on interoperability for IIoT: Taxonomy, standards, and future directions," *ACM Comput. Surv.*, vol. 55, no. 1, pp. 1–35, Jan. 2023.

[17] G. Xie, G. Zeng, R. Li, and K. Li, "Energy-aware processor merging algorithms for deadline constrained parallel applications in heterogeneous cloud computing," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 2, pp. 62–75, Apr. 2017.

[18] S. Nabi, M. Ibrahim, and J. M. Jimenez, "DRALBA: Dynamic and resource aware load balanced scheduling approach for cloud computing," *IEEE Access*, vol. 9, pp. 61283–61297, 2021.

[19] S. Omer, S. Azizi, M. Shojafar, and R. Tafazolli, "A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers," *J. Syst. Archit.*, vol. 115, May 2021, Art. no. 101996.

[20] R. Stewart, A. Raith, and O. Sinnen, "Optimising makespan and energy consumption in task scheduling for parallel systems," *Comput. Oper. Res.*, vol. 154, Jun. 2023, Art. no. 106212.

[21] B. Zhou and Y. Lei, "Bi-objective grey wolf optimization algorithm combined Levy flight mechanism for the FMC green scheduling problem," *Appl. Soft Comput.*, vol. 111, Nov. 2021, Art. no. 107717.

[22] G. Zhou, W. Tian, and R. Buyya, "Multi-search-routes-based methods for minimizing makespan of homogeneous and heterogeneous resources in cloud computing," *Future Gener. Comput. Syst.*, vol. 141, pp. 414–432, Apr. 2023.

[23] W. Zhang, R. Yadav, Y. Tian, S. K. S. Tyagi, I. A. Elgendy, and O. Kaiwartya, "Two-phase industrial manufacturing service management for energy efficiency of data centers," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7525–7536, Nov. 2022.

[24] A. Thakur and M. S. Goraya, "RAFL: A hybrid metaheuristic based resource allocation framework for load balancing in cloud computing environment," *Simul. Model. Pract. Theory*, vol. 116, Apr. 2022, Art. no. 102485.

[25] F. N. Al-Wesabi, M. Obayya, M. A. Hamza, J. S. Alzahrani, D. Gupta, and S. Kumar, "Energy aware resource optimization using unified metaheuristic optimization algorithm allocation for cloud computing environment," *Sustain. Comput., Informat. Syst.*, vol. 35, Sep. 2022, Art. no. 100686.

[26] F. A. Saif, R. Latip, Z. M. Hanapi, and K. Shafinah, "Multi-objective grey wolf optimizer algorithm for task scheduling in cloud-fog computing," *IEEE Access*, vol. 11, pp. 20635–20646, 2023.

[27] Y. Huang, H. Xu, H. Gao, X. Ma, and W. Hussain, "SSUR: An approach to optimizing virtual machine allocation strategy based on user requirements for cloud data center," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 670–681, Jun. 2021.

[28] Y. Biran, G. Collins, and J. Dubow, "Cloud computing cost and energy optimization through federated cloud SoS," *Syst. Eng.*, vol. 20, no. 3, pp. 280–293, May 2017.

[29] K. Bilal, A. Fayyaz, S. U. Khan, and S. Usman, "Power-aware resource allocation in computer clusters using dynamic threshold voltage scaling and dynamic voltage scaling: Comparison and analysis," *Cluster Comput.*, vol. 18, no. 2, pp. 865–888, Jun. 2015.
[30] Z. Zhou, M. Shojafar, M. Alazab, J. Abawajy, and F. Li, "AFED-EF: An energy-efficient VM allocation algorithm for IoT applications in a cloud data center," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 658–669, Jun. 2021.
[31] A. Mohammadzadeh, M. Masdari, and F. S. Gharehchopogh, "Energy and cost-aware workflow scheduling in cloud computing data centers using a multi-objective optimization algorithm," *J. Netw. Syst. Manag.*, vol. 29, no. 3, pp. 1–34, Jul. 2021.
[32] R. Mandal, M. K. Mondal, S. Banerjee, G. Srivastava, W. Alnumay, U. Ghosh, and U. Biswas, "MECpVmS: An SLA aware energy-efficient virtual machine selection policy for green cloud computing," *Cluster Comput.*, vol. 26, no. 1, pp. 651–665, Feb. 2023.
[33] L. Yan, H. Chen, Y. Tu, and X. Zhou, "A task offloading algorithm with cloud edge jointly load balance optimization based on deep reinforcement learning for unmanned surface vehicles," *IEEE Access*, vol. 10, pp. 16566–16576, 2022.
[34] W. Shu, K. Cai, and N. N. Xiong, "Research on strong agile response task scheduling optimization enhancement with optimal resource usage in green cloud computing," *Future Gener. Comput. Syst.*, vol. 124, pp. 12–20, Nov. 2021.
[35] S. Meng, L. Luo, X. Qiu, and Y. Dai, "Service-oriented reliability modeling and autonomous optimization of reliability for public cloud computing systems," *IEEE Trans. Rel.*, vol. 71, no. 2, pp. 527–538, Jun. 2022.
[36] M. L. Fisher, "An applications oriented guide to Lagrangian relaxation," *Interfaces*, vol. 15, no. 2, pp. 10–21, Apr. 1985.
[37] A. Singh, P. Goyal, and S. Batra, "An optimized round Robin scheduling algorithm for CPU scheduling," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 7, pp. 2383–2385, 2010.
[38] N. Harki, A. Ahmed, and L. Haji, "CPU scheduling techniques: A review on novel approaches strategy and performance assessment," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 2, pp. 48–55, May 2020.

KUANG-YEN TAI is currently pursuing the Ph.D. degree with the Department of Information Management, National Taiwan University, Taiwan. He is also a Lecturer with the Department of Computer Science, Tunghai University. His main research interests include parallel computing, artificial intelligence, cyber security, and software engineering.

FRANK YEONG-SUNG LIN received the B.S. degree in electrical engineering from the Electrical Engineering Department, National Taiwan University, in 1983, and the Ph.D. degree in electrical engineering from the Electrical Engineering Department, University of Southern California (USC), in 1991. After graduated from USC, he joined Telcordia Technologies (formerly Bell Communications Research, abbreviated as Bellcore), NJ, USA, where he was responsible for developing network planning and capacity/performance management algorithms. In 1994, he joined the Faculty of the Electronic Engineering, National Taiwan University of Science and Technology. Since 1996, he has been with the Faculty of the Information Management, National Taiwan University. His research interests include network optimization, network planning, network survivability, performance evaluation, high-speed networks, wireless networks, distributed algorithms, content-based information retrieval/filtering, biometrics, and network/information security.

CHIU-HAN HSIAO (Member, IEEE) received the Ph.D. degree in computer science from the Department of Information Management, National Taiwan University, Taiwan, in 2018. He is currently an Assistant Research Scientist with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan. His research interests include resource management of wireless communication (4G/5G), AI, and cloud computing technologies.