Received: 8 November 2021 | Revised: 30 June 2022 | Accepted: 23 July 2022
DOI: 10.1002/spe.3140

RESEARCH ARTICLE

WILEY

Workload management and server speed setting for cost-performance ratio optimization

Keqin Li

Department of Computer Science, State University of New York, New Paltz, New York, USA

Correspondence
Keqin Li, Department of Computer Science, State University of New York, New Paltz, NY 12561, USA.
Email: lik@newpaltz.edu

Abstract

The cost-performance tradeoff is a fundamental issue in a data center for cloud computing, which is closely related to two key metrics that both cloud consumers and service providers care the most, that is, quality of service and cost of service. While there are different definitions of quality of service, the average response time is a common choice of performance metric. While there are various considerations in cost of service, the average power consumption is a common choice of cost metric. Hence, the cost-performance tradeoff becomes the power-performance tradeoff. In this article, we deal with the power-performance tradeoff at the data center level. We study cost-performance ratio optimization by using the techniques of workload management and server speed setting. In particular, we make the following tangible contributions. We solve three optimization problems, that is, (1) *the workload management problem*—to find a workload distribution, such that the cost-performance ratio is minimized; (2) *the server speed setting problem*—to find a server speed setting, such that the cost-performance ratio is minimized; (3) *the workload management and server speed setting problem*—to find a workload distribution and a server speed setting, such that the cost-performance ratio is minimized. All the three optimization problems are analytically defined as multivariable optimization problems based on M/M/m queueing systems for multiple heterogeneous multiserver systems, together with two power consumption models, that is, the idle-speed model and the constant-speed model. Our approach makes it possible to quantitatively evaluate and optimize the cost-performance ratio of a data center within a rigorously developed framework. Each multivariable optimization problem is transformed to a nonlinear system of equations. Due to the sophistication of these equations, they are solved algorithmically by a numerical procedure. Furthermore, we provide approximate, accurate, and analytical solutions to the first two problems. Performance data are demonstrated for each problem, and the accuracy of our approximate solutions are also discussed. To the best of the author's knowledge, this is the first paper which analytically and algorithmically minimizes the cost-performance ratio of a data center with

Abbreviations: CPR, cost-performance ratio; QoS, quality of service; SLA, service level agreement.

multiple heterogeneous multiserver systems using the techniques of workload management and server speed setting.

KEYWORDS

cost-performance ratio, data center, power-performance tradeoff, server speed setting, workload management

1 INTRODUCTION

1.1 Background

The cost-performance tradeoff is a fundamental issue in a data center for cloud computing, which is closely related to two key metrics that both cloud consumers and service providers care the most. The first metric is quality of service.[1,2] A cloud consumer expects the highest quality of service, while a service provider attracts more customers by providing higher quality of service. The second metric is cost of service.[3,4] A cloud consumer expects the lowest cost/charge of service, while a service provider makes more profit by reducing the cost of service. However, attempting to simultaneously achieve the highest quality of service and the lowest cost of service yields two conflicting requirements. Therefore, it is a challenge to deal with the cost-performance tradeoff in cloud computing. While there are different definitions of quality of service,[5] the average response time is a common choice of performance metric. While there are various considerations in cost of service,[6] the average power consumption is a common choice of cost metric. Hence, the cost-performance tradeoff becomes the power-performance tradeoff.

The power-performance tradeoff can be dealt with at three different levels. (1) *Application level*—This is essentially power-aware and energy-efficient task scheduling, where an application is represented by a directed acyclic graph for tasks and their precedence constraints, which are scheduled on homogeneous or heterogeneous processors. The goal is to minimize the total execution time by consuming a given energy budget, or to minimize the total energy consumption by completing the tasks within a given performance bound.[7] (2) *Multiserver system level*—This is to consider a multiserver system with a stream of service requests. A multiserver system can be an inelastic multiserver system with fixed server size and speed, or a vertically/horizontally elastic and scalable multiserver system. A multiserver system is treated as a queueing model, typically an M/M/m model or its extensions and variations.[8] (3) *Data center level*—This is to consider multiple heterogeneous inelastic multiserver systems, and multiple heterogeneous vertically/horizontally elastic and scalable multiserver systems. A multiserver system can be treated as an M/M/m queueing model, and a single-server system can be treated as an M/M/1, an M/G/1, and a G/G/1 queueing model, where the queueing models can be extended to deal with elasticity and scalability.

The power-performance tradeoff can be studied from three different perspectives. (1) *Power constrained performance optimization*—This is to minimize the average response time, so that the average power consumption does not exceed certain power and cost constraint. (2) *Performance constrained power optimization*—This is to minimize the average power consumption, so that the average response time does not exceed certain performance and quality constraint. (3) *Cost-performance ratio optimization*—This is to minimize the cost-performance ratio, that is, the power-time product.

The power-performance tradeoff can be manipulated by using two effective techniques. (1) *Workload management*—Since the workload directly affects the average response time and the average power consumption, the power-performance tradeoff can be manipulated by distributing the workload in the optimal way. This is essentially load distribution and load balancing very effectively used in distributed computing, cluster computing, grid computing, cloud computing, and mobile edge computing. (2) *Server speed setting*—Since the server speed directly affects the average response time and the average power consumption, the power-performance tradeoff can be manipulated by setting the server speeds in an optimal way. This is essentially dynamic processor speed setting very widely employed in energy-efficient computing.

1.2 | New contributions

In this article, we deal with the power-performance tradeoff at the data center level. We study cost-performance ratio optimization by using the techniques of workload management and server speed setting. In particular, we make the following tangible contributions.

- We solve three optimization problems, that is, (1) *the workload management problem*—to find a workload distribution, such that the cost-performance ratio is minimized; (2) *the server speed setting problem*—to find a server speed setting, such that the cost-performance ratio is minimized; (3) *the workload management and server speed setting problem*—to find a workload distribution and a server speed setting, such that the cost-performance ratio is minimized.
- All the three optimization problems are analytically defined as multivariable optimization problems based on M/M/m queueing systems for multiple heterogeneous multiserver systems, together with two power consumption models, that is, the idle-speed model and the constant-speed model. Our approach makes it possible to quantitatively evaluate and optimize the cost-performance ratio of a data center within a rigorously developed framework.
- Each multivariable optimization problem is transformed to a nonlinear system of equations. Due to the sophistication of these equations, they are solved algorithmically by a numerical procedure. Furthermore, we provide approximate, accurate, and analytical solutions to the first two problems. Performance data are demonstrated for each problem, and the accuracy of our approximate solutions are also discussed.

To the best of the author's knowledge, this is the first paper which analytically and algorithmically minimizes the cost-performance ratio of a data center with multiple heterogeneous multiserver systems using the techniques of workload management and server speed setting.

Our problem formulation and solution based on queueing models are consistent with many existing studies of cost and performance optimization in cloud computing and edge computing (see Section 2). However, our cost-performance ratio optimization is novel and different from cost constrained performance optimization and performance constrained cost optimization in existing studies.

We would like to emphasize that the focus of the present study is to model, analyze, and optimize the cost-performance ratio of a data center using a theoretic approach. The methods and algorithms developed in this article are readily applicable to any data center as soon as all the parameters in our queueing system and power consumption models are available from the data center. The quality of our results depend only on the accuracy of the parameters from a real application environment. Note that such cost-performance ratio optimization is performed offline. It should be done when an application environment is changed, for example, when multiserver systems are added/removed or workload is increased/decreased.

We would like to mention two recent related research published on SPE. The first one is Reference 9, where the investigation in Reference 10 was extended to multiple classes of applications for power constrained performance optimization by using an optimal load distribution and an optimal server speed setting, the same techniques used in this article. However, performance constrained power optimization was not studied. The second one is Reference 11, which dealt with the cost-performance tradeoff (i.e., cost constrained performance optimization and performance constrained cost optimization) in mobile edge/cloud computing by server configuration optimization, where the M/M/m queueing model is used to characterize multiple heterogeneous edge servers. Note that the technique adopted is different from this article.

The rest of the article is organized as follows. In Section 2, we review related research. In Section 3, we present our multiserver model, power consumption models, and performance and cost measures. We also give several examples to motivate our investigation. In Sections 4–6, we address the three multivariable optimization problems respectively. We formally define each problem, develop a numerical algorithm to solve the problem, and demonstrate performance data. In Section 7, we conclude the article.

2 | RELATED RESEARCH

In this section, we review related research in analytical modeling and optimal handling of cost-performance tradeoff in two categories, that is, single multiserver system and multiple multiserver systems. We mainly focus on queueing model based approaches within the framework described in Section 1. Table 1 summarizes the related literature.

TABLE 1 Summary of related research

Reference	Environment	Model	Strategy	Cost-performance tradeoff
Single multiserver system				
12	Server farm	M/M/m	Server management	Minimizing energy-response time product
13	Cloud computing	M/M/m	Optimal server configuration	Minimizing response time with power consumption constraint; Minimizing power consumption with response time constraint
14	Cloud computing	Vertically elastic M/M/m	Workload dependent dynamic power management	Minimizing average response time with average power consumption constraint; Minimizing average power consumption with average response time constraint
15	Cloud computing	M/M/m	Optimal speed scheme	Minimizing cost-performance ratio
16	Cloud computing	M/G/1	Task type dependent server speed management	Power constrained performance optimization; Performance constrained power minimization
17	Cloud computing	Continuous-time Markov chain	Variable server size	Optimizing cost-performance ratio
Multiple Multiserver Systems				
10	Cloud computing	M/M/m	Optimal load distribution and optimal server speed setting	Power constrained performance optimization; Performance constrained power optimization
11	Mobile edge/cloud computing	M/M/m	Optimal server configuration	Cost constrained performance optimization; Performance constrained cost optimization
18	Mobile edge/cloud computing	M/M/m	Optimal server configuration and placement	Performance constrained cost minimization
19	Heterogeneous computing	M/G/1	Optimal load distribution and optimal server speed setting	Performance constrained power minimization (with dedicated tasks and general tasks)
20,21	Heterogeneous computing	M/M/1 with prioritization, preemption	Optimal load balancing and power allocation	Power constrained performance optimization; Performance constrained power optimization (with dedicated tasks and general tasks)
22	Data center	M/G/1	Optimal power allocation	Minimizing average task response time
23	Cloud computing	Vertically elastic M/M/m	Optimal task dispatching	Minimizing average response time; Minimizing average power consumption; Minimizing average cost-performance ratio
9	Cloud computing	M/M/m	Optimal load distribution and optimal server speed setting	Power constrained performance optimization (for multiple classes of applications)
24	Data center	G/G/1	Optimal server speed setting	Power constrained performance optimization; Performance constrained power optimization
25	Cloud computing	M/G/1	Optimal load distribution and optimal server speed scaling	Minimizing weighted sum of average response time and average power consumption
26	Data center	M/M/1	Resource provisioning adjustment and task allocation determination	Maximizing profit and optimizing average response time
27	Green computing	M/G/1	Power allocation	Performance constrained power minimization
This article	Cloud computing	M/M/m	Optimal workload management and optimal server speed setting	Minimizing cost-performance ratio

We would like to mention that the issue of cost-performance tradeoff has also been studied in various other related systems and environments with different techniques from diversified perspectives. Deng et al. investigated the tradeoff between power consumption and transmission delay in a fog-cloud computing system by finding an optimal workload allocation between fog and cloud to minimize power consumption with a service delay constraint.[28] Ding et al. proposed a Q-learning based task scheduling framework for optimizing average response time, server utilization, and energy consumption in cloud computing using the M/M/m queueing model.[29] Zhou et al. reduced the energy consumption of a data center while ensuring high quality of service (QoS) and minimizing service level agreement (SLA) violation rate.[30,31] The cost-performance tradeoff has also been considered by many researchers in the form of energy-latency tradeoff and energy-delay tradeoff for mobile edge computing.[32-36]

2.1 | Single multiserver system

Kong et al. observed that performance can improve as cost (i.e., the number of virtual machines, equivalent to the size of a multiserver system) increases; however, when the cost increases beyond certain level, the performance improves very slowly while the cost increases as usual, because the performance reaches its saturation point.[37] Such a phenomenon clearly implies that there is an optimal (i.e., minimum) value of the cost-performance ratio.

Gandhi et al. investigated management policies which minimize the energy-response time product (i.e., the power-time ^{2} product) for a server farm (i.e., a multiserver system), where each server can be in the state of *on*, *idle*, *sleep*, *off*, by using the M/M/m queueing model.[12] They found the optimal policy for a single-server system and a near-optimal policy for a multiserver system.

In Reference 13, the author considered the problem of power and performance management for a multicore server processor (treated as an M/M/m queueing system) in a cloud computing environment by optimal server configuration. It was shown that (1) for a given power consumption constraint, there is an optimal selection of server size and core speed, such that the minimum average response time can be achieved; (2) for a given task response time constraint, there is an optimal selection of server size and core speed, such that the minimum power consumption can be achieved.

In Reference 14, the author proposed the technique of using workload dependent dynamic power management (i.e., variable power and speed of a server according to the current workload) to improve system performance and to reduce energy consumption. This technique essentially creates a vertically elastic and scalable multiserver system with variable speed, which can be characterized by a variation of the standard M/M/m queueing model. It was shown that given certain average power consumption, there is an optimal speed scheme that minimizes the average response time, and that given certain average response time, there is an optimal speed scheme that minimizes the average power consumption. These are actually average response time optimization subject to power constraint and average power consumption optimization subject to performance constraint. In Reference 15, the author further found optimal single-speed schemes and double-speed schemes which minimize the cost-performance ratio. Actually, our effort in this article is to extend the study in Reference 15 from a single multiserver system to multiple multiserver systems.

In Reference 16, the author explored the technique of variable and task type dependent server speed management to optimize the server performance and to minimize the power consumption of a server with mixed applications. By establishing an M/G/1 queueing model for a server with variable and task type dependent speed, the problems of power constrained performance optimization and performance constrained power minimization were formulated and solved.

In Reference 17, the author developed a continuous-time Markov chain model (an extension of the M/M/m queueing model) for a horizontally elastic and scalable multiserver system with variable size, so that various performance and cost metrics can be obtained analytically and numerically, and the cost-performance ratio can be optimized. Using the results developed, a cloud service provider can predict its performance and cost guarantee and optimize its elastic scaling scheme to deliver the best cost-performance ratio, and a cloud consumer can compare cloud service providers and choose the best one.

2.2 | Multiple multiserver systems

In Reference 10, Cao et al. considered multiple heterogeneous inelastic multiserver systems (modeled as M/M/m queueing systems) across clouds and data centers, and solved the problems of power constrained performance optimization and performance constrained power optimization by using optimal power allocation (i.e., server speed setting) and load distribution. Our research in this article essentially is to minimize the cost-performance ratio within the same framework of Reference 10.

He et al. minimized operational expenditures while maintaining system performance at a predetermined level by optimal server configuration and suboptimal server placement in mobile edge computing, where edge servers were treated as M/G/m queueing systems.[18]

Huang et al. solved the problem of optimal distribution of general tasks among heterogeneous servers and optimal speed setting for the servers (treated as M/G/1 queueing systems), where each server has its own preloaded dedicated tasks and the servers have different queueing disciplines in scheduling dedicated tasks and general tasks, such that the average power consumption is minimized and that the average response time of general tasks does not exceed a given bound (i.e., performance constrained power minimization).[19] Huang et al. also minimized the average response time of generic tasks on heterogeneous embedded processors with dedicated tasks by optimal power allocation and load balancing, where the M/M/1 queueing model with prioritization and preemption was employed.[20,21]

In Reference 22, the author addressed power constrained performance optimization in a data center with multiple heterogeneous inelastic servers treated as M/G/1 queueing systems by optimal power allocation among multiple heterogeneous servers to minimize the average task response time. In Reference 23, a data center with multiple heterogeneous vertically elastic and scalable multiserver systems was considered. The author minimized the average task response time, the average power consumption, and the average cost-performance ratio by optimal task dispatching. In Reference 24, the author studied the problems of power constrained performance optimization and performance constrained power optimization in a data center with multiple heterogeneous and arbitrary servers treated as G/G/1 queueing systems through optimal server speed setting.

Tian et al. minimized a weighted sum of the average task response time and the average power consumption (i.e., the power-time sum), by optimal load distribution among multiple heterogeneous servers (treated as M/G/1 queueing systems) and optimal continuous and discrete service speed scaling.[25] Yang et al. employed a Stackelberg game, where a system monitor, who plays the role of the leader, can maximize profit by adjusting resource provisioning, whereas scheduler agents, who act as followers, can determine task allocation to obtain optimal average response time, and each server is modeled as an M/M/1 queuing system.[26] Zheng and Cai considered power allocation among servers in a server cluster with multiple classes of service requests to achieve satisfied service performance while still preserving energy efficiency, where each server is treated as an M/G/1 queueing system.[27]

3 | PRELIMINARY INFORMATION

In this section, we present our performance and cost measures. We also give several examples to motivate our study.

3.1 | Performance and cost measures

Our multiserver model and power consumption models are from Reference 10, where the reader can find detailed description. Table 2 provides a list of symbols and their definitions used in this article.

A cloud computing environment or data center serves users' service requests by using multiple heterogeneous multiserver systems. A data center maintains a pool of n heterogeneous multiserver systems $S_1, S_2, \dots, S_n$ with different sizes, speeds, power consumption models, workload, performance, and costs. A multiserver system S_i has m_i identical servers and is treated as an M/M/m queueing system. The average task response time of S_i is Reference 38

$$T_i = \bar{x}_i \left(1 + \frac{p_{i,m_i}}{m_i(1-\rho_i)^2} \right). \tag{1}$$

(Note: We use $\bar{y}$ to represent the expectation of a random variable y.)

Two categories of server speed and power consumption models are considered in this article. In the *idle-speed model*, we have

$$P_i = m_i(\rho_i \xi_i s_i^{\alpha_i} + P_i^*) = \lambda_i \bar{r} \xi_i s_i^{\alpha_i - 1} + m_i P_i^*. \tag{2}$$

TABLE 2 Symbols and definitions

Symbol	Definition
n	The number of heterogeneous multiserver systems
S_i	A multiserver system
m_i	The number of identical servers (i.e., the size) of S_i
λ_i	The arrival rate of the Poisson stream of service requests to S_i
λ	$= \lambda_1 + \lambda_2 + \cdots + \lambda_n$
r	Task execution requirement, an exponential random variable with mean $\bar{r}$
s_i	The identical execution speed of the servers of S_i
x_i	$= r/s_i$, task execution time on the servers of S_i, with mean $\bar{x}_i = \bar{r}/s_i$
μ_i	$= 1/\bar{x}_i = s_i/\bar{r}$, the average service rate of a server of S_i
ρ_i	$= \lambda_i/m_i\mu_i = \lambda_i\bar{x}_i/m_i = \lambda_i\bar{r}/m_i s_i$, the server utilization of S_i
$p_{i,k}$	The probability that there are k service requests in S_i
T_i	The average task response time of S_i
T	The overall average task response time of a data center
P_i	$= \xi_i s_i^{\alpha_i}$, dynamic power consumption of a server of S_i
P_i^*	Static power consumption of a server of S_i
P	The overall power consumption of a data center
R	$= PT$, cost-performance ratio

In the *constant-speed model*, we have

$$P_i = m_i(\xi_i s_i^{\alpha_i} + P_i^*). \tag{3}$$

The overall average task response time of a data center with n heterogeneous multiserver systems $S_1, S_2, \ldots, S_n$ is

$$T = \sum_{i=1}^{n} \frac{\lambda_i}{\lambda} T_i. \tag{4}$$

T is related to our performance measure. The overall power consumption of a data center with n heterogeneous multiserver systems $S_1, S_2, \ldots, S_n$ is

$$P = \sum_{i=1}^{n} P_i, \tag{5}$$

which is

$$P = \sum_{i=1}^{n} (\lambda_i \bar{r} \xi_i s_i^{\alpha_i - 1} + m_i P_i^*), \tag{6}$$

for the idle-speed model, and

$$P = \sum_{i=1}^{n} (m_i(\xi_i s_i^{\alpha_i} + P_i^*)), \tag{7}$$

for the constant-speed model. P is related to our cost measure.

Our performance measure is $1/T$, which is inversely proportional to the average task response time T, the higher, the better. The cost of cloud computing is determined by many different factors. Since the number n of multiserver systems and the sizes $m_1, m_2, \ldots, m_n$ of these multiserver systems are fixed in scale-up and scale-down auto-scaling schemes,[15] our cost measure is essentially the cost of power consumption P, the lower, the better. It is clear that the cost-performance ratio (CPR) refers to a data center's ability to deliver performance for certain cost. Generally speaking, data centers with lower CPR are more desirable, excluding other factors. In this article, we define CPR as cost/performance, that is, $R = PT$, that is, the power-time product.

3.2 Motivational examples

We provide a few illustrative examples to motivate our investigation.

Example 1. First, we given an example to illustrate optimal workload distribution. Consider two M/M/1 servers S_1 and S_2 with $m_1 = m_2 = 1$ and the constant-speed model. Then, we have

$$T_i = 1/(\mu_i - \lambda_i), \tag{8}$$

for $i = 1, 2$. Since P_i is independent of λ_i, minimizing $R = PT$ is equivalent to minimizing T, which is

$$T = \frac{1}{\lambda}\left(\frac{\lambda_1}{\mu_1 - \lambda_1} + \frac{\lambda_2}{\mu_2 - \lambda_2}\right). \tag{9}$$

Given certain workload λ, there is an optimal workload distribution (λ_1, λ_2) which minimizes T. Since $\lambda_2 = \lambda - \lambda_1$, we get

$$T = \frac{1}{\lambda}\left(\frac{\lambda_1}{\mu_1 - \lambda_1} + \frac{\lambda - \lambda_1}{(\mu_2 - \lambda) + \lambda_1}\right), \tag{10}$$

which is viewed as a function of λ_1. To minimize T, we need

$$\frac{\partial T}{\partial \lambda_1} = \frac{1}{\lambda}\left(\frac{\mu_1}{(\mu_1 - \lambda_1)^2} - \frac{\mu_2}{((\mu_2 - \lambda) + \lambda_1)^2}\right) = 0, \tag{11}$$

which gives rise to

$$\lambda_1 = \frac{\sqrt{\mu_1}\lambda + \sqrt{\mu_1\mu_2}(\sqrt{\mu_1} - \sqrt{\mu_2})}{\sqrt{\mu_1} + \sqrt{\mu_2}}, \tag{12}$$

and

$$\lambda_1 = \frac{\sqrt{\mu_2}\lambda + \sqrt{\mu_1\mu_2}(\sqrt{\mu_2} - \sqrt{\mu_1})}{\sqrt{\mu_1} + \sqrt{\mu_2}}. \tag{13}$$

Example 2. Next, we given an example to illustrate optimal server speed setting. Consider an M/M/1 server S_i with $m_i = 1$ and the idle-speed model. Then, we have

$$T_i = 1/(\mu_i - \lambda_i) = 1/(s_i/\bar{r} - \lambda_i), \tag{14}$$

and

$$P_i = \lambda_i \bar{r} \xi_i s_i^{\alpha_i - 1} + P_i^*. \tag{15}$$

Thus, we get

$$R_i = P_i T_i = \frac{\bar{r}(\lambda_i \bar{r} \xi_i s_i^{\alpha_i - 1} + P_i^*)}{s_i - \lambda_i \bar{r}}. \tag{16}$$

It is observed that T_i is a decreasing function of s_i, while P_i is an increasing function of s_i. Hence, there is an optimal s_i which minimizes R_i. If we view R_i as a function of s_i, then we need to have

$$\frac{\partial R_i}{\partial s_i} = \frac{\bar{r}(\lambda_i \bar{r} \xi_i (\alpha_i - 2) s_i^{\alpha_i - 1} - (\lambda_i \bar{r})^2 \xi_i (\alpha_i - 1) s_i^{\alpha_i - 2} - P_i^*)}{(s_i - \lambda_i \bar{r})^2} = 0. \tag{17}$$

When $\alpha_i = 3$, the above equation becomes

$$\lambda_i \bar{r} \xi_i s_i^2 - 2(\lambda_i \bar{r})^2 \xi_i s_i - P_i^* = 0, \tag{18}$$

which yields

$$s_i = \frac{(\lambda_i \bar{r})^2 \xi_i + \sqrt{(\lambda_i \bar{r})^4 \xi_i^2 + \lambda_i \bar{r} \xi_i P_i^*}}{\lambda_i \bar{r} \xi_i} = \lambda_i \bar{r} + \sqrt{(\lambda_i \bar{r})^2 + \frac{P_i^*}{\lambda_i \bar{r} \xi_i}}. \tag{19}$$

Since $s_i > 2\lambda_i \bar{r}$, we get $\rho_i = \lambda_i \bar{r}/s_i < 0.5$, which means that to minimize R_i, s_i should be large enough, so that server utilization is not high.

Example 3. Finally, we given an example to illustrate optimal workload distribution and server speed setting. Again, consider the two M/M/1 servers S_1 and S_2 with $m_1 = m_2 = 1$ and the constant-speed model. Since $\mu_1 = s_1/\bar{r}$ and $\mu_2 = s_2/\bar{r}$, from Example 1, we obtain

$$\lambda_1 = \frac{\sqrt{s_1}\lambda + \sqrt{s_1 s_2}/\bar{r}(\sqrt{s_1} - \sqrt{s_2})}{\sqrt{s_1} + \sqrt{s_2}}, \tag{20}$$

and

$$\lambda_2 = \frac{\sqrt{s_2}\lambda + \sqrt{s_1 s_2}/\bar{r}(\sqrt{s_2} - \sqrt{s_1})}{\sqrt{s_1} + \sqrt{s_2}}. \tag{21}$$

Therefore, T is viewed as a function of s_1 and s_2:

$$T = \frac{1}{\lambda}\left(\frac{\sqrt{s_1}\lambda + \sqrt{s_1 s_2}/\bar{r}(\sqrt{s_1} - \sqrt{s_2})}{(\sqrt{s_1} + \sqrt{s_2})s_1/\bar{r} - (\sqrt{s_1}\lambda + \sqrt{s_1 s_2}/\bar{r}(\sqrt{s_1} - \sqrt{s_2}))} + \frac{\sqrt{s_2}\lambda + \sqrt{s_1 s_2}/\bar{r}(\sqrt{s_2} - \sqrt{s_1})}{(\sqrt{s_1} + \sqrt{s_2})s_2/\bar{r} - (\sqrt{s_2}\lambda + \sqrt{s_1 s_2}/\bar{r}(\sqrt{s_2} - \sqrt{s_1}))} \right). \tag{22}$$

Furthermore,

$$P = P_1 + P_2 = \xi_1 s_1^{\alpha_1} + P_1^* + \xi_2 s_2^{\alpha_2} + P_2^* \tag{23}$$

is also a function of s_1 and s_2. To minimize $R = PT$, we need to consider $\partial R/\partial s_1 = 0$ and $\partial R/\partial s_2 = 0$. Unfortunately, the closed form solution is not available. However, such an optimal solution does exist, and there is an optimal workload distribution and server speed setting.

4 | WORKLOAD MANAGEMENT

In this section, we address the workload management problem.

4.1 | Problem formulation

Our optimization problem can be analytically defined as follows. Given certain workload specified by λ and $\bar{r}$, and n heterogeneous multiserver systems $S_1, S_2, \ldots, S_n$, where S_i is specified by $m_i, s_i, \xi_i, \alpha_i, P_i^*$, for all $1 \le i \le n$, find a workload distribution $(\lambda_1, \lambda_2, \dots, \lambda_n)$, such that the cost-performance ratio R is minimized, subject to the constraint that $\lambda_1 + \lambda_2 + \cdots + \lambda_n = \lambda$.

In the following (i.e., Equations (24)–(37)), we transform the above multivariable optimization problem to a nonlinear system of equations. We view $R(\lambda_1, \lambda_2, \dots, \lambda_n)$ as a function of $\lambda_1, \lambda_2, \dots, \lambda_n$. We can minimize $R(\lambda_1, \lambda_2, \dots, \lambda_n)$ subject to the constraint $C(\lambda_1, \lambda_2, \dots, \lambda_n) = \lambda_1 + \lambda_2 + \cdots + \lambda_n = \lambda$ by using the following Lagrange multiplier system,

$$\nabla R(\lambda_1, \lambda_2, \dots, \lambda_n) = \phi C(\lambda_1, \lambda_2, \dots, \lambda_n), \tag{24}$$

that is,

$$\frac{\partial R(\lambda_1, \lambda_2, \dots, \lambda_n)}{\partial \lambda_i} = \phi \frac{\partial C(\lambda_1, \lambda_2, \dots, \lambda_n)}{\partial \lambda_i} = \phi, \tag{25}$$

for all $1 \le i \le n$, where ϕ is a Lagrange multiplier.

For the constant-speed model, P is independent of $\lambda_1, \lambda_2, \dots, \lambda_n$. Therefore, we obtain

$$\frac{\partial R(\lambda_1, \lambda_2, \dots, \lambda_n)}{\partial \lambda_i} = \frac{1}{\lambda} P \left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i} \right), \tag{26}$$

for all $1 \le i \le n$.

For the idle-speed model, both P and T are dependent on $\lambda_1, \lambda_2, \dots, \lambda_n$. Let us rewrite $R = PT$ as

$$R = \frac{1}{\lambda} \left(\lambda_i P_i T_i + \left(\sum_{j \ne i} P_j \right) \lambda_i T_i + P_i \left(\sum_{j \ne i} \lambda_j T_j \right) + \left(\sum_{j \ne i} P_j \right) \left(\sum_{j \ne i} \lambda_j T_j \right) \right), \tag{27}$$

for all $1 \le i \le n$. Therefore, we obtain

$$\begin{aligned}
\frac{\partial R(\lambda_1, \lambda_2, \dots, \lambda_n)}{\partial \lambda_i} &= \frac{1}{\lambda} \left(P_i T_i + \lambda_i \overline{r} \xi_i s_i^{\alpha_i - 1} T_i + \lambda_i P_i \frac{\partial T_i}{\partial \lambda_i} + \left(\sum_{j \ne i} P_j)(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i} \right) + \overline{r} \xi_i s_i^{\alpha_i - 1} \left(\sum_{j \ne i} \lambda_j T_j \right) \right) \\
&= \frac{1}{\lambda} \left(\left(\sum_{j=1}^{n} P_j \right) \left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i} \right) + \overline{r} \xi_i s_i^{\alpha_i - 1} \left(\sum_{j=1}^{n} \lambda_j T_j \right) \right) \\
&= \frac{1}{\lambda} P \left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i} \right) + \overline{r} \xi_i s_i^{\alpha_i - 1} T,
\end{aligned} \tag{28}$$

for all $1 \le i \le n$.

Now, we derive $\partial T_i / \partial \lambda_i$. Recall that

$$T_i = \frac{\overline{r}}{s_i} \left(1 + \frac{p_{i,m_i}}{m_i (1 - \rho_i)^2} \right). \tag{29}$$

It is clear that

$$\frac{\partial T_i}{\partial \lambda_i} = \frac{\overline{r}}{m_i s_i} \left(\frac{2 p_{i,m_i}}{(1 - \rho_i)^3} \cdot \frac{\overline{r}}{m_i s_i} + \frac{1}{(1 - \rho_i)^2} \cdot \frac{\partial p_{i,m_i}}{\partial \lambda_i} \right), \tag{30}$$

where we notice that $\partial \rho_i / \partial \lambda_i = \overline{r} / m_i s_i$, for all $1 \le i \le n$. To further calculate $\partial T_i / \partial \lambda_i$, we need to examine $\partial p_{i,m_i} / \partial \lambda_i$. Recall that

$$p_{i,m_i} = \frac{m_i^{m_i}}{m_i!} \rho_i^{m_i} p_{i,0}. \tag{31}$$

Hence, we get

$$\frac{\partial p_{i,m_i}}{\partial \lambda_i} = \frac{m_i^{m_i}}{m_i!}\left(m_i \rho_i^{m_i-1} \frac{\bar{r}}{m_i s_i} p_{i,0} + \rho_i^{m_i} \frac{\partial p_{i,0}}{\partial \lambda_i} \right) = \frac{m_i^{m_i}}{m_i!} \rho_i^{m_i-1} \left(\frac{\bar{r}}{s_i} p_{i,0} + \rho_i \frac{\partial p_{i,0}}{\partial \lambda_i} \right), \tag{32}$$

for all $1 \leq i \leq n$. To further calculate $\partial p_{i,m_i}/\partial \lambda_i$, we need to examine $\partial p_{i,0}/\partial \lambda_i$. Recall that

$$p_{i,0} = \left(\sum_{k=0}^{m_i-1} \frac{m_i^k}{k!} \rho_i^k + \frac{m_i^{m_i}}{m_i!} \cdot \frac{\rho_i^{m_i}}{1-\rho_i} \right)^{-1}. \tag{33}$$

Thus, we have

$$\frac{\partial p_{i,0}}{\partial \lambda_i} = -p_{i,0}^2 \left(\sum_{k=1}^{m_i-1} \frac{m_i^{k-1}}{(k-1)!} \rho_i^{k-1} + \frac{m_i^{m_i-1}}{m_i!} \cdot \frac{m_i \rho_i^{m_i-1} - (m_i-1)\rho_i^{m_i}}{(1-\rho_i)^2} \right) \frac{\bar{r}}{s_i}, \tag{34}$$

for all $1 \leq i \leq n$.

An effective and efficient method is required to find $\lambda_1, \lambda_2, \ldots, \lambda_n$ and ϕ, which satisfy the equation $\partial R(\lambda_1, \lambda_2, \ldots, \lambda_n)/\partial \lambda_i = \phi$, for all $1 \leq i \leq n$, and $C(\lambda_1, \lambda_2, \ldots, \lambda_n) = \lambda$.

Therefore, we need to solve the following equation, that is,

$$\frac{1}{\lambda} P(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i}) + (\text{idle}) \bar{r} \xi_i s_i^{\alpha_i - 1} T = \phi, \tag{35}$$

where *idle* $= 1$ for the idle-speed model, and *idle* $= 0$ for the constant-speed model, or equivalently,

$$F_i = \frac{1}{\lambda} P(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i}) + (\text{idle}) \bar{r} \xi_i s_i^{\alpha_i - 1} T - \phi = 0, \tag{36}$$

for all $1 \leq i \leq n$. The above equations, together with

$$F_0 = \lambda_1 + \lambda_2 + \cdots + \lambda_n - \lambda = 0, \tag{37}$$

constitute a nonlinear system of $n+1$ equations with $n+1$ unknowns, that is, $\lambda_1, \lambda_2, \ldots, \lambda_n$, and ϕ.

An analytical solution to the above equations is infeasible. We take an algorithmic and numerical approach.

4.2 | A numerical algorithm

The following nonlinear system of equations needs to be solved:

$$\begin{cases} F_0(\phi, \lambda_1, \ldots, \lambda_n) = 0, \\ F_1(\phi, \lambda_1, \ldots, \lambda_n) = 0, \\ \quad \vdots \\ F_n(\phi, \lambda_1, \ldots, \lambda_n) = 0. \end{cases} \tag{38}$$

We represent the variables $\phi, \lambda_1, \ldots, \lambda_n$ using vector notation:

$$\mathbf{y} = (y_0, y_1, \ldots, y_n) = (\phi, \lambda_1, \ldots, \lambda_n), \tag{39}$$

and $F_i(\phi, \lambda_1, \ldots, \lambda_n) = F_i(y_0, y_1, \ldots, y_n) = F_i(\mathbf{y})$, where $F_i : \mathbb{R}^{n+1} \to \mathbb{R}$ maps $(n+1)$-dimensional space $\mathbb{R}^{n+1}$ into the real line $\mathbb{R}$. By defining a function $\mathbf{F} : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ which maps $\mathbb{R}^{n+1}$ into $\mathbb{R}^{n+1}$,

$$\mathbf{F}(\mathbf{y}) = (F_0(y_0, y_1, \ldots, y_n), F_1(y_0, y_1, \ldots, y_n), \ldots, F_n(y_0, y_1, \ldots, y_n)), \tag{40}$$

namely,

$$\mathbf{F}(\mathbf{y}) = (F_0(\mathbf{y}), F_1(\mathbf{y}), \ \dots \ , F_n(\mathbf{y})), \tag{41}$$

our nonlinear system of equations becomes

$$\mathbf{F}(\mathbf{y}) = \mathbf{0}, \tag{42}$$

where $\mathbf{0} = (0, 0, \ \dots \ , 0)$.

It is well known that we can solve the above nonlinear system of equations by using the standard Newton's method. For this purpose, we use the Jacobian matrix $J(\mathbf{y})$ defined as

$$J(\mathbf{y}) = \begin{bmatrix} \dfrac{\partial F_0(\mathbf{y})}{\partial y_0} & \dfrac{\partial F_0(\mathbf{y})}{\partial y_1} & \cdots & \dfrac{\partial F_0(\mathbf{y})}{\partial y_n} \\ \dfrac{\partial F_1(\mathbf{y})}{\partial y_0} & \dfrac{\partial F_1(\mathbf{y})}{\partial y_1} & \cdots & \dfrac{\partial F_1(\mathbf{y})}{\partial y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial F_n(\mathbf{y})}{\partial y_0} & \dfrac{\partial F_n(\mathbf{y})}{\partial y_1} & \cdots & \dfrac{\partial F_n(\mathbf{y})}{\partial y_n} \end{bmatrix}, \tag{43}$$

whose components are given in Equations (44)–(55). (These detailed derivations can be skipped without loss of continuity.)

For F_0, we have

$$\frac{\partial F_0(\mathbf{y})}{\partial y_0} = \frac{\partial F_0(\mathbf{y})}{\partial \phi} = 0, \tag{44}$$

and

$$\frac{\partial F_0(\mathbf{y})}{\partial y_j} = \frac{\partial F_0(\mathbf{y})}{\partial \lambda_j} = 1, \tag{45}$$

for all $1 \le j \le n$. For F_i, where $1 \le i \le n$, we have

$$\frac{\partial F_i(\mathbf{y})}{\partial y_0} = \frac{\partial F_i(\mathbf{y})}{\partial \phi} = -1. \tag{46}$$

Now, we examine $\partial F_i(\mathbf{y})/\partial y_i = \partial F_i(\mathbf{y})/\partial \lambda_i$, for all $1 \le i \le n$. For the idle-speed model,

$$F_i = \frac{1}{\lambda} P \left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i} \right) + \bar{r} \xi_i s_i^{\alpha_i - 1} T - \phi = 0, \tag{47}$$

where P is dependent on $\lambda_1, \lambda_2, \ \dots \ , \lambda_n$. Hence, we get

$$\begin{aligned} \frac{\partial F_i(\mathbf{y})}{\partial \lambda_i} &= \frac{1}{\lambda} \left(\bar{r} \xi_i s_i^{\alpha_i - 1} \left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i} \right) + P \left(2 \frac{\partial T_i}{\partial \lambda_i} + \lambda_i \frac{\partial^2 T_i}{\partial \lambda_i^2} \right) \right) + \frac{1}{\lambda} \bar{r} \xi_i s_i^{\alpha_i - 1} \left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i} \right) \\ &= \frac{1}{\lambda} \left(2 \bar{r} \xi_i s_i^{\alpha_i - 1} \left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i} \right) + P \left(2 \frac{\partial T_i}{\partial \lambda_i} + \lambda_i \frac{\partial^2 T_i}{\partial \lambda_i^2} \right) \right), \end{aligned} \tag{48}$$

where

$$\frac{\partial^2 T_i}{\partial \lambda_i^2} = \frac{\bar{r}}{m_i s_i} \left(\frac{6 p_{i,m_i}}{(1-\rho_i)^4} \left(\frac{\bar{r}}{m_i s_i} \right)^2 + \frac{4}{(1-\rho_i)^3} \cdot \frac{\bar{r}}{m_i s_i} \cdot \frac{\partial p_{i,m_i}}{\partial \lambda_i} + \frac{1}{(1-\rho_i)^2} \cdot \frac{\partial^2 p_{i,m_i}}{\partial \lambda_i^2} \right), \tag{49}$$

and

$$\frac{\partial^2 p_{i,m_i}}{\partial \lambda_i^2} = \frac{m_i^{m_i}}{m_i!}\left(\frac{m_i-1}{m_i}\left(\frac{\overline{r}}{s_i}\right)^2 \rho_i^{m_i-2} p_{i,0} + 2\frac{\overline{r}}{s_i}\rho_i^{m_i-1}\frac{\partial p_{i,0}}{\partial \lambda_i} + \rho_i^{m_i}\frac{\partial^2 p_{i,0}}{\partial \lambda_i^2}\right), \tag{50}$$

and

$$\begin{aligned}\frac{\partial^2 p_{i,0}}{\partial \lambda_i^2} = & -2p_{i,0}\frac{\partial p_{i,0}}{\partial \lambda_i}\left(\sum_{k=1}^{m_i-1}\frac{m_i^{k-1}}{(k-1)!}\rho_i^{k-1} + \frac{m_i^{m_i-1}}{m_i!}\cdot\frac{m_i\rho_i^{m_i-1}-(m_i-1)\rho_i^{m_i}}{(1-\rho_i)^2}\right)\frac{\overline{r}}{s_i} \\ & - p_{i,0}^2\left(\sum_{k=2}^{m_i-1}\frac{m_i^{k-1}}{(k-2)!}\rho_i^{k-2} + \frac{m_i^{m_i-1}}{m_i!}\cdot\frac{m_i(m_i-1)\rho_i^{m_i-2} - 2m_i(m_i-2)\rho_i^{m_i-1} + (m_i-2)(m_i-1)\rho_i^{m_i}}{(1-\rho_i)^3}\right)\left(\frac{\overline{r}}{s_i}\right)^2\frac{1}{m_i}. \end{aligned} \tag{51}$$

For the constant-speed model,

$$F_i = \frac{1}{\lambda}P\left(T_i + \lambda_i\frac{\partial T_i}{\partial \lambda_i}\right) - \phi = 0, \tag{52}$$

where P is independent of $\lambda_1, \lambda_2, \dots, \lambda_n$. Hence, we get

$$\frac{\partial F_i(\mathbf{y})}{\partial \lambda_i} = \frac{1}{\lambda}P\left(2\frac{\partial T_i}{\partial \lambda_i} + \lambda_i\frac{\partial^2 T_i}{\partial \lambda_i^2}\right). \tag{53}$$

Finally, we examine $\partial F_i(\mathbf{y})/\partial y_j = \partial F_i(\mathbf{y})/\partial \lambda_j$, for all $1 \le i \le n$ and $1 \le j \ne i \le n$. For the idle-speed model, we get

$$\frac{\partial F_i(\mathbf{y})}{\partial \lambda_j} = \frac{1}{\lambda}\left(\overline{r}\xi_j s_j^{\alpha_j-1}\left(T_i + \lambda_i\frac{\partial T_i}{\partial \lambda_i}\right) + \overline{r}\xi_i s_i^{\alpha_i-1}\left(T_j + \lambda_j\frac{\partial T_j}{\partial \lambda_j}\right)\right). \tag{54}$$

For the constant-speed model, we get

$$\frac{\partial F_i(\mathbf{y})}{\partial \lambda_j} = 0. \tag{55}$$

Algorithm 1 gives our numerical algorithm for finding an optimal workload distribution $(\lambda_1, \lambda_2, \dots, \lambda_n)$ and the Lagrange multiplier ϕ, that is, the vector $\mathbf{y} = (\phi, \lambda_1, \dots, \lambda_n)$ which satisfies the nonlinear system of equations $\mathbf{F}(\mathbf{y}) = \mathbf{0}$. Essentially, this is the standard Newton's iterative method[39(p. 451)]. Let

$$\lambda^* = \sum_{i=1}^{n}\frac{m_i s_i}{\overline{r}}, \tag{56}$$

which is the maximum workload that the n multiserver systems can handle collectively. A reasonable estimation of the λ_i's is that they are set in such a way that $\rho_1 = \rho_2 = \cdots = \rho_n = \rho = \lambda/\lambda^*$, that is,

$$\lambda_i = \rho\frac{m_i s_i}{\overline{r}} = \frac{\lambda}{\lambda^*}\cdot\frac{m_i s_i}{\overline{r}} = \left(\frac{m_i s_i}{\overline{r}}\Big/\sum_{i=1}^{n}\frac{m_i s_i}{\overline{r}}\right)\lambda, \tag{57}$$

for all $1 \le i \le n$. Our initial approximation of $\mathbf{y}$ is $\phi = 1$ and $\lambda_i = (\lambda/\lambda^*)(m_i s_i/\overline{r})$ for all $1 \le i \le n$ (line (1)). The value of $\mathbf{y}$ is repeatedly updated as $\mathbf{y} + \mathbf{z}$ (line (6)), where $\mathbf{z}$ is the solution to the linear system of equations $J(\mathbf{y})\mathbf{z} = -\mathbf{F}(\mathbf{y})$ (line (5)). Such update is iterated until $\|\mathbf{z}\| \le \epsilon$ (line (7)), where

$$\|\mathbf{z}\| = \sqrt{z_0^2 + z_1^2 + \cdots + z_n^2}, \tag{58}$$

and ϵ is a sufficiently small constant, say, 10^{-10}.

Algorithm 1. Optimal workload management

Input:Parameters λ, $\bar{r}$,m_i, s_i, ξ_i, α_i, P_i^*,for all $1 \le i \le n$.
Output:An optimal workload distribution and ϕ, that is,$\mathbf{y} = (\phi, \lambda_1, \ldots, \lambda_n)$,which satisfies $\mathbf{F}(\mathbf{y}) = \mathbf{0}$.

$\mathbf{y} \leftarrow (1, (\lambda/\lambda^*)(m_1 s_1/\bar{r}), \ldots, (\lambda/\lambda^*)(m_n s_n/\bar{r}))$; (1)
repeat (2)
  Calculate $J(\mathbf{y})$,where $J(\mathbf{y})_{i,j} = \partial F_i(\mathbf{y})/\partial y_j$for $0 \le i,j \le n$; (3)
  Calculate $\mathbf{F}(\mathbf{y}) = (F_0(\mathbf{y}), F_1(\mathbf{y}), \ldots, F_n(\mathbf{y}))$; (4)
  Solve the linear system of equations $J(\mathbf{y})\mathbf{z} = -\mathbf{F}(\mathbf{y})$; (5)
  $\mathbf{y} \leftarrow \mathbf{y} + \mathbf{z}$; (6)
until $\|\mathbf{z}\| \le \epsilon$. (7)

Since a matrix $J(\mathbf{y})$ is involved, the space complexity of Algorithm 1 is $O(n^2)$. During each repetition of the loop in lines (2)–(7), line (5) is the most time-consuming, which requires $O(n^2)$ time. The overall time complexity of Algorithm 1 is $O(Kn^2)$, where K is the number of repetitions and is determined by the required numerical accuracy ϵ.

For the idle-speed model, the solution to the linear system of equations in line (5) can be obtained by using the traditional algorithm of Gaussian elimination with backward substitution[39(pp. 268–269)]. For the constant-speed model, the Jacobian matrix $J(\mathbf{y})$ looks like

$$J(\mathbf{y}) = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ -1 & \dfrac{\partial F_1(\mathbf{y})}{\partial y_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & \cdots & \dfrac{\partial F_n(\mathbf{y})}{\partial y_n} \end{bmatrix}. \tag{59}$$

Therefore, we get $-z_0 + z_j \partial F_j(\mathbf{y})/\partial y_j = -F_j$, which implies that

$$z_j = \frac{z_0 - F_j}{\partial F_j(\mathbf{y})/\partial y_j}, \tag{60}$$

for all $1 \le j \le n$. Since $z_1 + z_2 + \cdots + z_n = -F_0$, we get

$$z_0 = \left(\sum_{j=1}^{n} \frac{F_j}{\partial F_j(\mathbf{y})/\partial y_j} - F_0 \right) \Bigg/ \left(\sum_{j=1}^{n} \frac{1}{\partial F_j(\mathbf{y})/\partial y_j} \right). \tag{61}$$

4.3 | Performance data

Consider $n = 3$ heterogeneous multiserver systems S_1, S_2, S_3, where the parameters of S_i are $m_i = 3 + i$, $s_i = 1.1 + 0.1i$, $\xi_i = 3.2 + 0.2i$, $\alpha_i = 3.4 - 0.1i$, and $P_i^* = 4.5 + 0.5i$, for all $1 \le i \le n$.

Let us assume that $\lambda = 18$, and the workload distribution has $\lambda_1 = 6 - d$, $\lambda_2 = 6$, $\lambda_3 = 6 + d$. In Figure 1, we display the cost-performance ratio R for $d = 1.3, 1.4, \ldots, 2.3$. It is clear that there is an optimal value of d which minimizes R. For the above set of d values, R is minimized as 438.349 and 462.436 for the idle-speed model and the constant-speed model respectively, when $d = 1.7$. However, this is certainly not the real minimum R.

In Tables 3 and 4, we show the optimal workload distribution $(\lambda_1, \lambda_2, \lambda_3)$, the corresponding server utilization (ρ_1, ρ_2, ρ_3), and the minimized cost-performance ratio R, for $\lambda = 15, 16, 17, 18, 19$. For instance, when $\lambda = 18$, R is minimized as 435.331 and 459.134 for the idle-speed model and the constant-speed model respectively.

As an approximate solution, we set $\lambda_i = (\lambda/\lambda^*)(m_i s_i/\bar{r})$, for all $1 \le i \le n$, such that all servers have the same utilization $\rho_i = \lambda/\lambda^*$. In Tables 3 and 4, we also show such workload distribution $(\lambda_1, \lambda_2, \lambda_3)$, the corresponding server utilization (ρ_1, ρ_2, ρ_3), the obtained cost-performance ratio R' and its relative error defined as $\Delta = (R' - R)/R$, for $\lambda = 15, 16, 17, 18, 19$. It is observed that the approximate solution is very close to the optimal solution.

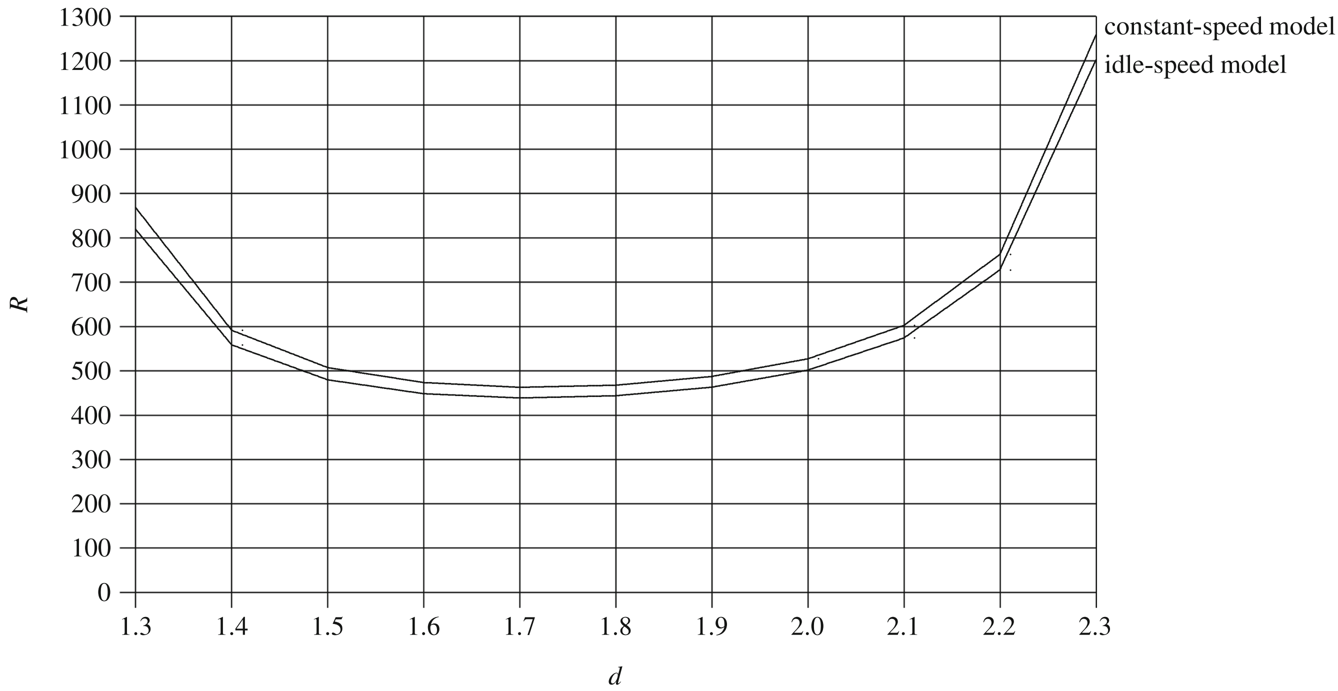

FIGURE 1 Cost-performance ratio R versus d

TABLE 3 Numerical data for optimal workload management (idle-speed model)

	Optimal				Approximate				
λ	λ_1, ρ_1	λ_2, ρ_2	λ_3, ρ_3	R	λ_1, ρ_1	λ_2, ρ_2	λ_3, ρ_3	R'	Δ
15	3.46799	4.93204	6.59997	194.957	3.65482	4.94924	6.39594	196.367	0.72318%
	0.72250	0.75878	0.78571		0.76142	0.76142	0.76142		
16	3.74781	5.26551	6.98668	232.140	3.89848	5.27919	6.82234	234.051	0.82320%
	0.78079	0.81008	0.83175		0.81218	0.81218	0.81218		
17	4.02980	5.59905	7.37114	296.327	4.14213	5.60914	7.24873	299.089	0.93208%
	0.83954	0.86139	0.87752		0.86294	0.86294	0.86294		
18	4.31380	5.93268	7.75352	435.331	4.38579	5.93909	7.67513	439.909	1.05151%
	0.89871	0.91272	0.92304		0.91371	0.91371	0.91371		
19	4.59941	6.26638	8.13421	970.118	4.62944	6.26904	8.10152	981.605	1.18401%
	0.95821	0.96406	0.96836		0.96447	0.96447	0.96447		

5 | SERVER SPEED SETTING

In this section, we address the server speed setting problem.

5.1 | Problem formulation

Our optimization problem can be analytically defined as follows. Given certain workload specified by λ and $\bar{r}$, and n heterogeneous multiserver systems $S_1, S_2, \ldots, S_n$, where S_i is specified by λ_i, m_i, ξ_i, α_i, P_i^*, for all $1 \le i \le n$, find a server speed setting $(s_1, s_2, \ldots, s_n)$, such that the cost-performance ratio R is minimized.

Notice that for each i, $1 \le i \le n$, there is s_i^* which minimizes $R_i = P_i T_i$. However, such a speed setting $(s_1^*, s_2^*, \ldots, s_n^*)$ does not necessarily minimize R.

TABLE 4 Numerical data for optimal workload management (constant-speed model)

	Optimal				Approximate				
λ	λ_1, ρ_1	λ_2, ρ_2	λ_3, ρ_3	R	λ_1, ρ_1	λ_2, ρ_2	λ_3, ρ_3	R'	Δ
15	3.45074	4.93070	6.61856	227.602	3.65482	4.94924	6.39594	229.883	1.00201%
	0.71890	0.75857	0.78792		0.76142	0.76142	0.76142		
16	3.73846	5.26479	6.99674	261.679	3.89848	5.27919	6.82234	264.396	1.03831%
	0.77885	0.80997	0.83295		0.81218	0.81218	0.81218		
17	4.02556	5.59874	7.37570	322.921	4.14213	5.60914	7.24873	326.428	1.08604%
	0.83866	0.86134	0.87806		0.86294	0.86294	0.86294		
18	4.31242	5.93258	7.75500	459.134	4.38579	5.93909	7.67513	464.398	1.14656%
	0.89842	0.91270	0.92321		0.91371	0.91371	0.91371		
19	4.59923	6.26636	8.13441	991.275	4.62944	6.26904	8.10152	1003.392	1.22233%
	0.95817	0.96406	0.96838		0.96447	0.96447	0.96447		

In the following (i.e., Equations (62)–(70)), we transform the above multivariable optimization problem to a non-linear system of equations. We view $R(s_1, s_2, \ldots, s_n)$ as a function of $s_1, s_2, \ldots, s_n$. We can minimize $R(s_1, s_2, \ldots, s_n)$ by considering

$$\frac{\partial R(s_1, s_2, \ldots, s_n)}{\partial s_i} = 0, \tag{62}$$

for all $1 \le i \le n$.

Let us rewrite $R = PT$ as

$$R = \frac{1}{\lambda}\left(\lambda_i P_i T_i + \lambda_i \left(\sum_{j \ne i} P_j\right) T_i + P_i \left(\sum_{j \ne i} \lambda_j T_j\right) + \left(\sum_{j \ne i} P_j\right)\left(\sum_{j \ne i} \lambda_j T_j\right)\right), \tag{63}$$

for all $1 \le i \le n$.

Therefore, we obtain

$$\begin{aligned} \frac{\partial R(s_1, s_2, \ldots, s_n)}{\partial s_i} &= \frac{1}{\lambda}\left(\lambda_i^2 \overline{r} \xi_i (\alpha_i - 1) s_i^{\alpha_i - 2} T_i + \lambda_i P_i \frac{\partial T_i}{\partial s_i} + \lambda_i \left(\sum_{j \ne i} P_j\right) \frac{\partial T_i}{\partial s_i} + \lambda_i \overline{r} \xi_i (\alpha_i - 1) s_i^{\alpha_i - 2} \left(\sum_{j \ne i} \lambda_j T_j\right)\right) \\ &= \frac{1}{\lambda}\left(\lambda_i \left(\sum_{j=1}^{n} P_j\right) \frac{\partial T_i}{\partial s_i} + \lambda_i \overline{r} \xi_i (\alpha_i - 1) s_i^{\alpha_i - 2} \left(\sum_{j=1}^{n} \lambda_j T_j\right)\right) \\ &= \frac{\lambda_i}{\lambda} P \frac{\partial T_i}{\partial s_i} + \lambda_i \overline{r} \xi_i (\alpha_i - 1) s_i^{\alpha_i - 2} T, \end{aligned} \tag{64}$$

for the idle-speed model, and

$$\begin{aligned} \frac{\partial R(s_1, s_2, \ldots, s_n)}{\partial s_i} &= \frac{1}{\lambda}\left(\lambda_i m_i \xi_i \alpha_i s_i^{\alpha_i - 1} T_i + \lambda_i P_i \frac{\partial T_i}{\partial s_i} + \lambda_i \left(\sum_{j \ne i} P_j\right) \frac{\partial T_i}{\partial s_i} + m_i \xi_i \alpha_i s_i^{\alpha_i - 1} \left(\sum_{j \ne i} \lambda_j T_j\right)\right) \\ &= \frac{1}{\lambda}\left(\lambda_i \left(\sum_{j=1}^{n} P_j\right) \frac{\partial T_i}{\partial s_i} + m_i \xi_i \alpha_i s_i^{\alpha_i - 1} \left(\sum_{j=1}^{n} \lambda_j T_j\right)\right) \\ &= \frac{\lambda_i}{\lambda} P \frac{\partial T_i}{\partial s_i} + m_i \xi_i \alpha_i s_i^{\alpha_i - 1} T, \end{aligned} \tag{65}$$

for the constant-speed model, for all $1 \le i \le n$. It is clear that

$$\frac{\partial T_i}{\partial s_i} = -\frac{T_i}{s_i} - \frac{\overline{r}}{m_i s_i}\left(\frac{2 p_{i,m_i}}{(1-\rho_i)^3} \cdot \frac{\rho_i}{s_i} - \frac{1}{(1-\rho_i)^2} \cdot \frac{\partial p_{i,m_i}}{\partial s_i}\right), \tag{66}$$

where we notice that $\partial \rho_i/\partial s_i = -\lambda_i \bar{r}/m_i s_i^2 = -\rho_i/s_i$, for all $1 \le i \le n$. To further calculate $\partial T_i/\partial s_i$, we need to examine $\partial p_{i,m_i}/\partial s_i$, which is

$$\frac{\partial p_{i,m_i}}{\partial s_i} = \frac{m_i^{m_i}}{m_i!}\rho_i^{m_i}\left(-\frac{m_i}{s_i}p_{i,0} + \frac{\partial p_{i,0}}{\partial s_i}\right), \tag{67}$$

for all $1 \le i \le n$. To further calculate $\partial p_{i,m_i}/\partial s_i$, we need to examine $\partial p_{i,0}/\partial s_i$, which is

$$\frac{\partial p_{i,0}}{\partial s_i} = p_{i,0}^2\left(\sum_{k=1}^{m_i-1}\frac{m_i^k}{(k-1)!}\rho_i^{k-1} + \frac{m_i^{m_i}}{m_i!}\cdot\frac{m_i\rho_i^{m_i-1} - (m_i-1)\rho_i^{m_i}}{(1-\rho_i)^2}\right)\frac{\rho_i}{s_i}, \tag{68}$$

for all $1 \le i \le n$.

Therefore, we need to solve the following nonlinear system of n equations, that is,

$$G_i = \frac{\lambda_i}{\lambda}P\frac{\partial T_i}{\partial s_i} + \lambda_i \bar{r}\xi_i(\alpha_i - 1)s_i^{\alpha_i-2}T = 0, \tag{69}$$

for the idle-speed model, and

$$G_i = \frac{\lambda_i}{\lambda}P\frac{\partial T_i}{\partial s_i} + m_i\xi_i\alpha_i s_i^{\alpha_i-1}T = 0, \tag{70}$$

for the constant-speed model, for all $1 \le i \le n$.

An analytical solution to the above equations is infeasible. We take an algorithmic and numerical approach.

5.2 | A numerical algorithm

We use the same method of Section 4.2. We have the following nonlinear system of equations, that is,

$$\begin{cases} G_1(s_1, s_2, \dots, s_n) = 0, \\ G_2(s_1, s_2, \dots, s_n) = 0, \\ \quad\vdots \\ G_n(s_1, s_2, \dots, s_n) = 0. \end{cases} \tag{71}$$

We represent the variables $s_1, s_2, \dots, s_n$ using vector notation:

$$\mathbf{y} = (y_1, y_2, \dots, y_n) = (s_1, s_2, \dots, s_n), \tag{72}$$

and $G_i(s_1, s_2, \dots, s_n) = G_i(y_1, y_2, \dots, y_n) = G_i(\mathbf{y})$, where $G_i : \mathbb{R}^n \to \mathbb{R}$ maps n-dimensional space $\mathbb{R}^n$ into the real line $\mathbb{R}$. By defining a function $\mathbf{G} : \mathbb{R}^n \to \mathbb{R}^n$ which maps $\mathbb{R}^n$ into $\mathbb{R}^n$,

$$\mathbf{G}(\mathbf{y}) = (G_1(y_1, y_2, \dots, y_n), G_2(y_1, y_2, \dots, y_n), \dots, G_n(y_1, y_2, \dots, y_n)), \tag{73}$$

namely,

$$\mathbf{G}(\mathbf{y}) = (G_1(\mathbf{y}), G_2(\mathbf{y}), \dots, G_n(\mathbf{y})), \tag{74}$$

our nonlinear system of equations becomes

$$\mathbf{G}(\mathbf{y}) = \mathbf{0}, \tag{75}$$

where $\mathbf{0} = (0, 0, \dots, 0)$.

Again, we can solve the above nonlinear system of equations by using Newton's method. For this purpose, we use the Jacobian matrix $J(\mathbf{y}) = (\partial G_i(\mathbf{y})/\partial y_j)_{n\times n}$, where $\partial G_i(\mathbf{y})/\partial y_j = \partial G_i(\mathbf{y})/\partial s_j$, $1 \le i,j \le n$, is calculated in Equations (76)–(82). (These detailed derivations can be skipped without loss of continuity.)

For the idle-speed model, we have

$$\frac{\partial G_i(\mathbf{y})}{\partial s_i} = \lambda_i \bar{r}\xi_i(\alpha_i-1)(\alpha_i-2)s_i^{\alpha_i-3}T + 2\frac{\lambda_i}{\lambda}\lambda_i\bar{r}\xi_i(\alpha_i-1)s_i^{\alpha_i-2}\frac{\partial T_i}{\partial s_i} + \frac{\lambda_i}{\lambda}P\frac{\partial^2 T_i}{\partial s_i^2}, \tag{76}$$

for all $1 \le i \le n$, and

$$\frac{\partial G_i(\mathbf{y})}{\partial s_j} = \lambda_j\bar{r}\xi_j(\alpha_j-1)s_j^{\alpha_j-2}\cdot\frac{\lambda_i}{\lambda}\cdot\frac{\partial T_i}{\partial s_i} + \lambda_i\bar{r}\xi_i(\alpha_i-1)s_i^{\alpha_i-2}\cdot\frac{\lambda_j}{\lambda}\cdot\frac{\partial T_j}{\partial s_j}, \tag{77}$$

for all $1 \le i \ne j \le n$.

For the constant-speed model, we have

$$\frac{\partial G_i(\mathbf{y})}{\partial s_i} = m_i\xi_i\alpha_i(\alpha_i-1)s_i^{\alpha_i-2}T + 2\frac{\lambda_i}{\lambda}m_i\xi_i\alpha_i s_i^{\alpha_i-1}\frac{\partial T_i}{\partial s_i} + \frac{\lambda_i}{\lambda}P\frac{\partial^2 T_i}{\partial s_i^2}, \tag{78}$$

for all $1 \le i \le n$, and

$$\frac{\partial G_i(\mathbf{y})}{\partial s_j} = m_j\xi_j\alpha_j s_j^{\alpha_j-1}\cdot\frac{\lambda_i}{\lambda}\cdot\frac{\partial T_i}{\partial s_i} + m_i\xi_i\alpha_i s_i^{\alpha_i-1}\cdot\frac{\lambda_j}{\lambda}\cdot\frac{\partial T_j}{\partial s_j}, \tag{79}$$

for all $1 \le i \ne j \le n$.

Furthermore, we have

$$\begin{aligned}
\frac{\partial^2 T_i}{\partial s_i^2} &= \frac{T_i}{s_i^2} - \frac{1}{s_i}\cdot\frac{\partial T_i}{\partial s_i} + \frac{\bar{r}}{m_i s_i^2}\left(\frac{2p_{i,m_i}}{(1-\rho_i)^3}\cdot\frac{\rho_i}{s_i} - \frac{1}{(1-\rho_i)^2}\cdot\frac{\partial p_{i,m_i}}{\partial s_i}\right)\\
&\quad - \frac{\bar{r}}{m_i s_i}\left(-\frac{2p_{i,m_i}}{(1-\rho_i)^3}\cdot\frac{\rho_i}{s_i^2} - \frac{2p_{i,m_i}}{(1-\rho_i)^3}\cdot\frac{\rho_i}{s_i^2} - \frac{6p_{i,m_i}}{(1-\rho_i)^4}\cdot\frac{\rho_i^2}{s_i^2} + \frac{2}{(1-\rho_i)^3}\cdot\frac{\rho_i}{s_i}\cdot\frac{\partial p_{i,m_i}}{\partial s_i}\right.\\
&\quad \left. + \frac{2}{(1-\rho_i)^3}\cdot\frac{\rho_i}{s_i}\cdot\frac{\partial p_{i,m_i}}{\partial s_i} - \frac{1}{(1-\rho_i)^2}\cdot\frac{\partial^2 p_{i,m_i}}{\partial s_i^2}\right)\\
&= \frac{T_i}{s_i^2} - \frac{1}{s_i}\cdot\frac{\partial T_i}{\partial s_i} + \frac{\bar{r}}{m_i s_i}\left(\frac{2\rho_i}{(1-\rho_i)^3 s_i^2}p_{i,m_i} - \frac{1}{(1-\rho_i)^2 s_i}\cdot\frac{\partial p_{i,m_i}}{\partial s_i}\right)\\
&\quad + \frac{\bar{r}}{m_i s_i}\left(\frac{2\rho_i(\rho_i+2)}{(1-\rho_i)^4 s_i^2}p_{i,m_i} - \frac{4\rho_i}{(1-\rho_i)^3 s_i}\cdot\frac{\partial p_{i,m_i}}{\partial s_i} + \frac{1}{(1-\rho_i)^2}\cdot\frac{\partial^2 p_{i,m_i}}{\partial s_i^2}\right)\\
&= \frac{T_i}{s_i^2} - \frac{1}{s_i}\cdot\frac{\partial T_i}{\partial s_i} + \frac{\bar{r}}{m_i s_i}\left(\frac{6\rho_i}{(1-\rho_i)^4 s_i^2}p_{i,m_i} - \frac{3\rho_i+1}{(1-\rho_i)^3 s_i}\cdot\frac{\partial p_{i,m_i}}{\partial s_i} + \frac{1}{(1-\rho_i)^2}\cdot\frac{\partial^2 p_{i,m_i}}{\partial s_i^2}\right),
\end{aligned} \tag{80}$$

and

$$\begin{aligned}
\frac{\partial^2 p_{i,m_i}}{\partial s_i^2} &= -\frac{m_i^{m_i}}{(m_i-1)!}\cdot\frac{\rho_i^{m_i}}{s_i}\left(-\frac{m_i}{s_i}p_{i,0} + \frac{\partial p_{i,0}}{\partial s_i}\right) + \frac{m_i^{m_i}}{m_i!}\rho_i^{m_i}\left(\frac{m_i}{s_i^2}p_{i,0} - \frac{m_i}{s_i}\cdot\frac{\partial p_{i,0}}{\partial s_i} + \frac{\partial^2 p_{i,0}}{\partial s_i^2}\right)\\
&= \frac{m_i^{m_i}}{m_i!}\rho_i^{m_i}\left(\frac{m_i^2}{s_i^2}p_{i,0} - \frac{m_i}{s_i}\cdot\frac{\partial p_{i,0}}{\partial s_i}\right) + \frac{m_i^{m_i}}{m_i!}\rho_i^{m_i}\left(\frac{m_i}{s_i^2}p_{i,0} - \frac{m_i}{s_i}\cdot\frac{\partial p_{i,0}}{\partial s_i} + \frac{\partial^2 p_{i,0}}{\partial s_i^2}\right)\\
&= \frac{m_i^{m_i}}{m_i!}\rho_i^{m_i}\left(\frac{m_i(m_i+1)}{s_i^2}p_{i,0} - 2\frac{m_i}{s_i}\cdot\frac{\partial p_{i,0}}{\partial s_i} + \frac{\partial^2 p_{i,0}}{\partial s_i^2}\right),
\end{aligned} \tag{81}$$

and

$$\begin{aligned}\frac{\partial^2 p_{i,0}}{\partial s_i^2} &= 2p_{i,0}\frac{\partial p_{i,0}}{\partial s_i}\cdot\frac{\rho_i}{s_i}\left(\sum_{k=1}^{m_i-1}\frac{m_i^k}{(k-1)!}\rho_i^{k-1}+\frac{m_i^{m_i}}{m_i!}\cdot\frac{m_i\rho_i^{m_i-1}-(m_i-1)\rho_i^{m_i}}{(1-\rho_i)^2}\right)\\
&\quad-2p_{i,0}^2\frac{\rho_i}{s_i^2}\left(\sum_{k=1}^{m_i-1}\frac{m_i^k}{(k-1)!}\rho_i^{k-1}+\frac{m_i^{m_i}}{m_i!}\cdot\frac{m_i\rho_i^{m_i-1}-(m_i-1)\rho_i^{m_i}}{(1-\rho_i)^2}\right)\\
&\quad-p_{i,0}^2\frac{\rho_i^2}{s_i^2}\left(\sum_{k=2}^{m_i-1}\frac{m_i^k}{(k-2)!}\rho_i^{k-2}+\frac{m_i^{m_i}}{m_i!}\cdot\frac{m_i(m_i-1)\rho_i^{m_i-2}-2m_i(m_i-2)\rho_i^{m_i-1}+(m_i-2)(m_i-1)\rho_i^{m_i}}{(1-\rho_i)^3}\right)\\
&=2p_{i,0}\frac{\partial p_{i,0}}{\partial s_i}\cdot\frac{\rho_i}{s_i}\left(\sum_{k=1}^{m_i-1}\frac{m_i^k}{(k-1)!}\rho_i^{k-1}+\frac{m_i^{m_i}}{m_i!}\cdot\frac{m_i\rho_i^{m_i-1}-(m_i-1)\rho_i^{m_i}}{(1-\rho_i)^2}\right)\\
&\quad-p_{i,0}^2\frac{\rho_i}{s_i^2}\left(\sum_{k=1}^{m_i-1}\frac{2m_i^k}{(k-1)!}\rho_i^{k-1}+\sum_{k=2}^{m_i-1}\frac{m_i^k}{(k-2)!}\rho_i^{k-1}+\frac{m_i^{m_i}}{m_i!}\cdot\frac{m_i(m_i+1)\rho_i^{m_i-1}-2(m_i^2-1)\rho_i^{m_i}+m_i(m_i-1)\rho_i^{m_i+1}}{(1-\rho_i)^3}\right),\end{aligned}\tag{82}$$

for all $1 \le i \le n$.

Algorithm 2 gives our numerical algorithm for finding an optimal server speed setting $(s_1, s_2, \ldots, s_n)$, that is, the vector $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ which satisfies the nonlinear system of equations $\mathbf{G}(\mathbf{y}) = \mathbf{0}$. Our initial approximation of $\mathbf{y}$ is $y_i = s_i = \lambda_i\bar{r}/m_i\rho$ for all $1 \le i \le n$ (line (1)), where ρ is a reasonably chosen utilization, for example, 0.7. The time and space complexities of Algorithm 2 are the same as those of Algorithm 1.

5.3 | Performance data

Let us consider the same heterogeneous multiserver systems S_1, S_2, S_3 in Section 4.3.

We assume that $\lambda = 18$, and the workload distribution has $\lambda_1 = 4.5$, $\lambda_2 = 6.0$, $\lambda_3 = 7.5$. The server speed setting has $s_i = \lambda_i\bar{r}/m_i\rho$ for all $1 \le i \le n$. In Figure 2, we display the cost-performance ratio R for ρ in the range (0,1). It is clear that there is an optimal value of ρ which minimizes R. For $\rho = 0.10, 0.15, 0.20, \ldots, 0.95$, R is minimized as 212.087 when $\rho = 0.65$ for the idle-speed model, and 280.646 when $\rho = 0.75$ for the constant-speed model respectively. However, this is certainly not the optimal choice of ρ, and not the real minimum R.

To find the optimal ρ, we rewrite T_i as

$$T_i = \frac{m_i}{\lambda_i}\rho\left(1+\frac{p_{i,m_i}}{m_i(1-\rho)^2}\right), \tag{83}$$

where

$$p_{i,m_i} = p_{i,0}\frac{(m_i\rho)^{m_i}}{m_i!}, \tag{84}$$

Algorithm 2. Optimal server speed setting

Input:Parameters λ, $\bar{r}$,λ_i, m_i, ξ_i, α_i, P_i^*,for all $1 \le i \le n$.
Output:An optimal server speed setting, that is,$\mathbf{y} = (s_1, s_2, \ldots, s_n)$,which satisfies $\mathbf{G}(\mathbf{y}) = \mathbf{0}$.

$\mathbf{y} \leftarrow (\lambda_1\bar{r}/m_1\rho, \lambda_2\bar{r}/m_2\rho, \ldots, \lambda_n\bar{r}/m_n\rho)$; (1)
repeat (2)
    Calculate $J(\mathbf{y})$,where $J(\mathbf{y})_{i,j} = \partial G_i(\mathbf{y})/\partial y_j$for $1 \le i,j \le n$; (3)
    Calculate $\mathbf{G}(\mathbf{y}) = (G_1(\mathbf{y}), G_2(\mathbf{y}), \ldots, G_n(\mathbf{y}))$; (4)
    Solve the linear system of equations $J(\mathbf{y})\mathbf{z} = -\mathbf{G}(\mathbf{y})$; (5)
    $\mathbf{y} \leftarrow \mathbf{y} + \mathbf{z}$; (6)
until $\|\mathbf{z}\| \le \varepsilon$. (7)

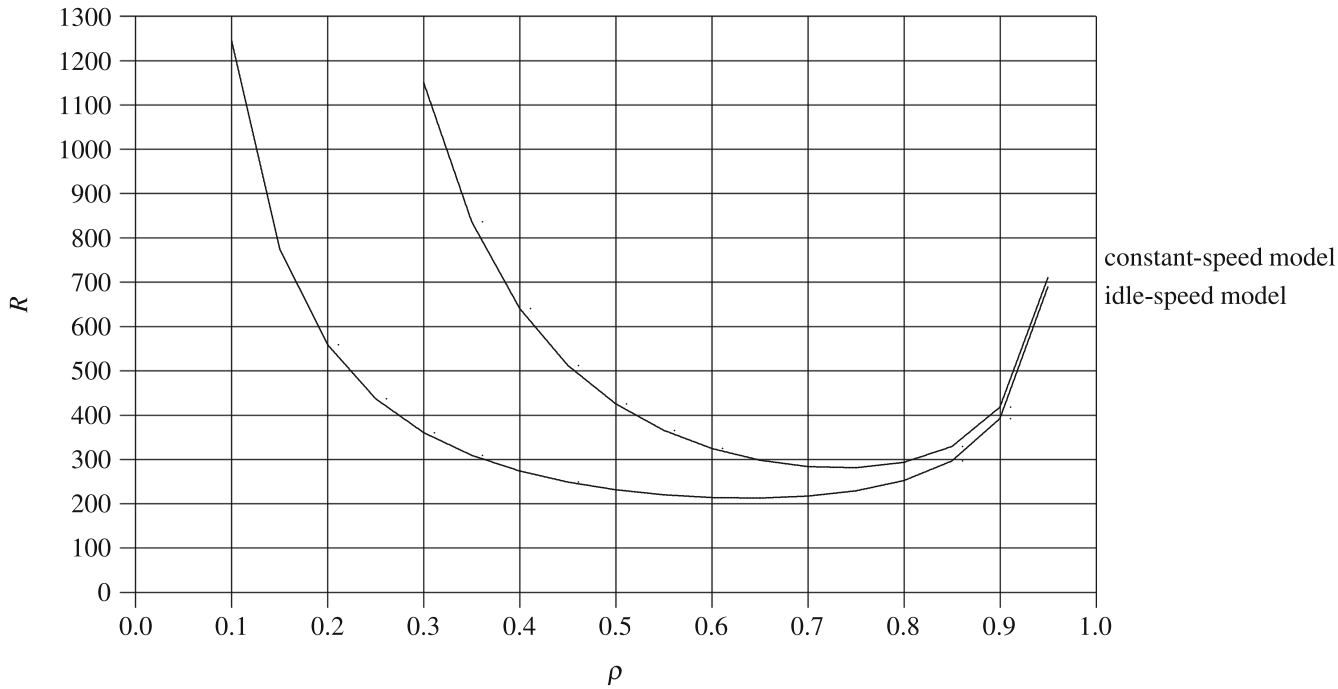

FIGURE 2 Cost-performance ratio R versus ρ.

and

$$p_{i,0} = \left(\sum_{k=0}^{m_i-1} \frac{(m_i\rho)^k}{k!} + \frac{(m_i\rho)^{m_i}}{m_i!} \cdot \frac{1}{1-\rho} \right)^{-1}, \tag{85}$$

for all $1 \le i \le n$. We can also rewrite P_i as

$$P_i = \frac{(\lambda_i \overline{r})^{\alpha_i}\xi_i}{m_i^{\alpha_i-1}} \cdot \frac{1}{\rho^{\alpha_i-1}} + m_i P_i^*, \tag{86}$$

for the idle-speed model, and

$$P_i = \frac{(\lambda_i \overline{r})^{\alpha_i}\xi_i}{m_i^{\alpha_i-1}} \cdot \frac{1}{\rho^{\alpha_i}} + m_i P_i^*, \tag{87}$$

for the constant-speed model, for all $1 \le i \le n$. The cost-performance ratio R is viewed as a function of ρ:

$$R = PT = \left(\sum_{i=1}^{n} P_i \right) \left(\sum_{i=1}^{n} \frac{\lambda_i}{\lambda} T_i \right). \tag{88}$$

Hence, we need to find ρ such that $\partial R/\partial \rho = 0$ (which is an increasing function of ρ), where

$$\frac{\partial R}{\partial \rho} = \left(\sum_{i=1}^{n} \frac{\partial P_i}{\partial \rho} \right) T + P \left(\sum_{i=1}^{n} \frac{\lambda_i}{\lambda} \cdot \frac{\partial T_i}{\partial \rho} \right). \tag{89}$$

We have

$$\frac{\partial P_i}{\partial \rho} = -\frac{(\alpha_i - 1)(\lambda_i \overline{r})^{\alpha_i}\xi_i}{m_i^{\alpha_i-1}} \cdot \frac{1}{\rho^{\alpha_i}}, \tag{90}$$

for the idle-speed model, and

$$\frac{\partial P_i}{\partial \rho} = -\frac{\alpha_i(\lambda_i\bar{r})^{\alpha_i}\xi_i}{m_i^{\alpha_i-1}} \cdot \frac{1}{\rho^{\alpha_i+1}}, \tag{91}$$

for the constant-speed model, for all $1 \le i \le n$. Furthermore, we have

$$\begin{aligned}\frac{\partial T_i}{\partial \rho} &= \frac{m_i}{\lambda_i}\left(1 + \frac{p_{i,m_i}}{m_i(1-\rho)^2} + \frac{\rho}{m_i}\left(\frac{2p_{i,m_i}}{(1-\rho)^3} + \frac{1}{(1-\rho)^2}\cdot\frac{\partial p_{i,m_i}}{\partial \rho}\right)\right)\\ &= \frac{1}{\lambda_i}\left(m_i + \frac{1+\rho}{(1-\rho)^3}p_{i,m_i} + \frac{\rho}{(1-\rho)^2}\cdot\frac{\partial p_{i,m_i}}{\partial \rho}\right),\end{aligned} \tag{92}$$

where

$$\frac{\partial p_{i,m_i}}{\partial \rho} = \frac{m_i^{m_i}}{m_i!}\left(m_i\rho^{m_i-1}p_{i,0} + \rho^{m_i}\frac{\partial p_{i,0}}{\partial \rho}\right), \tag{93}$$

and

$$\frac{\partial p_{i,0}}{\partial \rho} = -p_{i,0}^2\left(\sum_{k=1}^{m_i-1}\frac{m_i^k}{(k-1)!}\rho^{k-1} + \frac{m_i^{m_i}}{m_i!}\cdot\frac{m_i\rho^{m_i-1}-(m_i-1)\rho^{m_i}}{(1-\rho)^2}\right), \tag{94}$$

for all $1 \le i \le n$. It is clear that ρ can be found by using the classic bisection method[39(p. 21)].

Let $\lambda_i = q + 1.5i$, for all $1 \le i \le n$. In Tables 5 and 6, we show the optimal server speed setting (s_1, s_2, s_3), the corresponding server utilization (ρ_1, ρ_2, ρ_3), and the minimized cost-performance ratio R, for $q = 1, 2, 3, 4, 5$ and $\lambda = 12, 15, 18, 21, 24$. For instance, when $q = 3$ and $\lambda = 18$, R is minimized as 210.640 and 277.722 for the idle-speed model and the constant-speed model respectively. As we know from Example 2 in Section 3.2, the ρ_i's cannot be too high.

As an approximate solution, we set $s_i = \lambda_i\bar{r}/m_i\rho$, for all $1 \le i \le n$, such that all servers have the same utilization ρ. The value of ρ is determined in such a way that R is minimized. In Tables 5 and 6, we also show such server speed setting (s_1, s_2, s_3), the corresponding server utilization (ρ_1, ρ_2, ρ_3), the obtained cost-performance ratio R' and its relative error defined as $\Delta = (R' - R)/R$, for $\lambda = 12, 15, 18, 21, 24$. It is observed that the approximate solution is very close to the optimal solution. By the way, when $q = 3$ and $\lambda = 18$, the optimal choice of ρ is 0.63634 and 0.73533, which result in R to be 211.886 and 280.067 for the idle-speed model and the constant-speed model respectively.

TABLE 5 Numerical data for optimal server speed setting (idle-speed model)

	Optimal				**Approximate**				
λ	s_1, ρ_1	s_2, ρ_2	s_3, ρ_3	R	s_1, ρ_1	s_2, ρ_2	s_3, ρ_3	R'	Δ
12	1.49392	1.54997	1.59551	134.874	1.19836	1.53390	1.75759	139.904	3.72993%
	0.41836	0.51614	0.57453		0.52155	0.52155	0.52155		
15	1.64006	1.68548	1.71913	167.118	1.47654	1.68747	1.82809	169.728	1.56161%
	0.53352	0.59330	0.63017		0.59260	0.59260	0.59260		
18	1.86142	1.87957	1.89354	210.640	1.76791	1.88578	1.96435	211.886	0.59189%
	0.60438	0.63844	0.66014		0.63634	0.63634	0.63634		
21	2.12173	2.10520	2.09631	266.823	2.07328	2.11098	2.13611	267.259	0.16311%
	0.64805	0.66502	0.67579		0.66320	0.66320	0.66320		
24	2.40403	2.34961	2.31673	336.395	2.38964	2.35288	2.32837	336.439	0.01310%
	0.67595	0.68096	0.68343		0.68002	0.68002	0.68002		

TABLE 6 Numerical data for optimal server speed setting (constant-speed model)

λ	Optimal s_1, ρ_1	s_2, ρ_2	s_3, ρ_3	R	Approximate s_1, ρ_1	s_2, ρ_2	s_3, ρ_3	R'	Δ
12	1.10853	1.22475	1.30153	180.102	0.94749	1.21278	1.38965	187.553	4.13713%
	0.56381	0.65320	0.70430		0.65964	0.65964	0.65964		
15	1.34071	1.41787	1.46826	220.825	1.23888	1.41586	1.53385	225.101	1.93624%
	0.65264	0.70528	0.73784		0.70628	0.70628	0.70628		
18	1.59547	1.63054	1.65259	277.722	1.52993	1.63193	1.69993	280.067	0.84421%
	0.70512	0.73595	0.75639		0.73533	0.73533	0.73533		
21	1.86499	1.85622	1.84897	351.837	1.82511	1.85829	1.88041	352.913	0.30573%
	0.73727	0.75422	0.76619		0.75338	0.75338	0.75338		
24	2.14427	2.09072	2.05377	443.848	2.12480	2.09211	2.07032	444.151	0.06836%
	0.75784	0.76529	0.77094		0.76478	0.76478	0.76478		

6 WORKLOAD MANAGEMENT AND SERVER SPEED SETTING

In this section, we address the problem of workload management and server speed setting.

6.1 Problem formulation

Our optimization problem can be analytically defined as follows. Given certain workload specified by λ and $\bar{r}$, and n heterogeneous multiserver systems $S_1, S_2, \ldots, S_n$, where S_i is specified by m_i, ξ_i, α_i, P_i^*, for all $1 \le i \le n$, find a workload distribution $(\lambda_1, \lambda_2, \ldots, \lambda_n)$ and a server speed setting $(s_1, s_2, \ldots, s_n)$, such that the cost-performance ratio R is minimized, subject to the constraint that $\lambda_1 + \lambda_2 + \cdots + \lambda_n = \lambda$.

In the following (i.e., Equations (95)–(98)), we transform the above multivariable optimization problem to a nonlinear system of equations. We view $R(\lambda_1, \lambda_2, \ldots, \lambda_n, s_1, s_2, \ldots, s_n)$ as a function of $\lambda_1, \lambda_2, \ldots, \lambda_n, s_1, s_2, \ldots, s_n$. We can minimize $R(\lambda_1, \lambda_2, \ldots, \lambda_n, s_1, s_2, \ldots, s_n)$ subject to the constraint $C(\lambda_1, \lambda_2, \ldots, \lambda_n) = \lambda_1 + \lambda_2 + \cdots + \lambda_n = \lambda$ by using the following Lagrange multiplier system,

$$\nabla R(\lambda_1, \lambda_2, \ldots, \lambda_n, s_1, s_2, \ldots, s_n) = \phi C(\lambda_1, \lambda_2, \ldots, \lambda_n), \tag{95}$$

that is,

$$\frac{\partial R(\lambda_1, \lambda_2, \ldots, \lambda_n, s_1, s_2, \ldots, s_n)}{\partial \lambda_i} = \phi \frac{\partial C(\lambda_1, \lambda_2, \ldots, \lambda_n)}{\partial \lambda_i} = \phi, \tag{96}$$

where ϕ is a Lagrange multiplier, and

$$\frac{\partial R(\lambda_1, \lambda_2, \ldots, \lambda_n, s_1, s_2, \ldots, s_n)}{\partial s_i} = 0, \tag{97}$$

for all $1 \le i \le n$.

Therefore, we need to solve a nonlinear system of $2n + 1$ equations,

$$\begin{cases} H_0(\phi, \lambda_1, \ldots, \lambda_n, s_1, \ldots, s_n) = F_0(\phi, \lambda_1, \ldots, \lambda_n) = 0, \\ H_1(\phi, \lambda_1, \ldots, \lambda_n, s_1, \ldots, s_n) = F_1(\phi, \lambda_1, \ldots, \lambda_n) = 0, \\ \quad \vdots \\ H_n(\phi, \lambda_1, \ldots, \lambda_n, s_1, \ldots, s_n) = F_n(\phi, \lambda_1, \ldots, \lambda_n) = 0, \\ H_{n+1}(\phi, \lambda_1, \ldots, \lambda_n, s_1, \ldots, s_n) = G_1(s_1, \ldots, s_n) = 0, \\ \quad \vdots \\ H_{2n}(\phi, \lambda_1, \ldots, \lambda_n, s_1, \ldots, s_n) = G_n(s_1, \ldots, s_n) = 0, \end{cases} \tag{98}$$

with $2n+1$ unknowns, that is, $\lambda_1, \lambda_2, \dots, \lambda_n, s_1, \dots, s_n$, and ϕ. It is noticed that $H_i = F_i$ in Section 4.1 for all $0 \le i \le n$, and $H_{n+i} = G_i$ in Section 5.1 for all $1 \le i \le n$.

An analytical solution to the above equations is infeasible. We take an algorithmic and numerical approach.

6.2 | A numerical algorithm

We represent the variables $\phi, \lambda_1, \dots, \lambda_n, s_1, \dots, s_n$ using vector notation:

$$\mathbf{y} = (y_0, y_1, \dots, y_n, y_{n+1}, \dots, y_{2n}) = (\phi, \lambda_1, \dots, \lambda_n, s_1, \dots, s_n), \tag{99}$$

and $H_i(\phi, \lambda_1, \dots, \lambda_n, s_1, \dots, s_n) = H_i(y_0, y_1, \dots, y_n, y_{n+1}, \dots, y_{2n}) = H_i(\mathbf{y})$, where $H_i : \mathbb{R}^{2n+1} \to \mathbb{R}$ maps $(2n+1)$-dimensional space $\mathbb{R}^{2n+1}$ into the real line $\mathbb{R}$. By defining a function $\mathbf{H} : \mathbb{R}^{2n+1} \to \mathbb{R}^{2n+1}$ which maps $\mathbb{R}^{2n+1}$ into $\mathbb{R}^{2n+1}$,

$$\mathbf{H}(\mathbf{y}) = (H_0(y_0, y_1, \dots, y_{2n}), H_1(y_0, y_1, \dots, y_{2n}), \dots, H_{2n}(y_0, y_1, \dots, y_{2n})), \tag{100}$$

namely,

$$\mathbf{H}(\mathbf{y}) = (H_0(\mathbf{y}), H_1(\mathbf{y}), \dots, H_{2n}(\mathbf{y})), \tag{101}$$

our nonlinear system of equations becomes

$$\mathbf{H}(\mathbf{y}) = \mathbf{0}, \tag{102}$$

where $\mathbf{0} = (0, 0, \dots, 0)$.

Once more, we can solve the above nonlinear system of equations by using Newton's method. For this purpose, we use the Jacobian matrix $J(\mathbf{y})$ defined as

$$J(\mathbf{y}) = \begin{bmatrix} \frac{\partial H_0(\mathbf{y})}{\partial y_0} & \frac{\partial H_0(\mathbf{y})}{\partial y_1} & \cdots & \frac{\partial H_0(\mathbf{y})}{\partial y_n} & \frac{\partial H_0(\mathbf{y})}{\partial y_{n+1}} & \cdots & \frac{\partial H_0(\mathbf{y})}{\partial y_{2n}} \\ \frac{\partial H_1(\mathbf{y})}{\partial y_0} & \frac{\partial H_1(\mathbf{y})}{\partial y_1} & \cdots & \frac{\partial H_1(\mathbf{y})}{\partial y_n} & \frac{\partial H_1(\mathbf{y})}{\partial y_{n+1}} & \cdots & \frac{\partial H_1(\mathbf{y})}{\partial y_{2n}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial H_n(\mathbf{y})}{\partial y_0} & \frac{\partial H_n(\mathbf{y})}{\partial y_1} & \cdots & \frac{\partial H_n(\mathbf{y})}{\partial y_n} & \frac{\partial H_n(\mathbf{y})}{\partial y_{n+1}} & \cdots & \frac{\partial H_n(\mathbf{y})}{\partial y_{2n}} \\ \frac{\partial H_{n+1}(\mathbf{y})}{\partial y_0} & \frac{\partial H_{n+1}(\mathbf{y})}{\partial y_1} & \cdots & \frac{\partial H_{n+1}(\mathbf{y})}{\partial y_n} & \frac{\partial H_{n+1}(\mathbf{y})}{\partial y_{n+1}} & \cdots & \frac{\partial H_{n+1}(\mathbf{y})}{\partial y_{2n}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial H_{2n}(\mathbf{y})}{\partial y_0} & \frac{\partial H_{2n}(\mathbf{y})}{\partial y_1} & \cdots & \frac{\partial H_{2n}(\mathbf{y})}{\partial y_n} & \frac{\partial H_{2n}(\mathbf{y})}{\partial y_{n+1}} & \cdots & \frac{\partial H_{2n}(\mathbf{y})}{\partial y_{2n}} \end{bmatrix}, \tag{103}$$

which is actually

$$J(\mathbf{y}) = \begin{bmatrix} 0 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ -1 & \frac{\partial F_1(\mathbf{y})}{\partial \lambda_1} & \cdots & \frac{\partial F_1(\mathbf{y})}{\partial \lambda_n} & \frac{\partial F_1(\mathbf{y})}{\partial s_1} & \cdots & \frac{\partial F_1(\mathbf{y})}{\partial s_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -1 & \frac{\partial F_n(\mathbf{y})}{\partial \lambda_1} & \cdots & \frac{\partial F_n(\mathbf{y})}{\partial \lambda_n} & \frac{\partial F_n(\mathbf{y})}{\partial s_1} & \cdots & \frac{\partial F_n(\mathbf{y})}{\partial s_n} \\ 0 & \frac{\partial G_1(\mathbf{y})}{\partial \lambda_1} & \cdots & \frac{\partial G_1(\mathbf{y})}{\partial \lambda_n} & \frac{\partial G_1(\mathbf{y})}{\partial s_1} & \cdots & \frac{\partial G_1(\mathbf{y})}{\partial s_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{\partial G_n(\mathbf{y})}{\partial \lambda_1} & \cdots & \frac{\partial G_n(\mathbf{y})}{\partial \lambda_n} & \frac{\partial G_n(\mathbf{y})}{\partial s_1} & \cdots & \frac{\partial G_n(\mathbf{y})}{\partial s_n} \end{bmatrix}, \tag{104}$$

and equivalently,

$$J(\mathbf{y}) = \begin{bmatrix} 0 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ -1 & \dfrac{\partial^2 R}{\partial \lambda_1^2} & \cdots & \dfrac{\partial R}{\partial \lambda_1 \lambda_n} & \dfrac{\partial R}{\lambda_1 \partial s_1} & \cdots & \dfrac{\partial R}{\lambda_1 \partial s_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -1 & \dfrac{\partial R}{\partial \lambda_n \partial \lambda_1} & \cdots & \dfrac{\partial^2 R}{\partial \lambda_n^2} & \dfrac{\partial R}{\partial \lambda_n \partial s_1} & \cdots & \dfrac{\partial R}{\partial \lambda_n \partial s_n} \\ 0 & \dfrac{\partial R}{\partial s_1 \partial \lambda_1} & \cdots & \dfrac{\partial R}{\partial s_1 \partial \lambda_n} & \dfrac{\partial^2 R}{\partial s_1^2} & \cdots & \dfrac{\partial R}{\partial s_1 \partial s_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dfrac{\partial R}{\partial s_n \partial \lambda_1} & \cdots & \dfrac{\partial R}{\partial s_n \partial \lambda_n} & \dfrac{\partial R}{\partial s_n \partial s_1} & \cdots & \dfrac{\partial^2 R}{\partial s_n^2} \end{bmatrix}. \tag{105}$$

Notice that $\partial H_i(\mathbf{y})/\partial y_j = \partial F_i(\mathbf{y})/\partial \lambda_j$, $1 \le i, j \le n$, has been calculated in Section 4.2, and $\partial H_{n+i}(\mathbf{y})/\partial y_{n+j} = \partial G_i(\mathbf{y})/\partial s_j$, $1 \le i, j \le n$, has been calculated in Section 5.2.

In Equations (106)–(119), we derive other components of $J(\mathbf{y})$. (These detailed derivations can be skipped without loss of continuity.)

Now, we consider $\partial H_i(\mathbf{y})/\partial y_{n+j} = \partial F_i(\mathbf{y})/\partial s_j$. For all $1 \le i \le n$, we have

$$\begin{aligned} \frac{\partial F_i(\mathbf{y})}{\partial s_i} &= \frac{1}{\lambda}\left(\lambda_i \bar{r}\xi_i(\alpha_i - 1)s_i^{\alpha_i - 2}\left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i}\right) + P\left(\frac{\partial T_i}{\partial s_i} + \lambda_i \frac{\partial^2 T_i}{\partial \lambda_i \partial s_i}\right)\right) \\ &\quad + \bar{r}\xi_i(\alpha_i - 1)s_i^{\alpha_i - 2} T + \bar{r}\xi_i s_i^{\alpha_i - 1}\frac{\lambda_i}{\lambda} \cdot \frac{\partial T_i}{\partial s_i}, \end{aligned} \tag{106}$$

for the idle-speed model, and

$$\frac{\partial F_i(\mathbf{y})}{\partial s_i} = \frac{1}{\lambda}\left(m_i \xi_i \alpha_i s_i^{\alpha_i - 1}\left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i}\right) + P\left(\frac{\partial T_i}{\partial s_i} + \lambda_i \frac{\partial^2 T_i}{\partial \lambda_i \partial s_i}\right)\right), \tag{107}$$

for the constant-speed model. For all $1 \le i \ne j \le n$, we have

$$\frac{\partial F_i(\mathbf{y})}{\partial s_j} = \frac{1}{\lambda}\lambda_j \bar{r}\xi_j(\alpha_j - 1)s_j^{\alpha_j - 2}\left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i}\right) + \bar{r}\xi_i s_i^{\alpha_i - 1}\frac{\lambda_j}{\lambda} \cdot \frac{\partial T_j}{\partial s_j}, \tag{108}$$

for the idle-speed model, and

$$\frac{\partial F_i(\mathbf{y})}{\partial s_j} = \frac{1}{\lambda} m_j \xi_j \alpha_j s_j^{\alpha_j - 1}\left(T_i + \lambda_i \frac{\partial T_i}{\partial \lambda_i}\right), \tag{109}$$

for the constant-speed model.

Now, we consider $\partial H_{n+i}(\mathbf{y})/\partial y_j = \partial G_i(\mathbf{y})/\partial \lambda_j$. For all $1 \le i \le n$, we have

$$\frac{\partial G_i(\mathbf{y})}{\partial \lambda_i} = \frac{1}{\lambda}\left(P\frac{\partial T_i}{\partial s_i} + \lambda_i \bar{r}\xi_i s_i^{\alpha_i - 1}\frac{\partial T_i}{\partial s_i} + \lambda_i P \frac{\partial^2 T_i}{\partial s_i \partial \lambda_i}\right) + \bar{r}\xi_i(\alpha_i - 1)s_i^{\alpha_i - 2}\left(T + \lambda_i \frac{\lambda_i}{\lambda} \cdot \frac{\partial T_i}{\partial \lambda_i}\right), \tag{110}$$

for the idle-speed model, and

$$\frac{\partial G_i(\mathbf{y})}{\partial \lambda_i} = \frac{P}{\lambda}\left(\frac{\partial T_i}{\partial s_i} + \lambda_i \frac{\partial^2 T_i}{\partial s_i \partial \lambda_i}\right) + m_i \xi_i \alpha_i s_i^{\alpha_i - 1}\frac{\lambda_i}{\lambda} \cdot \frac{\partial T_i}{\partial \lambda_i}, \tag{111}$$

for the constant-speed model. For all $1 \le i \ne j \le n$, we have

$$\frac{\partial G_i(\mathbf{y})}{\partial \lambda_j} = \frac{\lambda_i}{\lambda}\bar{r}\xi_j s_j^{\alpha_j - 1}\frac{\partial T_i}{\partial s_i} + \lambda_i \bar{r}\xi_i(\alpha_i - 1)s_i^{\alpha_i - 2}\frac{\lambda_j}{\lambda} \cdot \frac{\partial T_j}{\partial \lambda_j}, \tag{112}$$

for the idle-speed model, and

$$\frac{\partial G_i(\mathbf{y})}{\partial \lambda_j} = m_i \xi_i \alpha_i s_i^{\alpha_i - 1} \frac{\lambda_j}{\lambda} \cdot \frac{\partial T_j}{\partial \lambda_j}, \tag{113}$$

for the constant-speed model.

Furthermore, we have

$$\begin{aligned}\frac{\partial^2 T_i}{\partial \lambda_i \partial s_i} = &-\frac{1}{s_i} \cdot \frac{\partial T_i}{\partial \lambda_i} + \frac{\overline{r}}{m_i s_i} \left(-\frac{6\rho_i p_{i,m_i}}{(1-\rho_i)^4} \cdot \frac{\overline{r}}{m_i s_i^2} - \frac{2p_{i,m_i}}{(1-\rho_i)^3} \cdot \frac{\overline{r}}{m_i s_i^2} + \frac{2}{(1-\rho_i)^3} \cdot \frac{\overline{r}}{m_i s_i} \cdot \frac{\partial p_{i,m_i}}{\partial s_i} \right. \\ &\left. - \frac{2\rho_i}{(1-\rho_i)^3 s_i} \cdot \frac{\partial p_{i,m_i}}{\partial \lambda_i} + \frac{1}{(1-\rho_i)^2} \cdot \frac{\partial^2 p_{i,m_i}}{\partial \lambda_i \partial s_i} \right),\end{aligned} \tag{114}$$

and

$$\frac{\partial^2 p_{i,m_i}}{\partial \lambda_i \partial s_i} = \frac{m_i^{m_i}}{m_i!} \left(-(m_i - 1) \frac{\rho_i^{m_i - 1}}{s_i} \left(\frac{\overline{r}}{s_i} p_{i,0} + \rho_i \frac{\partial p_{i,0}}{\partial \lambda_i} \right) + \rho_i^{m_i - 1} \left(-\frac{\overline{r}}{s_i^2} p_{i,0} + \frac{\overline{r}}{s_i} \cdot \frac{\partial p_{i,0}}{\partial s_i} - \frac{\rho_i}{s_i} \cdot \frac{\partial p_{i,0}}{\partial \lambda_i} + \rho_i \frac{\partial^2 p_{i,0}}{\partial \lambda_i \partial s_i} \right) \right), \tag{115}$$

and

$$\begin{aligned}\frac{\partial^2 p_{i,0}}{\partial \lambda_i \partial s_i} = &-\frac{1}{s_i} \cdot \frac{\partial p_{i,0}}{\partial \lambda_i} - 2p_{i,0} \frac{\partial p_{i,0}}{\partial s_i} \left(\sum_{k=1}^{m_i - 1} \frac{m_i^{k-1}}{(k-1)!} \rho_i^{k-1} + \frac{m_i^{m_i - 1}}{m_i!} \cdot \frac{m_i \rho_i^{m_i - 1} - (m_i - 1)\rho_i^{m_i}}{(1-\rho_i)^2} \right) \frac{\overline{r}}{s_i} \\ &+ p_{i,0}^2 \left(\sum_{k=2}^{m_i - 1} \frac{m_i^{k-1}}{(k-2)!} \rho_i^{k-2} + \frac{m_i^{m_i - 1}}{m_i!} \cdot \frac{m_i(m_i - 1)\rho_i^{m_i - 2} - 2m_i(m_i - 2)\rho_i^{m_i - 1} + (m_i - 2)(m_i - 1)\rho_i^{m_i}}{(1-\rho_i)^3} \right) \frac{\overline{r}\rho_i}{s_i^2},\end{aligned} \tag{116}$$

for all $1 \leq i \leq n$.

Furthermore, we have

$$\frac{\partial^2 T_i}{\partial s_i \partial \lambda_i} = -\frac{1}{s_i} \cdot \frac{\partial T_i}{\partial \lambda_i} - \frac{\overline{r}}{m_i s_i} \left(\frac{2(1+2\rho_i)\overline{r}}{(1-\rho_i)^4 m_i s_i^2} p_{i,m_i} + \frac{2\rho_i}{(1-\rho_i)^3 s_i} \cdot \frac{\partial p_{i,m_i}}{\partial \lambda_i} - \frac{2\overline{r}}{(1-\rho_i)^3 m_i s_i} \cdot \frac{\partial p_{i,m_i}}{\partial s_i} - \frac{1}{(1-\rho_i)^2} \cdot \frac{\partial^2 p_{i,m_i}}{\partial s_i \partial \lambda_i} \right), \tag{117}$$

and

$$\frac{\partial^2 p_{i,m_i}}{\partial s_i \partial \lambda_i} = \frac{m_i^{m_i}}{m_i!} \left(m_i \rho_i^{m_i - 1} \frac{\overline{r}}{m_i s_i} \left(-\frac{m_i}{s_i} p_{i,0} + \frac{\partial p_{i,0}}{\partial s_i} \right) + \rho_i^{m_i} \left(-\frac{m_i}{s_i} \cdot \frac{\partial p_{i,0}}{\partial \lambda_i} + \frac{\partial^2 p_{i,0}}{\partial s_i \partial \lambda_i} \right) \right), \tag{118}$$

and

$$\begin{aligned}\frac{\partial^2 p_{i,0}}{\partial s_i \partial \lambda_i} = &\, 2p_{i,0} \frac{\overline{r}}{m_i s_i} \left(\sum_{k=1}^{m_i - 1} \frac{m_i^k}{(k-1)!} \rho_i^{k-1} + \frac{m_i^{m_i}}{m_i!} \cdot \frac{m_i \rho_i^{m_i - 1} - (m_i - 1)\rho_i^{m_i}}{(1-\rho_i)^2} \right) \frac{\rho_i}{s_i} \\ &+ p_{i,0}^2 \left(\sum_{k=1}^{m_i - 1} \frac{m_i^k}{(k-1)!} \rho_i^{k-1} + \frac{m_i^{m_i}}{m_i!} \cdot \frac{m_i \rho_i^{m_i - 1} - (m_i - 1)\rho_i^{m_i}}{(1-\rho_i)^2} \right) \frac{\overline{r}}{m_i s_i^2} \\ &+ p_{i,0}^2 \left(\sum_{k=2}^{m_i - 1} \frac{m_i^k}{(k-2)!} \rho_i^{k-2} + \frac{m_i^{m_i}}{m_i!} \cdot \frac{m_i(m_i - 1)\rho_i^{m_i - 2} - 2m_i(m_i - 2)\rho_i^{m_i - 1} + (m_i - 2)(m_i - 1)\rho_i^{m_i}}{(1-\rho_i)^3} \right) \frac{\rho_i \overline{r}}{m_i s_i^2},\end{aligned} \tag{119}$$

for all $1 \leq i \leq n$.

Algorithm 3 gives our numerical algorithm for finding an optimal workload distribution $(\lambda_1, \lambda_2, \ldots, \lambda_n)$, an optimal server speed setting $(s_1, s_2, \ldots, s_n)$, and the Lagrange multiplier ϕ, that is, the vector

$$\mathbf{y} = (\phi, \lambda_1, \ldots, \lambda_n, s_1, \ldots, s_n), \tag{120}$$

Algorithm 3. Optimal workload management and server speed setting

Input:Parameters λ, $\bar{r}$,m_i, ξ_i, α_i, P_i^*,for all $1 \le i \le n$.
Output:An optimal workload distribution $(\lambda_1, \lambda_2, \dots, \lambda_n)$,an optimal server speed setting $(s_1, s_2, \dots, s_n)$,and ϕ, that is,$\mathbf{y} = (\phi, \lambda_1, \dots, \lambda_n, s_1, \dots, s_n)$,which satisfies $\mathbf{H}(\mathbf{y}) = \mathbf{0}$.

$\mathbf{y} \leftarrow (1, (\lambda/\lambda^*)(m_1 s_1/\bar{r}), \dots, (\lambda/\lambda^*)(m_n s_n/\bar{r}), s_1, \dots, s_n)$;
repeat
Calculate $J(\mathbf{y})$,where $J(\mathbf{y})_{i,j} = \partial H_i(\mathbf{y})/\partial y_j$for $0 \le i,j \le 2n$;
Calculate $\mathbf{H}(\mathbf{y}) = (H_0(\mathbf{y}), H_1(\mathbf{y}), \dots, H_{2n}(\mathbf{y}))$;
Solve the linear system of equations $J(\mathbf{y})\mathbf{z} = -\mathbf{H}(\mathbf{y})$;
$\mathbf{y} \leftarrow \mathbf{y} + \mathbf{z}$;
until $\|\mathbf{z}\| \le \epsilon$.

which satisfies the nonlinear system of equations $\mathbf{H}(\mathbf{y}) = \mathbf{0}$. The time and space complexities of Algorithm 3 are the same as those of Algorithms 1 and 2.

6.3 | Performance data

Let us consider the same heterogeneous multiserver systems S_1, S_2, S_3 in Sections 4.3 and 5.3.

In Tables 7 and 8, we show the optimal workload distribution $(\lambda_1, \lambda_2, \lambda_3)$, the optimal server speed setting (s_1, s_2, s_3), the corresponding server utilization (ρ_1, ρ_2, ρ_3), and the minimized cost-performance ratio R, for $\lambda = 13, 14, \dots, 22$. We notice that compared with Tables 5 and 6, the reduction of R is not significant. This means that optimal server speed setting has more impact than optimal workload distribution. Although the workload distribution in Tables 5 and 6 is not optimal, the resulted R by optimal server speed setting alone can already generate close-to-optimal R.

We would like to mention that as an approximate solution, we can set $\lambda_i = (m_i s_i/\bar{r})\rho$, for all $1 \le i \le n$, such that all servers have the same utilization $\rho = \lambda/\lambda^*$, where λ^* is defined in Section 4.2. This implies that each λ_i is a function of $s_1, s_2, \dots, s_n$. Hence, our problem of workload management and server speed setting only has n unknowns, that is, $s_1, s_2, \dots, s_n$. However, obtaining such an approximate solution is by no means straightforward, and is probably not worth of investigation, since there is no accurate and analytical solution.

TABLE 7 Numerical data for optimal workload management and server speed setting (idle-speed model)

	S_1			S_2			S_3			
λ	λ_1	s_1	ρ_1	λ_2	s_2	ρ_2	λ_3	s_3	ρ_3	R
13	3.10938	1.55960	0.49843	4.30110	1.57669	0.54559	5.58952	1.59901	0.58260	144.202
14	3.36255	1.60383	0.52414	4.63202	1.62410	0.57041	6.00543	1.65005	0.60659	155.036
15	3.61221	1.65505	0.54563	4.96227	1.67902	0.59109	6.42552	1.70922	0.62655	167.042
16	3.85874	1.71175	0.56357	5.29195	1.73986	0.60832	6.84931	1.77484	0.64319	180.276
17	4.10246	1.77277	0.57854	5.62113	1.80540	0.62270	7.27641	1.84563	0.65708	194.781
18	4.34364	1.83726	0.59105	5.94987	1.87476	0.63473	7.70649	1.92066	0.66874	210.587
19	4.58255	1.90456	0.60152	6.27821	1.94724	0.64483	8.13925	1.99920	0.67854	227.716
20	4.81938	1.97416	0.61031	6.60618	2.02230	0.65333	8.57443	2.08067	0.68683	246.185
21	5.05434	2.04565	0.61769	6.93382	2.09952	0.66052	9.01183	2.16463	0.69387	266.007
22	5.28760	2.11871	0.62392	7.26115	2.17854	0.66661	9.45126	2.25070	0.69987	287.192

TABLE 8 Numerical data for optimal workload management and server speed setting (constant-speed model)

	S_1			S_2			S_3			
λ	λ_1	s_1	ρ_1	λ_2	s_2	ρ_2	λ_3	s_3	ρ_3	R
13	3.14940	1.24554	0.63213	4.31316	1.27982	0.67402	5.53744	1.30901	0.70504	191.458
14	3.40617	1.30862	0.65072	4.64382	1.34482	0.69063	5.95001	1.37679	0.72028	205.215
15	3.65947	1.37333	0.66617	4.97415	1.41212	0.70450	6.36638	1.44747	0.73305	220.688
16	3.90981	1.43945	0.67904	5.30415	1.48139	0.71610	6.78604	1.52063	0.74378	237.902
17	4.15755	1.50680	0.68980	5.63384	1.55237	0.72584	7.20861	1.59595	0.75280	256.877
18	4.40300	1.57520	0.69880	5.96324	1.62481	0.73402	7.63376	1.67314	0.76042	277.631
19	4.64638	1.64449	0.70636	6.29235	1.69853	0.74092	8.06126	1.75199	0.76687	300.176
20	4.88791	1.71454	0.71271	6.62121	1.77334	0.74675	8.49088	1.83228	0.77234	324.523
21	5.12775	1.78524	0.71808	6.94981	1.84910	0.75170	8.92244	1.91384	0.77701	350.680
22	5.36606	1.85648	0.72261	7.27818	1.92569	0.75590	9.35577	1.99653	0.78100	378.654

7 | CONCLUDING REMARKS

We have established a framework to study the power-performance tradeoff in a data center for cloud computing, which consists of three different levels, three different perspectives, and two effective techniques. We have dealt with the power-performance tradeoff at the data center level by considering multiple heterogeneous multiserver systems, which are treated as M/M/m queueing systems with two power consumption models. We have studied the important and fundamental issue of cost-performance ratio optimization by using the techniques of workload management and server speed setting. In particular, we have formulated and solved three multivariable optimization problems, that is, the workload management problem, the server speed setting problem, and the workload management and server speed setting problem. Our method to solve these problems is to solve the equivalent nonlinear systems of equations using numerical algorithms.

We would like to make the following comments regarding the practicability and applicability of our approach. All our optimization problems are defined with just a few parameters which are easily available in any data center. The kernel of all our algorithms is to solve linear systems of equations, which can be implemented in $O(n^2)$ time and space. Our experiments reveal that all our algorithms can be implemented very efficiently. For instance, all the date in Tables 3–8 can be obtained in just seconds. Therefore, we would like to emphasize that the low computational costs of our algorithms make it easy to integrate and incorporate them into a real-world system. Furthermore, the low time and space complexities of our numerical procedures make our algorithms scalable to large data centers with many heterogeneous multiserver systems.

We point out two possible directions for further research. First, it will be interesting and important to consider more general queueing models, for example, M/G/m and G/G/m, for multiserver systems. However, for these models, there might be only approximate expressions of the average task response time. Therefore, analytical results of cost-performance ratio optimization should be verified by simulations and experiments. Such investigation is certainly challenging, but very useful in real applications. Second, since applications can be classified and categorized into various types, cost-performance ratio optimization can be conducted for each type of applications. Such optimization requires more sophisticated queueing models and more involved power and performance analysis. Fortunately, the framework and methodology developed in this article should still be effective and applicable.

ACKNOWLEDGMENTS

The author deeply appreciates the editor and the anonymous reviewers for their extensive suggestions and comments on improving the presentation of the manuscript.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

ORCID

Keqin Li https://orcid.org/0000-0001-5224-4048

REFERENCES

1. Abdelmaboud A, Jawawi DNA, Ghani I, Elsafi A, Kitchenham B. Quality of service approaches in cloud computing: a systematic mapping study. *J Syst Software*. 2015;101:159-179.
2. Garg SK, Versteeg S, Buyya R. A framework for ranking of cloud computing services. *Future Gen Comput Syst*. 2013;29(4):1012-1023.
3. Jennings B, Stadler R. Resource management in clouds: survey and research challenges. *J Network Syst Managt*. 2015;23:567-619.
4. Singh S, Chana I. A survey on resource scheduling in cloud computing: issues and challenges. *J Grid Comput*. 2016;14:217-264.
5. Ardagna D, Casale G, Ciavotta M, Pérez JF, Wang W. Quality-of-service in cloud computing: modeling techniques and their applications. *J Int Serv Appl*. 2014;5(11):17.
6. Bardsiri AK, Hashemi SM. QoS metrics for cloud computing services evaluation. *Int J Int Syst Appl*. 2014;12:27-33.
7. Li K. Power allocation and task scheduling on multiprocessor computers with energy and time constraints. In: Zomaya AY, Lee YC, eds. *Energy-Efficient Distributed Computing Systems*. John Wiley & Sons; 2012:1-37.
8. Li K. Analytical modeling and optimization of an elastic cloud server system. In: Adamatzky A, Akl SG, Sirakoulis GC, eds. *From Parallel to Emergent Computing*. Taylor & Francis Group of CRC Press; 2019:31-48.
9. Li K. Optimal load distribution for multiple classes of applications on heterogeneous servers with variable speeds. *Software Practice Exp*. 2018;48(10):1805-1819.
10. Cao J, Li K, Stojmenović I. Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers. *IEEE Trans Comput*. 2014;63(1):45-58.
11. He Z, Li K, Li K, Zhou W, Liu J. Server configuration optimization in mobile edge computing: a cost-performance tradeoff perspective. *Software Practice Exp*. 2021;51(9):1847-1981.
12. Gandhi A, Gupta V, Harchol-Balter M, Kozuch MA. Optimality analysis of energy-performance trade-off for server farm management. *Performance Evaluation*. 2010;67(11):1155-1171.
13. Li K. Optimal configuration of a multicore server processor for managing the power and performance tradeoff. *J Supercomput*. 2012;61(1):189-214.
14. Li K. Improving multicore server performance and reducing energy consumption by workload dependent dynamic power management. *IEEE Trans Cloud Comput*. 2016;4(2):122-137.
15. Li K. Auto speed scaling scheme optimization for elastic cloud computing platforms. In: Clary TS, ed. *Horizons in Computer Science Research*. Vol 17. Nova Science Publishers, Inc.; 2018:183-224.
16. Li K. Optimal speed setting for cloud servers with mixed applications. *IEEE Trans Industrial Info*. 2019;15(4):1947-1955.
17. Li K. Quantitative modeling and analytical calculation of elasticity in cloud computing. *IEEE Trans Cloud Comput*. 2020;8(4):1135-1148.
18. He Z, Li K, Li K. Cost-efficient server configuration and placement for mobile edge computing. *IEEE Trans Parallel Distrib Syst*. 2022; 33(9):2198-2212. doi:10.1109/TPDS.2021.3135955
19. Huang J, Li R, An J, Ntalasha D, Yang F, Li K. Energy-efficient resource utilization for heterogeneous embedded computing systems. *IEEE Trans Comput*. 2017;66(9):1518-1531.
20. Huang J, Liu Y, Li R, et al. Optimal power allocation and load balancing for non-dedicated heterogeneous distributed embedded computing systems. *J Parallel Distributed Comput*. 2019;130:24-36.
21. Huang J, Li R, Wei Y, An J, Chang W. Bi-directional timing-power optimisation on heterogeneous multi-core architectures. *IEEE Trans Sustainable Comput*. 2021;6(4):572-585.
22. Li K. Optimal power allocation among multiple heterogeneous servers in a data center. *Sustainable Comput Info Syst*. 2012;2(1):13-22.
23. Li K. Optimal task dispatching for multiple heterogeneous multiserver systems with dynamic speed and power management. *IEEE Trans Sustainable Comput*. 2017;2(2):167-182.
24. Li K. Optimal power and performance management for heterogeneous and arbitrary cloud servers. *IEEE Access*. 2019;7(1):5071-5084.
25. Tian Y, Lin C, Li K. Managing performance and power consumption tradeoff for multiple heterogeneous servers in cloud computing. *Cluster Comput*. 2014;17(3):943-955.
26. Yang B, Li Z, Chen S, Wang T, Li K. A Stackelberg game approach for energy-aware resource allocation in data centers. *IEEE Trans Parallel Distrib Syst*. 2016;27(12):3646-3658.
27. Zheng X, Cai Y. Achieving energy proportionality in server clusters. *Int J Comput Networks*. 2010;1(2):21-35.
28. Deng R, Lu R, Lai C, Luan TH, Liang H. Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE Internet Things J*. 2016;3(6):1171-1181.
29. Ding D, Fan X, Zhao Y, Kang K, Yin Q, Zeng J. Q-learning based dynamic task scheduling for energy-efficient cloud computing. *Future Gen Comput Syst*. 2020;108:361-371.
30. Zhou Z, Abawajy J, Chowdhury M, et al. Minimizing SLA violation and power consumption in cloud data centers using adaptive energy-aware algorithms. *Future Gen Comput Syst*. 2018;86:836-850.
31. Zhou Z, Li K, Abawajy J, et al. An adaptive energy-aware stochastic task execution algorithm in virtualized networked datacenters. *IEEE Trans Sustainable Comput*. 2022;7(2):371-385. doi:10.1109/TSUSC.2021.3115388
32. Mao S, Leng S, Maharjan S, Zhang Y. Energy efficiency and delay tradeoff for wireless powered mobile-edge computing systems with multi-access schemes. *IEEE Trans Wireless Commun*. 2020;19(3):1855-1867.

33. Qin M, Cheng N, Jing Z, et al. Service-oriented energy-latency tradeoff for IoT task partial offloading in MEC-enhanced multi-RAT networks. *IEEE Internet Things J*. 2021;8(3):1896-1907.
34. Tao X, Ota K, Dong M, Qi H, Li K. Performance guaranteed computation offloading for mobile-edge cloud computing. *IEEE Wireless Commun Lett*. 2017;6(6):774-777.
35. Zhang G, Zhang W, Cao Y, Li D, Wang L. Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices. *IEEE Trans Industrial Info*. 2018;14(10):4642-4655.
36. Zhang J, Hu X, Ning Z, et al. Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks. *IEEE Internet Things J*. 2018;5(4):2633-2645.
37. Kong S, Li Y, Feng L. Cost-performance driven resource configuration for database applications in IaaS cloud environments. In: Ivanov I, van Sinderen M, Shishkov B, eds. *Cloud Computing and Services Science*. Springer; 2012:111-129.
38. Kleinrock L. *Queueing Systems, Volume 1: Theory*. John Wiley and Sons; 1975.
39. Burden RL, Faires JD, Reynolds AC. *Numerical Analysis*. 2nd ed. Prindle, Weber & Schmidt; 1981.

How to cite this article: Li K. Workload management and server speed setting for cost-performance ratio optimization. *Softw Pract Exper*. 2022;1-29. doi: 10.1002/spe.3140