Received 13 December 2019; revised 20 January 2020; accepted 2 February 2020. Date of publication 6 February 2020; date of current version 26 February 2020.

Digital Object Identifier 10.1109/OJCOMS.2020.2971613

A Game-Theoretic Approach for Non-Cooperative Load Balancing Among Competing Cloudlets

SOURAV MONDAL [1] (Student Member, IEEE), GOUTAM DAS [2] (Member, IEEE), AND ELAINE WONG [1] (Senior Member, IEEE)

[1]Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, VIC 3010, Australia
[2]G. S. Sanyal School of Telecommunications, Indian Institute of Technology Kharagpur, Kharagpur 721302, India

CORRESPONDING AUTHOR: S. MONDAL (e-mail: smondal@student.unimelb.edu.au)

ABSTRACT To deliver high performance and reliability to the mobile users in accessing mobile cloud services, the major interest is currently given to the integration of centralized cloud computing and distributed edge computing infrastructures. In such a heterogeneous network ecosystem, multiple cloudlets from different service providers coexist. However, to meet the stringent latency requirements of computation-intensive and mission-critical applications, overloaded cloudlets can offload some of the incoming job requests to their relatively under-loaded neighboring cloudlets. In this paper, we propose a novel economic and non-cooperative game-theoretic model for load balancing among competitive cloudlets. This model aims to maximize the utilities of all the competing cloudlets while meeting the end-to-end latency of the users. We characterize the problem as a generalized Nash equilibrium problem and investigate the existence and uniqueness of a pure-strategy Nash equilibrium. We design a variational inequality based algorithm to compute the pure-strategy Nash equilibrium. We show that all the competing cloudlets are able to maximize their utilities by employing our proposed Nash equilibrium computation offload strategy in both under- and overloaded conditions. We also show through numerical evaluations that our load balancing model outperforms some of the existing game-theoretic load balancing frameworks, especially in a highly overloaded condition.

INDEX TERMS Cloudlet computing, non-cooperative load balancing, generalized Nash equilibrium, variational inequality.

I. INTRODUCTION

THE MOBILE cloud computing technology facilitated the usage of communication and computation intensive applications in mobile devices by compensating their limitations in battery, memory, and computational resources. With the proliferation of low-latency applications such as virtual reality, augmented reality, teleoperation, online gaming, and tactile Internet, requiring an end-to-end latency constraint of 10-100 ms, the evolution of *edge-computing solutions* was expedited [1]. Based on the recent proposal for the use of clusters of computers called *cloudlets* [2], researchers designed efficient frameworks for cloudlet placement over wireless access networks [3]–[5]. In addition to the overwhelming popularity of wireless access networks, very recently the authors of [6]–[10] proposed cloudlet placement frameworks over optical and fiber-wireless access networks.

Note that cloudlet computing systems are essentially distributed computing systems, and in any distributed system, *job request allocation* and *load balancing* are considered as important research challenges [11]. To address the job request problem, the authors of [12]–[16] proposed efficient frameworks for *job request allocation from mobile devices to cloudlets*. Nonetheless, due to the highly dynamic mobility pattern of mobile devices and randomness of job request arrival process, cloudlets get overloaded or under-loaded at different points of time. Thus, researchers realized the necessity of designing efficient frameworks for *load balancing among neighboring cloudlets* as an immediate challenge [17] and the authors of [18]–[26] designed efficient load balancing frameworks among neighboring cloudlets.

We critically observe that minimization of end-to-end latency of the cloudlets is stressed in most of the existing

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

works on load balancing among cloudlets. However, practical users are satisfied if the job requests are processed within the predefined quality-of-service (QoS) latency target. Hence, latency minimization beyond the QoS latency target for the cloudlets is not always a desired objective from a practical viewpoint. For example, a user is equally satisfied if a job request is processed by 5 ms, 9 ms, or 10 ms, when the actual QoS latency target is 10 ms. Nonetheless, if the cloudlets fail to meet the QoS latency target, a significant penalty should be incurred, especially for low-latency applications. With this realization, we design a novel game-theoretic objective function that yields the maximum utility for cloudlets when the end-to-end latency is equal to the QoS latency target. Furthermore, our designed objective function provides an opportunity to each cloudlet for accepting some extra load from their neighboring cloudlets and gain some economic benefit, whenever the concerned cloudlet is operating well within its QoS latency target.

We also observe that none of the existing load balancing frameworks considers a *practical heterogeneous cloudlet environment*, where cloudlets from the same as well as different service providers compete with each other over the same customer base. This implies that each cloudlet needs to pay some extra incentive for offloading job requests to neighboring cloudlets from different service providers, but no such payment is required for offloading job requests to cloudlets from the same service provider. Thus, the multi-party economic interaction among heterogeneous neighboring cloudlets has yet to be captured in [18]–[26] and the implementation of economic and game theoretic models to analyze the interaction among multiple cloudlets to participate in the market competition have yet to be studied. Note that an optimization framework formulates a common objective function for all the neighboring cloudlets to decide their optimal load balancing strategies. When all the neighboring cloudlets belong to the same service provider, they abide by this solution. Nonetheless, when cloudlets from the same as well as different service providers coexist, some cloudlets may deviate from a centralized optimal solution if they find it more profitable. Thus, a game-theoretic framework becomes inevitable for such cases as a game-theoretic solution like Nash equilibrium (NE) is *self-enforcing* in nature [27].

Recently, researchers are also exploring various machine learning based approaches to solve practical problems, but complex supervised learning models like artificial neural networks rely heavily on historical data to make decisions. Such frameworks are highly inefficient for load balancing among cloudlets in real-time, especially in a dynamic scenario where there is low correlation between the trained data and real-time data [28]. We also observe that a large body of the existing works on load balancing among cloudlets use distributed frameworks, but they are not designed for job requests with stringent QoS latency targets. Thus, we propose a centralized control protocol that makes the load balancing decision before the actual job request arrival and cloudlets can start to process them immediately after arrival. In many practical market competitions, the business among various service providers is facilitated through a mediator who provides some mandatory rules or guidelines [29]. Similarly, in this work also, we consider that *a neutral mediator supervises the computation offloading game amongst the cloudlets*. Our conjecture is that the proposed mediator installs some computational facility in the neighborhood of the cloudlets for computation of NE and performing other network management operations. If the computational facility is installed by some particular service provider, it may produce biased load balancing decisions. All the competing cloudlets send their predicted job request arrival rates to the mediator, which computes the NE load balancing strategies for the cloudlets, and broadcasts to them before the actual job request arrival. Our primary contributions in this paper are summarized as follows:

(i) We formulate the load balancing problem among cloudlets from multiple service providers as a novel economic and non-cooperative game theoretic problem. In this setup, each neighboring cloudlet, both from the same and different service providers, gets a scope to maximize its utility while meeting the QoS latency targets of mobile device users.

(ii) Furthermore, we show that the objective functions of all cloudlets and the constraints create a convex problem. Thus, by applying the variational inequality approach we solve the formulated generalized NE (GNE) problem and design an efficient centralized algorithm suitable for our problem.

(iii) Finally, we show that all the participating cloudlets are able to achieve higher utilities under different network load conditions by mutual computation offloading according to our proposed NE strategies than some of the recent game-theoretic load balancing models. The performance of our proposed model in terms of average end-to-end latency and utility values is also better than these frameworks, especially in highly overloaded conditions.

The rest of this paper is organized as follows. Section II reviews some related works. Section III presents the fundamental system design considerations and control design for the load balancing game among neighboring cloudlets. Section IV formulates the non-cooperative game and discusses properties like existence and uniqueness of the GNE of the game. Section V designs an efficient algorithm for the computation of the GNE. Section VI presents and discusses the simulation results. Finally, Section VII summarizes our primary observations and achievements by using the game theoretic framework.

II. RELATED WORKS

Load balancing among edge computing nodes like cloudlets is an important research problem and recently, a few researchers proposed load balancing models based on optimization and game-theoretic methods. In existing literature, primarily centralized and decentralized control models

TABLE 1. Comparison with existing load balancing works.

Models	Network setup	Objective	Solution approach
[18]	Centralized	Minimize maximum response time subject to communication constraints	Network-flow based heuristic algorithm
[19]	Centralized	Minimize blocking probability and waiting time experienced by mobile users	Analytical closed-form expressions
[24]	Centralized	Minimize overall end-to-end latency subject to communication constraints	Non-cooperative game theory, variational inequality and iterative proximal algorithm
[21]	Decentralized	Minimize blocking probability of cloudlets subject to the execution delay of the tasks	Dynamic cooperative game-theoretic algorithm
[22]	Decentralized	Maximize the payoff function of the edge devices consisting of reward, penalty and cost components	Cooperative game-theoretic framework with a dynamic incentive feedback mechanism
[23]	Decentralized	Minimize the overall system cost and maximize total number of mobile users	Potential non-cooperative game with decentralized algorithm to compute NE
[25]	Centralized, decentralized	Minimize overall latency cost subject to energy, latency constraints	Lyapunov drift-plus-penalty algorithm, distributed non-cooperative game theory
Our game	Centralized	Maximize economic utility while meeting QoS latency requirements	Economic and non-cooperative game theory, variational-inequality based algorithm

are used to solve load balancing problems [17]. In *centralized control models*, an oracle controller node makes the load balancing decision that is aware of the entire network status in real-time. All the distributed nodes under its supervision send their local information to the controller node and in turn, they are informed about their load balancing strategies. This model is very easy to implement but may face some performance issues due to inter-node communication bottleneck when the network is spread across large geography and dynamic in nature. The authors of [18] designed a centralized framework by formulating a latency minimization problem and proposing a network-flow based heuristic algorithm to solve the problem. In *optimization method based models*, a common objective function and a set of constraints are formulated to compute the optimal load balancing strategies for all the cloudlets. Such models can provide fast and efficient methods to solve the problem, but they are difficult to impose on a practical network scenario where cloudlets from different service providers coexist. The authors of [19] studied and compared three load balancing schemes, viz., no sharing, random sharing, and least loaded sharing among neighboring cloudlets with different degrees of collaboration.

On the other hand, in *decentralized control models*, all the distributed nodes decide their load balancing strategies through their local interaction with neighboring nodes and the supervision of a controller node is not required. In [20], the authors proposed a distributed load balancing scheme for minimizing the average latency of Internet-of-Things devices associated to fog nodes co-located with cellular base stations. Although this model is more robust for large networks, it introduces excessive exchange of control messages and computational load in the network. Recently, the interest to apply cooperative and non-cooperative game-theoretic models on various network-related problems is growing because game theory provides several powerful methods to analyze and study the interaction among distributed agents under conflicts and cooperation [27]. The authors of [21] proposed a cooperative load balancing scheme where under-loaded cloudlets cooperate with their neighboring overloaded cloudlets to minimize their blocking probability and processing latency. In [22], the authors proposed a cooperative game-theoretic task allocation framework with a dynamic incentive feedback mechanism. The authors showed that their proposed framework can satisfy QoS requirements of the applications while the edge devices gained a much higher payoff compared to state-of-the-art frameworks. Again, the authors of [23] formulated a potential non-cooperative game for cost-effective edge user allocation to edge computing nodes that also keeps the workload distribution balanced. The authors showed that this problem admits a unique NE solution and designed an efficient decentralized algorithm to compute the NE. In addition to this, reinforcement learning algorithms can also prove to be a useful approach for solving load balancing problems but may present various complexity and convergence challenges in real-time [30].

Two recent works, [24], [25] are very much close to our current work. In [25], the authors proposed a distributed non-cooperative load balancing game among neighboring cloudlets in small-cell networks and compared its performance with a centralized load balancing framework. In this game formulation, each cloudlet tries to minimize their end-to-end latency cost subject to explicit energy and latency constraints. Hence, this model performs really well if the network is moderately loaded, but performs very poorly under high load conditions. However, when the latency constraints of the cloudlets start to violate, the overloaded cloudlets are no longer able to offload any job requests and the overall latency performance degrades. In [24], the authors formulate a non-cooperative load balancing game by defining the expected latency of each cloudlet as a disutility function and try to minimize its value. The authors propose an iterative proximal algorithm to compute a Nash equilibrium solution. In this algorithm, at first all the cloudlets are sorted depending on their server availability and none of the cloudlets is allowed to offload until their incoming job requests reach a certain threshold. Nonetheless, due to this load balancing strategy, it happens often that the most under-loaded cloudlets receive a large number of job requests from

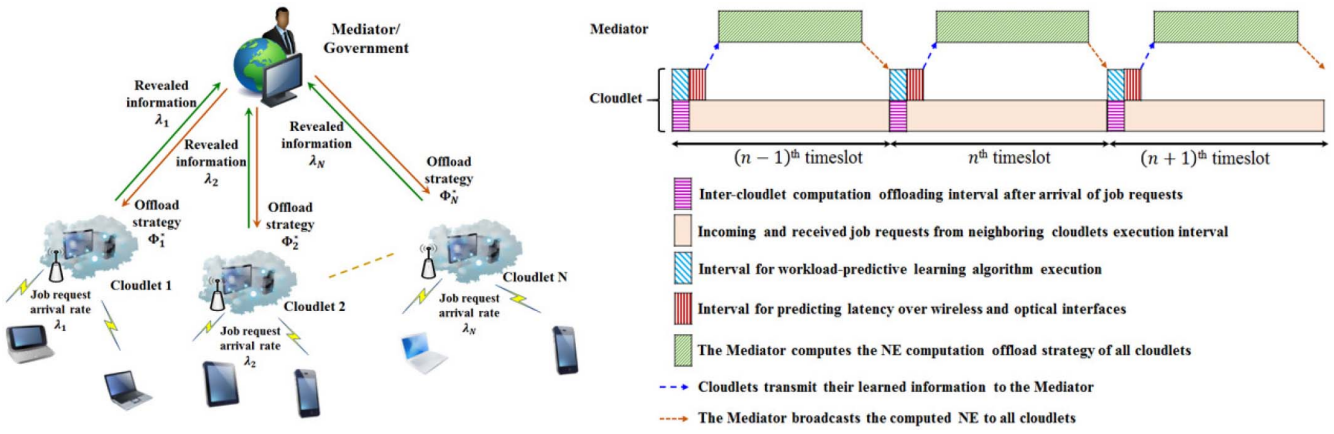


FIGURE 1. A schematic diagram showing the interactions among *N* competing cloudlets, supervised by a neutral mediator and the a timing diagram illustrating the overall control design.

their overloaded neighboring cloudlets and the end-to-end latency overshoots.

To avoid the aforementioned issues, we incorporate the QoS latency target in our game formulation in a tactful way so that the game does not become infeasible, even under high load conditions. Thus, overloaded cloudlets may not be able to offload their complete extra load to their under-loaded neighbors, but offload to the maximum extent possible. In such cases, the overloaded cloudlets may exceed the QoS latency target, but the under-loaded cloudlets will meet the QoS latency target, while all the cloudlets maximizing their utilities. The utility of each cloudlet includes the revenue earned for all the job requests received and the penalty for end-to-end latency. In Table 1, we provide a brief comparison of our current work with some of the existing works.

III. SYSTEM MODEL

A. FUNDAMENTAL SYSTEM DESIGN CONSIDERATIONS

Fig. 1 presents a schematic diagram that shows the interaction among the participants of our proposed load balancing game viz., mobile devices, competing cloudlets, and a mediator. Each mobile device offloads its job requests to the nearest cloudlet. As we consider a heterogeneous deployment scenario, the neighboring cloudlets may belong to same as well as different service providers. We consider that there are N competing cloudlets in the network, where $N \geq 2$, and the set of cloudlets is denoted by $\mathcal{C} = \{1, 2, \ldots, N\}$. A computational facility is installed by the mediator in the proximity of all the competing cloudlets to supervise the computation offloading game among them. The set of competing cloudlets $\mathcal{C}$ and their respective processing capacity are *common knowledge*, which implies that all the cloudlets and the mediator are aware of each others' presence [31].

We consider *quasi-static mobility* to model the mobility of mobile users and assume that a typical mobile user cannot move beyond the coverage area of a cloudlet within 1-10 ms. This implies that the mobile users can be considered to be almost stationary during computation offloading period to a cloudlet but may move later [16]. The cloudlets either start to process the received job requests or strategically offload a fraction of it to its neighboring cloudlets to meet the intended QoS latency target D_Q. Note that if any highly overloaded cloudlet cannot process some of its total incoming job requests within D_Q, it needs to drop those job request and pay a penalty for that. In our system model, we consider that each cloudlet first estimates their future incoming job requests of $(n+1)^{\text{th}}$ time-slot at n^{th} time-slot by using the information from $(n-1)^{\text{th}}$ time-slot. Although, job request arrival process is a non-stationary process, but it is *pseudo-stationary* in nature, because the mean arrival rate does not vary abruptly, rather gradually. Thus, a quick prediction of incoming job request arrival rate with 80-90% accuracy is possible [32]. Each cloudlet also estimates the transmission latency of the incoming job requests from the mobile devices and the intermediate transmission latencies with its neighboring cloudlets [33]. Each cloudlet periodically executes the learning algorithms at an interval of D_Q and sends this information to the mediator for computation of NE load balancing strategies. Nonetheless, the design of load-predictive algorithms is beyond the scope of this paper and hence, consider the algorithm in [33].

Each cloudlet has a finite (possibly different) number of processors. We further assume that each unit processor present in the network have similar job processing capability. If we assume that the i^{th} cloudlet has a single processor then the average service rate is μ_i (jobs/s) depending on the incoming job requests. In practice, all the competing cloudlets can have different values of μ_i because of the nature of the jobs that arrive at each of this cloudlet. Therefore μ_i indicate a parametric description of arrived jobs at the i^{th} cloudlet. We further assume that the cloudlets use virtual machines to process multiple job requests received from mobile devices in parallel. From the Google cluster-usage traces, it can be shown that

job request arrival times and their service times follow *exponential distributions*, and hence can be considered as *Poisson processes* [34]. Therefore, we model the cloudlets as $M/M/1$ queuing systems [25]. Note that, $M/M/1$ queue provides the upper bound of processing latency of a cloudlet when the aggregated processing rate of all the processors are considered, i.e., the worst-case processing latency of the cloudlets are considered. This ensures that when the average latency of each cloudlet meets the QoS latency target D_Q, then all the incoming job requests are processed within D_Q.

We consider that the average job request arrival rate from all the corresponding mobile devices to a cloudlet $i \in \mathcal{C}$ is λ_i. The individual job requests from mobile devices can be denoted by the total number of CPU cycles required to complete the job w, and the desired QoS latency target D_Q [16]. However, in this work, we are considering a batch of incoming job requests to the cloudlets rather than individual job requests. Thus, the computational and latency requirements of all the incoming job requests to i^{th} cloudlet are denoted by the consolidated tuple (μ_i, λ_i, D_Q). We assume that the network has sufficient bandwidth to accommodate all the job request packets. The job request arrival rate λ_i can be directly determined by counting the number of incoming packets over each time-slot. If the i^{th} cloudlet has n_i number of processors with complete parallel processing enabled, then the required service rate for supporting the total CPU cycles of all the job requests received within a time-slot rate can be determined as $\mu_{ii} = n_i\mu_i$. Moreover, when i^{th} cloudlet offloads some job requests to its neighboring j^{th} cloudlet, then the corresponding service rate for the respective job requests is defined as $\mu_{ij} = n_i\mu_j$. Note that, we consider a statistical average value of D_Q of all the incoming job requests to each cloudlet and hence, different cloudlets can have different D_Q values. However, through some internal job scheduling scheme, each cloudlet can prioritize the processing of some of the incoming job requests, based on their priority or urgency while meeting the average latency target D_Q over each time interval.

The average job request arrival rate varies over different time intervals, and hence, we assume that each λ_i is *independently and uniformly distributed* over the support $\Lambda_i = [0, \lambda_i^{max}], \forall i \in \mathcal{C}$. Therefore, the *computation load profile* of all the competing cloudlets is represented as $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_N) \in \boldsymbol{\Lambda} = (\Lambda_1 \times \Lambda_2 \times \cdots \times \Lambda_N)$. In a completely distributed computation offload scenario, neighboring cloudlets need to exchange various information about incoming job requests and their offload strategies to compute the NE. This introduces a huge number of control-packet exchanges among the cloudlets. This problem can be very easily solved if all the cloudlets periodically send the information about incoming job requests to the computational facility installed by the mediator. The mediator computes the NE strategies for computation offloading for all the cloudlets over that time-slot by using a centralized algorithm and broadcast to the competing cloudlets, as shown in Fig. 1.

As the incoming job requests to the cloudlets are randomly varying over time, the actual scenario can be thought of a *Bayesian game setup*, but a *Bayesian NE* of a game is much weaker than a *pure-strategy NE* [31]. Thus, if the competing cloudlets are regulated to reveal their computation load profile truthfully under the supervision of a mediator, computation of a strong pure-strategy NE becomes feasible. The competing cloudlets are non-cooperative and rational utility maximizers, but the mediator, on the other hand, does not have any utility associated with the incoming job requests. Hence, they can supervise a fair competition among the participating cloudlets. Note that, an *incentive compatible mechanism* can be used to ensure the elicitation of truthful information from the participating cloudlets, but we consider the mechanism design beyond the scope of this paper.

B. CONTROL DESIGN OF THE LOAD-BALANCING GAME

We show the overall *control mechanism of this game formulation* among neighboring competing cloudlets under the supervision of a mediator with the aid of a brief timing diagram in Fig. 1 and summarize the fundamental stages as follows:

(a) Each cloudlet periodically executes a *load-predictive learning algorithm* at every n^{th} time-slot by using the information from $(n-1)^{\text{th}}$ time-slot and the data available regarding the job arrival history to predict the incoming job request arrival rate of the $(n+1)^{\text{th}}$ time-slot.

(b) Along with job request arrival rate, each cloudlet also estimates the transmission latency of the incoming job requests from the mobile devices by using the given stochastic parameters (latency) of wireless and optical interfaces, as well as, estimates the intermediate transmission latencies with its neighboring cloudlets.

(c) After learning all this information, each cloudlet communicates the estimated incoming job request arrival rate and the transmission latencies to the computational facility installed by the mediator.

(d) The mediator computes the NE computation offloading strategy by employing a centralized algorithm and broadcasts to all the competing cloudlets before the $(n+1)^{\text{th}}$ time-slot.

(e) Accordingly, the cloudlets offload some fraction of their total incoming job requests to their neighboring cloudlets when the $(n+1)^{\text{th}}$ time-slot arrives.

IV. ECONOMIC AND NON-COOPERATIVE LOAD BALANCING GAME AMONG CLOUDLETS

In this section, we formulate the load balancing problem among $N \geq 2$ neighboring cloudlets from same as well as different service providers, under the supervision of a neutral mediator, as a continous-kernel non-cooperative game. This game theoretic model provides us rules/guidelines on how load balancing strategies should be determined, based on the communicated information.

A. LOAD BALANCING PROBLEM AMONG N ≥ 2 CLOUDLETS

In a practical deployment scenario, each cloudlet tries to minimize the end-to-end latency of the job requests received from the mobile devices, while maximizing their utilities. Thus, when overloaded cloudlets intend to offload a fraction of its job requests to its under-loaded neighboring cloudlets and the under-loaded cloudlets intend to receive some additional job requests from its overloaded neighboring cloudlets. The complete job request offloading strategy space of all cloudlets is defined as a matrix $\Phi = (\Phi_1^T, \Phi_2^T, \ldots, \Phi_N^T)^T \subset \mathbb{R}^{N\times N}$, where $\boldsymbol{\varphi}_i = (\varphi_{i1}, \varphi_{i2}, \ldots, \varphi_{iN}) \in \Phi_i \subset \mathbb{R}^N$, $\varphi_{ij} \in \Phi_{ij} = [0,1] \subset \mathbb{R}$, and $\sum_{j=1}^{N} \varphi_{ij} = 1, \forall i \in \mathcal{C}$. Each φ_{ij} denotes the fraction of job requests i^{th} cloudlet offloads to its j^{th} neighboring cloudlet. Due to the non-homogeneous service rates of the neighboring cloudlets, all the received job requests at each i^{th} cloudlet are served with different service rates, e.g., $\varphi_{ii}\lambda_i = (1 - \sum_{j\neq i}\varphi_{ij})\lambda_i$ jobs/s are served with service rate μ_{ii} job/s and $\sum_{j\neq i}\varphi_{ji}\lambda_j$ jobs/s are served with service rate μ_{ji} jobs/s. Therefore, to compute the overall processing and queuing latency of the received job requests at each cloudlet, we need to use a *multi-dimensional Markov chain* for $M/M/1$ queues [35]. For this analysis, we make the following assumptions:

- Each i^{th} cloudlet serves $(m_{i1}, m_{i2}, \ldots, m_{iN})$ job requests over each time-slot, where m_{ij} denotes the independent job requests received from j^{th} cloudlet.
- The detailed balance equations for each i^{th} cloudlet hold for all the pairs of adjacent states $(m_{i1}, \ldots, m_{ij}, \ldots, m_{iN})$ and $(m_{i1}, \ldots, m_{ij}+1, \ldots, m_{iN})$,

$$\varphi_{ji}\lambda_j \mathbb{P}_i(m_{i1}, \ldots, m_{ij}, \ldots, m_{iN}) = \mu_{ji}\mathbb{P}_i(m_{i1}, \ldots, m_{ij}+1, \ldots, m_{iN}), \forall i,j \in \mathcal{C}.$$

- The stationary state probability distribution of each i^{th} cloudlet can be expressed in the following *product form*,

$$\mathbb{P}_i(m_{i1}, m_{i2}, \ldots, m_{iN}) = \mathbb{P}_{i1}(m_{i1})\mathbb{P}_{i2}(m_{i2})\ldots\mathbb{P}_{iN}(m_{iN}).$$

- The number of job requests received from j^{th} neighboring cloudlet by each i^{th} cloudlet follows the *geometric distribution*, $\mathbb{P}_i(m_{ij}) = \rho_{ji}^{m_{ij}}(1-\rho_{ji})$, where $\rho_{ji} = \frac{\varphi_{ji}\lambda_j}{\mu_{ji}}$.

With the above mentioned assumptions, we derive the following closed form expression for *average number of job requests* served by each i^{th} cloudlet:

$$\mathbb{M}_i = \sum_{m_{i1}=0}^{\infty}\sum_{m_{i2}=0}^{\infty}\cdots\sum_{m_{iN}=0}^{\infty}\left\{(m_{i1}+\ldots+m_{iN})\prod_{j=1}^{N}\rho_{ji}^{m_{ij}}(1-\rho_{ji})\right\} = \frac{\sum_{j=1}^{N}\left\{\rho_{ji}\prod_{k=1,k\neq j}^{N}(1-\rho_{ki})\right\}}{\prod_{j=1}^{N}(1-\rho_{ji})}. \quad (1)$$

Therefore, by using *Little's theorem* [35] and (1), we get the overall processing and queuing latency of the job requests at i^{th} cloudlet as follows:

$$\mathbb{T}_i(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}) = \frac{1}{\varphi_{ii}\lambda_i + \sum_{j\neq i}\varphi_{ji}\lambda_j} \times \left(\frac{\sum_{j=1}^{N}\left\{\rho_{ji}\prod_{k=1,k\neq j}^{N}(1-\rho_{ki})\right\}}{\prod_{j=1}^{N}(1-\rho_{ij})}\right) = \frac{1}{\varphi_{ii}\lambda_i + \sum_{j\neq i}\varphi_{ji}\lambda_j} \times \left(\frac{\sum_{j=1}^{N}\left\{\varphi_{ji}\lambda_j\prod_{k=1,k\neq j}^{N}(\mu_{ki}-\varphi_{ki}\lambda_k)\right\}}{\prod_{j=1}^{N}(\mu_{ji}-\varphi_{ij}\lambda_j)}\right). \quad (2)$$

Nonetheless, in a stable market scenario, all the service providers tend to install cloudlets with similar processing capabilities (i.e., $n_i = n_j, \forall i,j$) to maintain the feasibility of the non-cooperative load balancing competition over the same customer base and the corresponding service rate request of arrived jobs at each cloudlet tends to become equal as they arrive from almost similar locality and customer base (i.e., $\mu_i = \mu_j, \forall i,j$). Thus, we consider that the processors of all the neighboring cloudlets have similar service rates for the incoming job requests from their associated mobile devices, i.e., $\mu_{ii} = \mu_{ji}$ and hence, the overall processing and queuing latency of the job requests at i^{th} cloudlet can be derived as follows:

$$\tilde{\mathbb{T}}_i(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}) = \frac{1}{\mu_{ii} - \left(1 - \sum_{j\neq i}\varphi_{ij}\right)\lambda_i - \sum_{j\neq i}\varphi_{ji}\lambda_j}. \quad (3)$$

With this, the average end-to-end latency of the job requests arriving at i^{th} cloudlet during each time interval can be computed as follows:

$$\mathcal{T}_i = t_{ui} + \left(1 - \sum_{j\neq i}\varphi_{ij}\right)\left[\frac{1}{\mu_{ii} - \left(1-\sum_{j\neq i}\varphi_{ij}\right)\lambda_i - \sum_{j\neq i}\varphi_{ji}\lambda_j}\right] + \sum_{j=1,j\neq i}^{N}\varphi_{ij}\left[t_{ij} + \frac{1}{\mu_{jj} - \varphi_{ij}\lambda_i - \sum_{k\neq i}\varphi_{kj}\lambda_k}\right], \quad (4)$$

where, t_{ui} denotes the average round-trip data transmission latency among mobile devices and the corresponding i^{th} cloudlet, and t_{ij} denotes the inter-cloudlet round-trip data transmission latency. We note that unlike cloud servers, cloudlets have finite computational resources and may fail to meet the QoS latency target for all the incoming job requests over a time interval. In that case, the cloudlets drop those job requests that are not processed within the QoS latency target D_Q. Therefore, as a performance measure, we find the probability that the incoming job requests at i^{th} cloudlet are dropped as,

$$P_{\geq D_Q} = \int_{D_Q}^{\infty}\frac{1}{\mathcal{T}_i}\exp\left\{-\frac{1}{\mathcal{T}_i}x\right\}dx = \exp\left\{-\frac{D_Q}{\mathcal{T}_i}\right\}. \quad (5)$$

Nonetheless, if we consider each cloudlet as an $M/M/1/K$ queueing system with a finite capacity for job requests over each timeslot, then we can find the equivalent capacity of

each cloudlet by equating (5) to its blocking probability as follows:

$$\frac{(\mu_{ii}-\lambda_i)\lambda_i^K}{\left(\mu_{ii}^{K+1}-\lambda_i^{K+1}\right)}=\exp\left\{-\frac{D_Q}{\mathcal{T}_i}\right\}. \tag{6}$$

B. ECONOMIC AND NON-COOPERATIVE GAME FORMULATION

In this paper, we consider the most commonly used pricing schemes, e.g., *pay-as-you-go* policy, where users pay a fixed price per job request without any long-term commitments [36]. Note that, each cloudlet receives a linearly proportional price per workload (Ω_1) for the total amount of incoming job requests from all the connected mobile devices. Each cloudlet pays a linearly proportional price per workload (Ω_2) for offloading job requests to a neighboring cloudlet from a different service provider and also, receives a linearly proportional price for executing its neighbor's offloaded jobs. The cloudlets can also cooperate or bargain among themselves to decide the value of Ω_2, but this leads to a cooperative or bargaining game-theoretic model, which is part of our future work. We define a parameter γ_{ij} to distinguish the price for offloading a job request to neighboring cloudlets as follows:

$$\gamma_{ij}=\begin{cases}1; & \text{if a cloudlet offloads job requests to a cloudlet from a different service provider}\\ 0; & \text{if a cloudlet offloads job requests to a cloudlet that shares the same service provider}\end{cases}$$

This implies that every i^{th} cloudlet needs to pay a price to j^{th} cloudlet for offloading any job requests only when it belongs to a different service provider, i.e., $\gamma_{ij}=1$. In addition to these, each cloudlet pays a penalty price with a proportionality cost factor (Ω_3) for exceeding the QoS target latency D_Q. In this work, we consider a linear penalty price similar to the linear latency cost designed in [25]. To capture the complete economic interaction among mobile devices and cloudlets in this game, we choose to define the complete *utility function* of each competing cloudlet $U_i^N(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_{-i}),\forall i\in\mathcal{C}$, where $\boldsymbol{\varphi}_{-i}=(\boldsymbol{\varphi}_1,\ldots,\boldsymbol{\varphi}_{i-1},\boldsymbol{\varphi}_{i+1},\ldots,\boldsymbol{\varphi}_N)$, as follows:

$$\begin{aligned}U_i^N(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_{-i})&=\Omega_1\frac{\lambda_i}{\mu_{ii}}+\Omega_2\sum_{j=1,j\neq i}^{N}\gamma_{ji}\varphi_{ji}\frac{\lambda_j}{\mu_{ii}}\\&-\Omega_2\sum_{j=1,j\neq i}^{N}\gamma_{ij}\varphi_{ij}\frac{\lambda_i}{\mu_{jj}}-\left\{\zeta\left[\frac{\varphi_{ii}\lambda_i+\sum_{j\neq i}\varphi_{ji}\lambda_j}{\mu_{ii}}\right]+\eta\right\}\\&-\Omega_3\frac{\lambda_i}{\mu_{ii}}\left\{t_{ui}+\varphi_{ii}\left[\frac{1}{\mu_{ii}-\varphi_{ii}\lambda_i-\sum_{j\neq i}\varphi_{ji}\lambda_j}\right]\right.\\&\left.+\sum_{j=1,j\neq i}^{N}\varphi_{ij}\left[t_{ij}+\frac{1}{\mu_{jj}-\varphi_{ij}\lambda_i-\sum_{k\neq i}\varphi_{kj}\lambda_k}\right]-D_Q\right\},\forall i\in\mathcal{C}.\end{aligned} \tag{7}$$

The first term in (7) denotes the total payment received by the cloudlet from mobile users and is linearly proportional to the average workload. The second term denotes the payment i^{th} cloudlet receives from j^{th} cloudlet to execute its offloaded job requests and the third term denotes the payment i^{th} cloudlet makes to j^{th} cloudlet for offloading job requests. Note that these terms are essential to distinguish payments for offloading job requests among heterogeneous cloudlets from same as well as different service providers. The fourth term denotes the cost of operation of i^{th} cloudlet, where ζ denotes cloudlet operation cost per unit processing rate and η denotes the default cost of cloudlet operation. The default cost of cloudlet operation arises due to the idle energy consumption of the cloudlet, which is nearly 70% of the maximum energy consumption by the cloudlet. Finally, the fifth term denotes the penalty of i^{th} cloudlet when the overall latency (sum of transmission, processing, and queueing latencies) of all the job requests received from the associated mobile devices exceeds the QoS target latency D_Q. We consider that the competing cloudlets are *individually rational* and impose the following constraint such that the utility of each cloudlet is more than or equal to their default utility without offloading any job requests, i.e.,

$$\begin{aligned}U_i^N(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_{-i})\geq U_i^0=\Omega_1\frac{\lambda_i}{\mu_{ii}}-\left\{\zeta\left[\frac{\lambda_i}{\mu_{ii}}\right]+\eta\right\}\\-\Omega_3\frac{\lambda_i}{\mu_{ii}}\left\{t_{ui}+\frac{1}{\mu_{ii}-\lambda_i}-D_Q\right\},\forall i\in\mathcal{C}.\end{aligned} \tag{8}$$

Clearly, none of the cloudlets try to offload any job requests to their neighboring cloudlets until they are overloaded. The above constraints defined in (8) are called *jointly shared constraints* as they impact upon all the competing cloudlets. It can be shown that these constraints are *jointly concave* in nature because $U_i^N(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_{-i}),\forall i\in\mathcal{C}$ are concave functions of $\boldsymbol{\varphi}$. As the average incoming job request arrival rate λ_i is a continuous parameter and $\boldsymbol{\varphi}_i$ is a vector of continuous variables, hence the utility function defined in (7) is also continuous and is expected to provide finite utilities to both the cloudlets, as long as the QoS target latency is met. We consider the utility function only under the *condition of stable operation*, i.e., $[\mu_{ii}-(1-\sum_{j\neq i}\varphi_{ij})\lambda_i-\sum_{j\neq i}\varphi_{ji}\lambda_j]\geq 0,\forall i,j\neq i\in\mathcal{C}$, and the following constraint is also applied:

$$0\leq\varphi_{ij}\leq 1,\sum_{j=1}^{N}\varphi_{ij}=1,\forall i\in\mathcal{C}. \tag{9}$$

In general, all the competing cloudlets are *risk neutral* and intend to solve the maximization problem summarized as follows:

$$\begin{aligned}\boldsymbol{P}:\quad&\max_{\boldsymbol{\varphi}_i\in\Phi_i}\quad U_i^N(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_{-i})\\&\text{subject to}\ \ 0\leq\varphi_{ij}\leq 1,\ \sum_{j=1}^{N}\varphi_{ij}=1,\\&\qquad\qquad U_i^N(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_{-i})\geq U_i^0.\end{aligned}$$

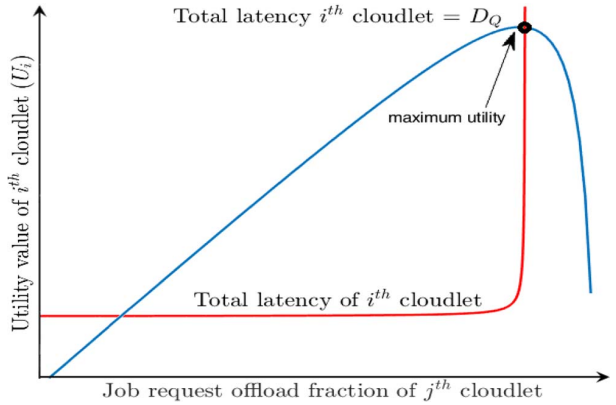


FIGURE 2. A sample utility function of i^{th} cloudlet against job request offload fraction of neighboring j^{th} cloudlet.

It is interesting to note that in this load balancing game, each competing cloudlet is interested in *maximizing their individual utilities rather than strictly minimizing the average end-to-end latency* as most of the existing works. Hence, the cloudlets are always interested in receiving some job requests from neighboring cloudlets as long as the QoS latency requirement D_Q is met and some extra incentive is gained. Fig. 2 shows that with a sufficient amount of job requests and a set of properly chosen parameters Ω_1, Ω_2, and Ω_3, the utility function i^{th} cloudlet monotonically increases as more job requests are offloaded by the neighboring j^{th} cloudlet until the total end-to-end latency is equal to the target QoS latency value D_Q. The maximum utility is achieved at the point where the total end-to-end latency is equal to D_Q and the utility starts to decrease beyond this point. Therefore, the fraction of incoming job requests offloaded by an overloaded cloudlet is controlled by the overloaded cloudlet itself as well as its under-loaded neighboring cloudlets. Hence, it is essential to include all the latency terms in the utility function (7). Furthermore, note that due to the utility function (7) and constraints (9)-(8), which does not provide an explicit latency bound on the participating cloudlets, even highly over-loaded cloudlets can participate in the game and can offload some of the job requests to the relatively under-loaded neighboring cloudlets. This makes their utility higher than the utility by not participating in the game. Nonetheless, under such conditions the game formulation in [25] that has explicit delay bound on participating cloudlets becomes infeasible and hence, a valid NE solution can not be obtained. Now, we analyze the existence and uniqueness criteria of our proposed game formulation in the following subsection.

C. EXISTENCE AND UNIQUENESS OF THE NASH EQUILIBRIUM

We specify a *non-cooperative load balancing game* $\Gamma = \langle \mathcal{C}, (\Theta_i)_{i\in\mathcal{C}}, (U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}))_{i\in\mathcal{C}} \rangle$, among competing cloudlets as a *tuple* consisting of the members of the set of competing cloudlets $\mathcal{C}$, the strategy space of each cloudlet is $\Theta_i = \{\varphi_i \in \Phi_i : U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}) \geq U_i^0, \forall i \in \mathcal{C}\}$, and the utility functions of all cloudlets $U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}), \forall i \in \mathcal{C}$. With the given game form Γ, we prefer to investigate the NE of the problem, because none of the competing cloudlets find it beneficial to deviate unilaterally from the NE computational offload strategy set $\boldsymbol{\varphi}^* = (\boldsymbol{\varphi}_1^{*T}, \boldsymbol{\varphi}_2^{*T}, \ldots, \boldsymbol{\varphi}_N^{*T})$. In other words, with the above game formulation, each competing cloudlet has to maximize the utility function (7) subject to constraints (9)-(8). Thus, not only the utility functions of all competing cloudlets are coupled like in a usual NE problem, but the strategy sets of all competing cloudlets are also coupled. Hence, due to the presence of the *jointly shared constraints* (8), we classify this problem as a *GNE problem* [37]. In other words, the complete solution space Θ of the game Γ cannot be constructed as a direct Cartesian product of the solution space of the individual competing cloudlets, but we also need to apply the shared constraints on the Cartesian product of the solution space of the individual competing cloudlets. In general, the GNE problems are very difficult to solve and almost intractable in most of the cases as they present severe analytical difficulties [38]. However, a special class of GNE problems can be solved by mapping the problem to a *variational inequality* (VI) and such equilibria are called *variational equilibria* [39]. Note that, VI is a well-known technique used to solve a broader class of convex optimization problems and we recall the definition as: given a closed and convex subset $\mathcal{K} \subseteq \mathbb{R}^n$ and a vector-valued function $F : \mathcal{K} \to \mathbb{R}^n$, the VI problem, denoted $\mathbf{VI}(\mathcal{K}, F)$, consists in finding a vector $\boldsymbol{x}^* \in \mathcal{K}$ (called a solution of the VI) such that:

$$\big(\boldsymbol{y} - \boldsymbol{x}^*\big)^T F\big(\boldsymbol{x}^*\big) \geq 0, \forall \boldsymbol{y} \in \mathcal{K}. \tag{17}$$

The following theorems show that our non-cooperative game formulation also falls under the special class of GNE problems that can be solved using VI. Moreover, the solution of the GNE is *unique when it exists*.

Theorem 1: The game $\Gamma = \langle \mathcal{C}, (\Theta_i)_{i\in\mathcal{C}}, (U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}))_{i\in\mathcal{C}} \rangle$ is equivalent to $\mathbf{VI}(\Psi, F)$, where $F \coloneqq \nabla_{\boldsymbol{\varphi}_i}(U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}))_{i\in\mathcal{C}}$.

This theorem shows that our current game formulation can be successfully mapped to an equivalent VI problem by using a few fundamental characteristics. Please see Appendix A for a detailed proof.

Theorem 2: The game $\Gamma = \langle \mathcal{C}, (\Theta_i)_{i\in\mathcal{C}}, (U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}))_{i\in\mathcal{C}} \rangle$ represented by $\mathbf{VI}(\Psi, F)$, where $F \coloneqq \nabla_{\boldsymbol{\varphi}_i}(U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}))_{i\in\mathcal{C}}$ admits a unique pure-strategy NE.

A NE (in pure strategies) of the game Γ can be defined as a strategy profile $\boldsymbol{\varphi}^*$ that satisfies, $U_i^N(\boldsymbol{\varphi}_i^*, \boldsymbol{\varphi}_{-i}^*) \geq U_i^N(\boldsymbol{\varphi}_i', \boldsymbol{\varphi}_{-i}^*), \forall \boldsymbol{\varphi}_i' \in \Theta_i$ and $\forall i \in \mathcal{C}$. Moreover, this NE can also be interpreted as the *intersection* or the *fixed point* of the *best-response functions* of the competing cloudlets [31]. Please see Appendix B for a detailed proof.

D. EFFICIENCY OF NE OF THE LOAD BALANCING GAME

It is a general notion that the NE of non-cooperative games yields sub-optimal utilities for the players. Therefore, we

need to analyze the NE of our proposed load balancing game among cloudlets corresponding to the best case and the worst case. For the proposed load balancing game Γ, the *price of anarchy (PoA)* is defined as the ratio of the sum of the utility functions of all players at the worst case NE and at the social optimum solution [31], as follows:

$$\text{PoA}(\Gamma) = \frac{\min_{\boldsymbol{\varphi}\in\Phi^{NE}} \sum_{i=1}^{N} U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i})}{\max_{\boldsymbol{\varphi}\in\Phi} \sum_{i=1}^{N} U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i})} \tag{18}$$

where Φ^{NE} denotes the set of all possible NE solutions. Again, the *price of stability (PoS)* of the load balancing game Γ is defined as the ratio of the sum of the utility functions of all players at the best case NE and at the social optimum solution [31], as follows:

$$\text{PoS}(\Gamma) = \frac{\max_{\boldsymbol{\varphi}\in\Phi^{NE}} \sum_{i=1}^{N} U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i})}{\max_{\boldsymbol{\varphi}\in\Phi} \sum_{i=1}^{N} U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i})} \tag{19}$$

Note that the values of PoA and PoS are different in general, but $\text{PoA}(\Gamma) = \text{PoS}(\Gamma)$ for the load balancing game Γ as there exists a unique NE. Furthermore, we can derive bounds on PoA by exploiting the smoothness of our game formulation. We note that $\boldsymbol{\varphi}^*$ is a NE and $\boldsymbol{\varphi}'_i \in \Phi_i$. Hence, we can write $U_i^N(\boldsymbol{\varphi}_i^*, \boldsymbol{\varphi}_{-i}^*) \geq U_i^N(\boldsymbol{\varphi}'_i, \boldsymbol{\varphi}_{-i}^*), \forall i \in \mathcal{C}$ as well as,

$$\begin{aligned}\sum_{i=1}^{N} U_i^N(\boldsymbol{\varphi}_i^*, \boldsymbol{\varphi}_{-i}^*) &\geq \sum_{i=1}^{N} U_i^N(\boldsymbol{\varphi}'_i, \boldsymbol{\varphi}_{-i}^*) \\ &\geq \nu \sum_{i=1}^{N} U_i^N(\boldsymbol{\varphi}'_i, \boldsymbol{\varphi}'_{-i}) - \delta \sum_{i=1}^{N} U_i^N(\boldsymbol{\varphi}_i^*, \boldsymbol{\varphi}_{-i}^*),\end{aligned} \tag{20}$$

where, $\nu > 0$ and $\delta \geq 1$. The inequality (20) holds as the utility functions are smooth and concave functions defined over a compact and convex set [40]. Now, considering $\boldsymbol{\varphi}'$ as the social optimal solution, we derive the following bound on PoA:

$$\frac{\nu}{1+\delta} \leq \text{PoA}(\Gamma) \leq 1, \tag{21}$$

where the best-possible bound can be derived from $\sup\{\frac{\nu}{1+\delta}\}$ such that inequality (20) is satisfied.

V. ALGORITHM FOR COMPUTATION OF GNE

Based on the computation load profile received from competing cloudlets, the computational facility installed by the mediator can centrally compute the NE of the game Γ. Assuming that $\boldsymbol{\varphi}^*$ is a solution of the GNE problem, if for a competing cloudlet $i \in \mathcal{C}$, a suitable constraint qualification holds, then there exist vectors of Lagrange multipliers, e.g., $\boldsymbol{\alpha}_i \in \mathbb{R}^{N-1}$, $\boldsymbol{\beta}_i \in \mathbb{R}^{N-1}$, and $\boldsymbol{\xi}_i \in \mathbb{R}^N$ so that the classical KKT conditions are satisfied as below:

$$\nabla_{\boldsymbol{\varphi}_i} U_i^N + \nabla_{\boldsymbol{\varphi}_i}\left(\boldsymbol{\alpha}_i^T \boldsymbol{\varphi}_i + \boldsymbol{\beta}_i^T(1-\boldsymbol{\varphi}_i) + \boldsymbol{\xi}_i^T\left(U_i^N - U_i^0\right)_{i\in\mathcal{C}}\right) = 0, \tag{22}$$

$$\boldsymbol{\alpha}_i^T \boldsymbol{\varphi}_i = 0, \tag{23}$$

$$\boldsymbol{\beta}_i^T(\mathbf{1} - \boldsymbol{\varphi}_i) = 0, \tag{24}$$

Algorithm 1 Projection Algorithm With Constant Step Size

1: **Initialization:** Choose any Lagrange multipliers $\boldsymbol{\alpha}^0$, $\boldsymbol{\beta}^0$, $\boldsymbol{\xi}^0 \geq 0$, step size $\omega > 0$, and tolerance limit $\epsilon > 0$. Set the index $t = 0$.
2: **Output:** NE of the computation offload game $\boldsymbol{\varphi}^*$.
3: If $\boldsymbol{\alpha}^t$, $\boldsymbol{\beta}^t$, and $\boldsymbol{\xi}^t$ satisfies a desirable tolerance limit: STOP.
4: Given $\boldsymbol{\alpha}^t$, $\boldsymbol{\beta}^t$, and $\boldsymbol{\xi}^t$, compute $\boldsymbol{\varphi}^t(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^t, \boldsymbol{\xi}^t)$ as the NE solution of the GNE problem (22)-(25) with fixed Lagrange multipliers $\boldsymbol{\alpha} = \boldsymbol{\alpha}^t$, $\boldsymbol{\beta} = \boldsymbol{\beta}^t$, and $\boldsymbol{\xi} = \boldsymbol{\xi}^t$;
5: Update the Lagrange multipliers: for all $i, j \in \mathcal{C}$, compute

$$\begin{aligned}\alpha_{ij}^{t+1} &= [\alpha_{ij}^t - \omega(\varphi_{ij})]^+, \forall i \neq j, \\ \beta_{ij}^{t+1} &= [\beta_{ij}^t - \omega(1-\varphi_{ij})]^+, \forall i \neq j, \\ \xi_{ij}^{t+1} &= [\xi_{ij}^t - \omega(U_i^N - U_i^0)]^+, \forall i, j,\end{aligned}$$

where $[z]^+ = \max\{0, z\}$.
6: Set $t \leftarrow t+1$; go to Step 3.

$$\boldsymbol{\xi}_i^T\left(U_i^N - U_i^0\right)_{i\in\mathcal{C}} = 0. \tag{25}$$

Now, we stack all the KKT conditions of all the competing cloudlets to formulate a VI corresponding to our current game formulation. The solution of this VI problem is the NE of our currently formulated game.

Theorem 3: A solution $\boldsymbol{\varphi}^*$ is a variational equilibrium of the game (22)-(25) if and only if Lagrange multipliers $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\xi}^*$ exists such that $(\boldsymbol{\varphi}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\xi}^*)$ is a solution of $\mathbf{VI}(\Psi, F)$.

Please see Appendix C for a detailed proof. By using the above theorem, we design a *gradient projection algorithm* with constant step size [39]. The convergence rate of such an algorithm is greatly dependent on the desired tolerance limit ϵ we want to achieve, and the number of iterations required for achieving $\|\boldsymbol{\varphi}^t - \boldsymbol{\varphi}^*\| \leq \epsilon$ is $t \geq \bar{t}$ with

$$\bar{t} = \log\left(\frac{\epsilon(1-\|\boldsymbol{\Gamma}_F\|)}{\|\boldsymbol{\varphi}^{(1)} - \boldsymbol{\varphi}^{(0)}\|}\right) / \log(\|\boldsymbol{\Gamma}_F\|), \tag{26}$$

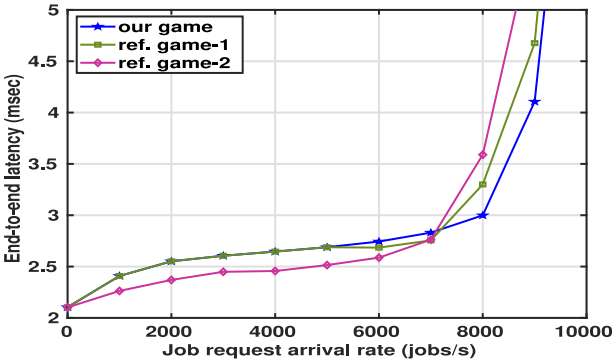
where $\boldsymbol{\varphi}^*$ is the unique NE of the game Γ and $\|\boldsymbol{\Gamma}_F\| < 1$ is the *best-response contraction constant*, as defined in [39]. If the network consists of a large number of cloudlets, to prevent an exponential increase in the strategy space of each cloudlet, we can discourage the cloudlets to offload job requests to far-away cloudlets. This can be done by observing that the intermediate transmission latency among neighboring cloudlets t_{ij} is greater than some threshold latency of choice. As the first-order expressions (11), as shown at the bottom of the next page, (12) and (13), as shown at the bottom of p. 11, create a system of nonlinear implicit equations of $\boldsymbol{\varphi}$, we use *Newton-Raphson method* [41] to compute $\boldsymbol{\varphi}^t$ numerically. The steps of the algorithm are summarized in Algorithm 1. Note that, the *quasi-linear utility functions* of each competing cloudlets (10), as shown at the bottom of

the next page almost linearly increase as long as the overall latency is below the target QoS latency, but as soon as the cloudlets are overloaded, their utility starts to decrease due to the sharp non-linear increase of latency penalty. Therefore, overloaded cloudlets always try to offload to their under-loaded neighboring cloudlets. On the other hand, during very low load condition, cloudlets either do not prefer to offload or offload in a very small amount.

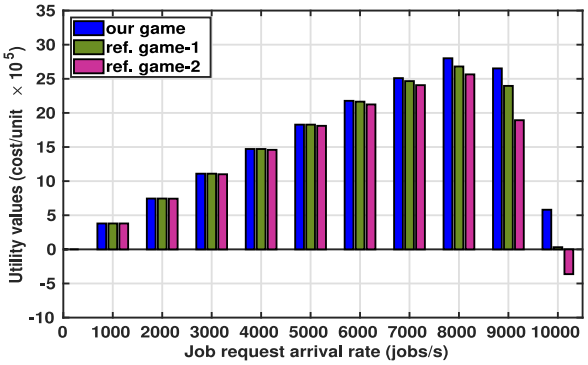
VI. RESULTS AND DISCUSSIONS

In this section, we investigate various behavioural aspects of the proposed load balancing strategy through numerical evaluations. For this purpose, we consider a set of 10 neighboring cloudlets from same as well as different service providers. In this work, we consider average processing rate μ_{ii} varies within 10000-15000 jobs/s and incoming job request to each cloudlet λ_i varies within 0-15000 jobs/s. The QoS latency target considered is $D_Q = 10$ ms. We consider the average value of t_{ui} between mobile devices and cloudlets as 2 msec. The intermediate transmission latency among neighboring cloudlets t_{ij} varies within 0.5-1 msec. In actual practice, sometimes the proper price factors are also determined by applying the *multiple criteria decision-making theory* [42]. In this work, we arbitrarily choose normalized values of $\Omega_1 = 5 \times 10^2$, $\Omega_2 = 1 \times 10^6$, $\Omega_3 = 5 \times 10^8$, $\zeta = 300$, and $\eta = 700$ just to illustrate various properties of this game-theoretic computation offloading framework. Moreover, for our VI based algorithm to compute NE of the load balancing game, we choose a step size $\omega = 0.1$, and a tolerance limit $\epsilon = 10^{-4}$.

Fig. 3a shows a comparison among average end-to-end latency performance of all the participating cloudlets against job request arrival rate with our currently proposed game and games proposed in [25] (labelled as "ref. game-1") and [24] (labelled as "ref. game-2"), respectively. In this case, we consider a *high variance in job request arrival rates among under and overloaded cloudlets* (within 0-4000 jobs/s) and the service rates of all the cloudlets are $\mu_{ii} = 10000$ jobs/s. We see that when the load condition is low or moderate, the ref. game-1 performs best because it always tries



(a) μ_{ii} = 10000 jobs/s and λ_i variance = 0-4000 jobs/s, $\forall i \in C$.



(b) μ_{ii} = 10000 jobs/s and λ_i variance = 0-4000 jobs/s, $\forall i \in C$.

FIGURE 3. Comparison of end-to-end latency and utility values of competing cloudlets with high variance (0-4000 jobs/s) in incoming job request arrival rates among neighboring cloudlets with our proposed game and other games proposed in [24], [25].

$$U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}) = \Omega_1 \frac{\lambda_i}{\mu_{ii}} + \Omega_2 \sum_{j=1, j\neq i}^{N} \gamma_{ji}\varphi_{ji} \frac{\lambda_j}{\mu_{ii}} - \Omega_2 \sum_{j=1, j\neq i}^{N} \gamma_{ij}\varphi_{ij} \frac{\lambda_i}{\mu_{jj}} - \left\{ \zeta \left[\frac{\left(1 - \sum_{j\neq i} \varphi_{ij}\right)\lambda_i + \sum_{j\neq i} \varphi_{ji}\lambda_j}{\mu_{ii}} \right] + \eta \right\}$$
$$- \Omega_3 \frac{\lambda_i}{\mu_{ii}} \left\{ t_{ui} + \left(1 - \sum_{j=1, j\neq i}^{N} \varphi_{ij}\right) \left[\frac{1}{\left(\mu_{ii} - \left(1 - \sum_{j\neq i} \varphi_{ij}\right)\lambda_i - \sum_{j\neq i} \varphi_{ji}\lambda_j\right)} \right] \right.$$
$$\left. + \sum_{j=1, j\neq i}^{N} \varphi_{ij} \left[t_{ij} + \frac{1}{\left(\mu_{jj} - \varphi_{ij}\lambda_i - \sum_{k\neq i} \varphi_{kj}\lambda_k\right)} \right] - D_Q \right\}, \forall i \in \mathcal{C} \tag{10}$$

$$\frac{\partial U_i^N}{\partial \varphi_{ij}} = -\Omega_2 \gamma_{ij} \frac{\lambda_i}{\mu_{jj}} + \zeta \frac{\lambda_i}{\mu_{ii}} - \Omega_3 \frac{\lambda_i}{\mu_{ii}} \left\{ - \frac{\mu_{ii} - \sum_{j\neq i} \varphi_{ji}\lambda_j}{\left(\mu_{ii} - \left(1 - \varphi_{ij} - \sum_{k\neq i,j} \varphi_{ik}\right)\lambda_i - \sum_{j\neq i} \varphi_{ji}\lambda_j\right)^2} + t_{ij} \right.$$
$$\left. + \frac{\mu_{jj} - \left(1 - \sum_{k\neq j} \varphi_{jk}\right)\lambda_j - \sum_{k\neq i,j} \varphi_{kj}\lambda_k}{\left(\mu_{jj} - \left(1 - \sum_{k\neq j} \varphi_{jk}\right)\lambda_j - \varphi_{ij}\lambda_i - \sum_{k\neq i,j} \varphi_{kj}\lambda_k\right)^2} \right\}, \forall i, j \in \mathcal{C} \tag{11}$$

to minimize the end-to-end latency. Under such conditions, both our proposed game and ref. game-2 performs poorer because these models do not allow the cloudlets to offload anything. However, after reaching a certain threshold in incoming job requests, ref. game-2 allows the cloudlets to offload job requests and their latency performance starts to improve. Nonetheless, when all the cloudlets become sufficiently overloaded, our game allows the cloudlets to offload and performs relatively better than both ref. game-1 and ref. game-2. This happens because, in our game it is ensured that all the under-loaded cloudlets operate within the QoS target latency bound, D_Q and over-loaded cloudlets may exceed D_Q, but they are allowed to offload to the maximum extent possible. Whereas, ref. game-1 becomes infeasible in high load condition as their explicit latency constraints start to violate and ref. game-2 tends to overload the under-loaded cloudlets by offloading job requests in an uncontrolled way to them.

Next, Fig. 3b shows a comparison among average utility values of all the participating cloudlets against job request arrival rate with our game, ref. game-1, and ref. game-2. It is clear that under all the network condition, the average economic utility values of the cloudlets with our game is relatively better than both ref. game-1, and ref. game-2. This primarily happens due to the typical characteristics of our utility function definition as shown in Fig. 2. With our game formulation, cloudlets do not offload until they are sufficiently overloaded and the under-loaded cloudlets receive job requests until their end-to-end latency reaches D_Q, where the maximum utility is achieved. Hence, the mutual incentive payment for computation offloading is not required. When the cloudlets are sufficiently overloaded, they start to offload so that the latency penalty does not become high enough to reduce the actual utility U_i^N below their default utility U_i^0.

Similarly, Fig. 4a shows a comparison among average end-to-end latency performance and Fig. 4b shows a comparison among average utility values of all the participating cloudlets against job request arrival rate with our game, ref. game-1, and ref. game-2. Nonetheless, in this case, we consider a *moderate variance in job request arrival rates among under and overloaded cloudlets* (within 0-2000 jobs/s) and the service rates of all the cloudlets are $\mu_{ii} = 10000$ jobs/s. Note that both the plots show similar behavior as in Fig. 3a and Fig. 3b. However, as the difference in job request arrival rates among under and overloaded cloudlets is lesser than the previous case, the room for offloading job requests by overloaded cloudlets to under-loaded cloudlets is also lesser. Hence, the average end-to-end latency starts to overshoot much earlier and the average utility gained is also less.

Again, we present a comparison among average end-to-end latency performance in Fig. 5a and comparison among average utility values of all the participating cloudlets in Fig. 5b with our game, ref. game-1, and ref. game-2. In this case, we keep the job request arrival rates of all the cloudlets equal but vary their service rates. Thus, to calculate the processing latency of each cloudlet, we need to use (2) instead of (3) and this creates a very general load balancing scenario. At first, we consider a high variance (10000-15000 jobs/s) of service rates among neighboring cloudlets and observe similar patterns of the graphs as before, but the end-to-end latency and utility values are relatively better as the service rates of some cloudlets are much higher than the average job request arrival rate. We present similar graphs in Fig. 6a and in Fig. 6b, but we consider

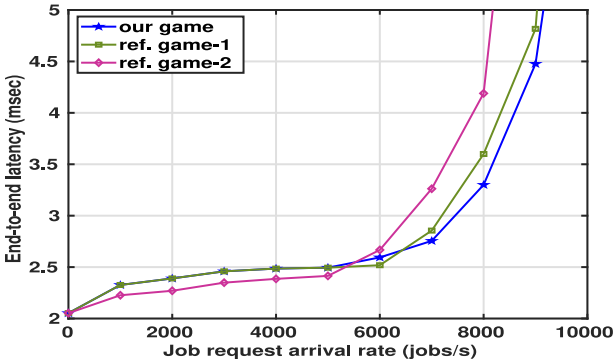
$$\frac{\partial U_i^N}{\partial \varphi_{ji}} = \Omega_2 \gamma_{ji} \frac{\lambda_j}{\mu_{ii}} - \zeta \frac{\lambda_j}{\mu_{ii}} - \Omega_3 \frac{\lambda_i}{\mu_{ii}} \left\{ \frac{\left(1 - \sum_{j \neq i} \varphi_{ij}\right)\lambda_j}{\left(\mu_{ii} - \left(1 - \sum_{j \neq i} \varphi_{ij}\right)\lambda_i - \sum_{j \neq i} \varphi_{ji}\lambda_j\right)^2} - \frac{\varphi_{ij}\lambda_j}{\left(\mu_{jj} - \left(1 - \varphi_{ji} - \sum_{k \neq i,j} \varphi_{jk}\right)\lambda_j - \sum_{k \neq j} \varphi_{kj}\lambda_k\right)^2} \right\}, \quad \forall i, j \in \mathcal{C} \tag{12}$$

$$\frac{\partial U_i^N}{\partial \varphi_{jk}} = \Omega_3 \frac{\lambda_i}{\mu_{ii}} \left\{ \frac{\varphi_{ij}\lambda_j}{\left(\mu_{jj} - \left(1 - \varphi_{ji} - \sum_{k \neq i,j} \varphi_{jk}\right)\lambda_j - \sum_{k \neq j} \varphi_{kj}\lambda_k\right)^2} \right\}, \forall i, j, k \in \mathcal{C} \tag{13}$$

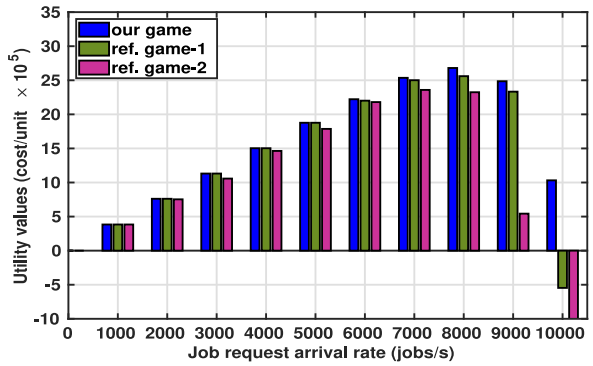
$$\frac{\partial^2 U_i^N}{\partial \varphi_{ij}^2} = -2\Omega_3 \frac{\lambda_i^2}{\mu_{ii}} \left\{ \frac{\mu_{ii} - \sum_{j \neq i} \varphi_{ji}\lambda_j}{\left(\mu_{ii} - \left(1 - \varphi_{ij} - \sum_{k \neq i,j} \varphi_{ik}\right)\lambda_i - \sum_{j \neq i} \varphi_{ji}\lambda_j\right)^3} + \frac{\mu_{jj} - \left(1 - \sum_{k \neq j} \varphi_{jk}\right)\lambda_j - \sum_{k \neq i,j} \varphi_{kj}\lambda_k}{\left(\mu_{jj} - \left(1 - \sum_{k \neq j} \varphi_{jk}\right)\lambda_j - \varphi_{ij}\lambda_i - \sum_{k \neq i,j} \varphi_{kj}\lambda_k\right)^3} \right\} < 0, \quad \forall i, j \in \mathcal{C} \tag{14}$$

$$\frac{\partial^2 U_i^N}{\partial \varphi_{ij} \partial \varphi_{ik}} = -2\Omega_3 \frac{\lambda_i^2}{\mu_{ii}} \left\{ \frac{\mu_{ii} - \sum_{j \neq i} \varphi_{ji}\lambda_j}{\left(\mu_{ii} - \left(1 - \varphi_{ij} - \sum_{k \neq i,j} \varphi_{ik}\right)\lambda_i - \sum_{j \neq i} \varphi_{ji}\lambda_j\right)^3} \right\} < 0, \forall i, j, k \in \mathcal{C} \tag{15}$$

$$\frac{\partial^2 U_i^N}{\partial \varphi_{ji}^2} = -2\Omega_3 \frac{\lambda_i \lambda_j}{\mu_{ii}} \left\{ \frac{\left(1 - \sum_{j \neq i} \varphi_{ij}\right)\lambda_j}{\left(\mu_{ii} - \left(1 - \sum_{j \neq i} \varphi_{ij}\right)\lambda_i - \sum_{j \neq i} \varphi_{ji}\lambda_j\right)^3} + \frac{\varphi_{ij}\lambda_j}{\left(\mu_{jj} - \left(1 - \varphi_{ji} - \sum_{k \neq i,j} \varphi_{jk}\right)\lambda_j - \sum_{k \neq j} \varphi_{kj}\lambda_k\right)^3} \right\} < 0, \forall i, j \in \mathcal{C} \tag{16}$$

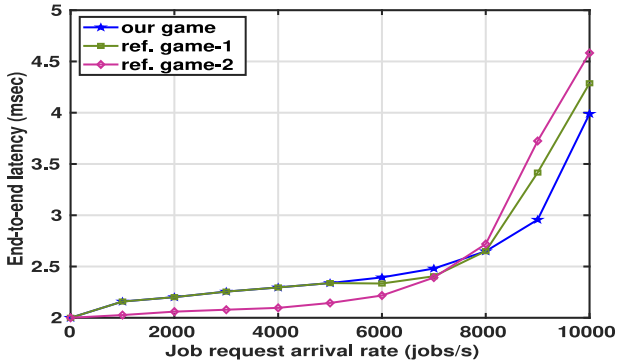


(a) μ_{ii} = 10000 jobs/s and λ_i variance = 0-2000 jobs/s, $\forall i \in C$.

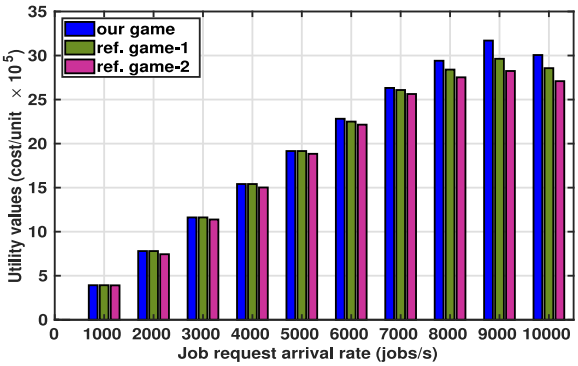


(b) μ_{ii} = 10000 jobs/s and λ_i variance = 0-2000 jobs/s, $\forall i \in C$.

FIGURE 4. Comparison of end-to-end latency and utility values of competing cloudlets with moderate variance (0-2000 jobs/s) in incoming job request arrival rates among neighboring cloudlets with our proposed game and other games proposed in [24], [25].



(a) μ_{ii} variance = 10000-15000 jobs/s and λ_i is same $\forall i \in C$.



(b) μ_{ii} variance = 10000-15000 jobs/s and λ_i is same $\forall i \in C$.

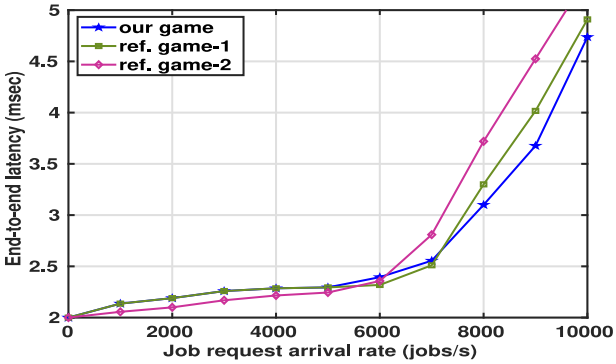
FIGURE 5. Comparison of end-to-end latency and utility values of competing cloudlets with high variance (10000-15000 jobs/s) in service rates among neighboring cloudlets and same job request arrival rates with our proposed game and other games proposed in [24], [25].

a moderate variance (10000-12000 jobs/s) of service rates among neighboring cloudlets. As a consequence, the latency and utility performance is slightly poorer than the previous graphs as the under-loaded cloudlets have lesser room to receive job requests from overloaded cloudlets.

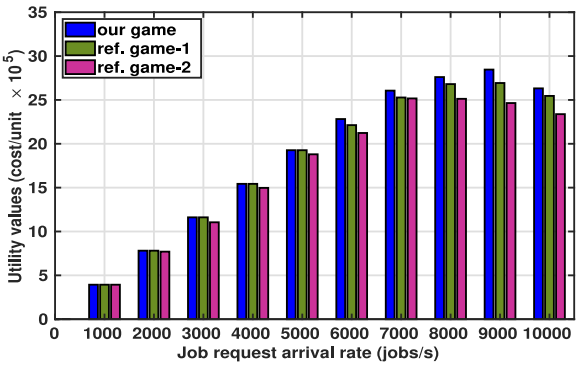
To further understand the efficiency of the NE of our proposed load balancing game, we provide another comparative result against the global optimal solution in Fig. 7. Again we consider a high variance in job request arrival rates among under and overloaded cloudlets (within 0-4000 jobs/s) and the service rates of all the cloudlets are $\mu_{ii} = 10000$ jobs/s. This figure shows that average utility values of the cloudlets are almost same with both NE and global optimal load balancing strategies under low and high load conditions. During such cases, neighboring cloudlets do not offload any job requests to each other. However, under moderate load conditions, when overloaded cloudlets can offload some job requests to their under-loaded neighbors, the average utility of the cloudlets are slightly better with global optimal solution.

Fig. 8 shows the convergence rate of Algorithm 1 for three neighboring cloudlets such that the figure is not unnecessarily overcrowded. We consider an average processing rate $\mu_{ii} = 10000$ jobs/s for each cloudlet and the respective incoming job requests are $\lambda_1 = 9500$ jobs/s, $\lambda_2 = 9200$ jobs/s, and $\lambda_3 = 5000$ jobs/s. In this case, both Cloudlet-1 and Cloudlet-2 are overloaded and hence, offloads certain fraction of their total incoming job requests to the under-loaded Cloudlet-3. As Cloudlet-1 is the most overloaded, the computation offload fraction of Cloudlet-1 is more than that of Cloudlet-2.

Furthermore, in Fig. 9, we show the impact of prediction accuracy of the incoming job request arrival rates on the NE utility values of the competing cloudlets. For the job request arrival rate prediction, we use the moving-average method based ARMA algorithm proposed in [33] and consider that job request arrival rates to each cloudlet remains stationary for 30 seconds. This algorithm works very efficiently when the incoming job requests are self-similar in nature and varies gradually over time. Again, we consider three cloudlets with $\mu_{ii} = 10000$ jobs/s and λ_i varying within 0-10000 jobs/s. Therefore, from the plot we observe that whenever the actual job request arrival rates of the cloudlets change, the NE utility values of the cloudlets based on the predicted job request arrival rate are slightly erroneous. However, within a few time-slots, each cloudlet is able to accurately predict the actual job request arrival rate and hence, the NE utilities of cloudlets with predicted job request arrival rates match to the NE utilities with actual job request arrival rates.



(a) μ_{ii} variance = 10000-12000 jobs/s and λ_i is same $\forall i \in C$.



(b) μ_{ii} variance = 10000-12000 jobs/s and λ_i is same $\forall i \in C$.

FIGURE 6. Comparison of end-to-end latency and utility values of competing cloudlets with moderate variance (10000-12000 jobs/s) in service rates among neighboring cloudlets and same job request arrival rates with our proposed game and other games in [24], [25].

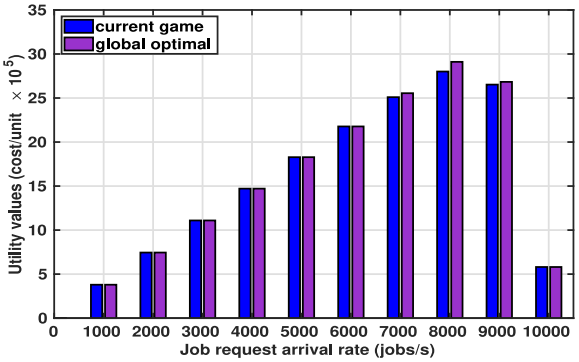


FIGURE 7. Comparison of NE utility values of competing cloudlets with high variance (0-4000 jobs/s) in incoming job request arrival rates among neighboring cloudlets with the global optimal solution.

It is interesting to note that the NE solution of our game formulation is actually dependent on the primary game design parameters Ω_1, Ω_2, Ω_3, ζ and η. Therefore, the optimal values of these parameters can be determined by studying various market equilibrium conditions for providing cloud-based services. Thus, in Figs. 10a and 10b we show the variation of utilities of two under-loaded and overloaded cloudlets against Ω_2 and Ω_3, respectively with $\lambda_i = 9990$ jobs/s and $\lambda_j = 5000$ jobs/s. Firstly, we consider $\Omega_1 = 500$, $\Omega_3 = 5\times10^8$, $\zeta = 300$, and $\eta = 700$ and tune Ω_2,

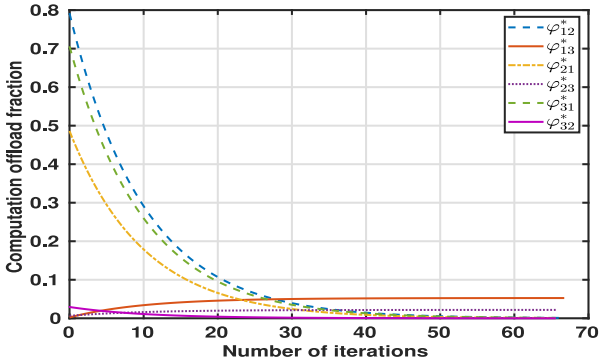


FIGURE 8. Computation offload decision of cloudlets versus (parallel update) iterations achieved using Algorithm 1. We consider three cloudlets with μ_{ii} = 10000 jobs/s and λ_1 = 9500 jobs/s, λ_2 = 9200 jobs/s, and λ_3 = 5000 jobs/s.

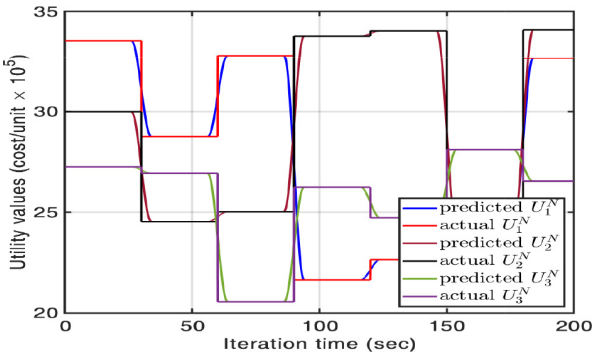
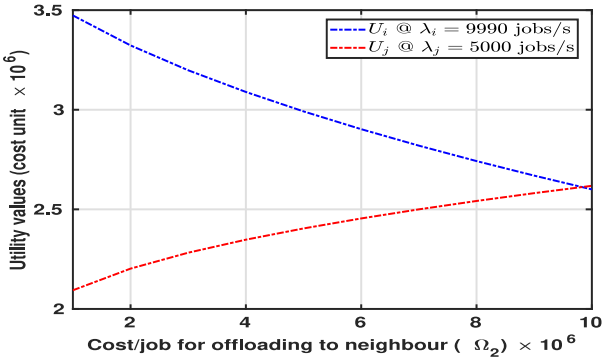


FIGURE 9. Comparison of NE utility values of competing cloudlets with actual and predicted in incoming job request arrival rates to neighboring cloudlets with μ_{ii} = 10000 jobs/s and λ_i varying within 0-10000 jobs/s.

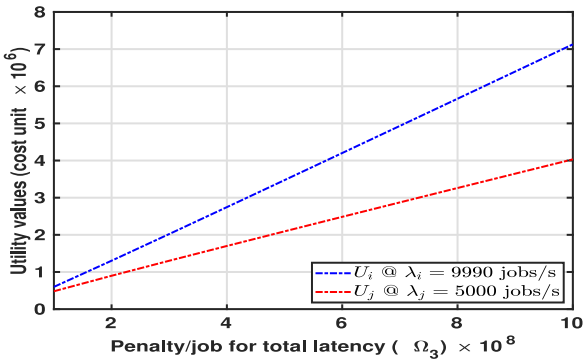
and secondly, we consider $\Omega_1 = 500$, and $\Omega_2 = 1\times10^6$ and tune Ω_3. By observing such plots with different combinations of network parameters, suitable proportionality price factors can be chosen.

VII. CONCLUSION

In this paper, we have proposed a novel economic and non-cooperative game theoretic model among multiple competitive cloudlets from same as well as different service providers in a heterogeneous cloudlet deployment scenario. We have identified the problem as a special class of GNE problems with jointly-concave shared constraints, which can be solved by VI approach. In turn, we have rigorously proven the existence and uniqueness of a pure-strategy NE of the formulated problem. Moreover, we have designed an efficient algorithm to compute the pure-strategy NE load balancing strategy among multiple cloudlets. By applying this framework over a heterogeneous cloudlet network, we have shown that all the competing cloudlets are able to maximize their utilities in under-loaded and overloaded conditions by strategically offloading their incoming job requests to their neighboring cloudlets according to our proposed NE strategy, which is greater than or equal to the utility when not participating in the market competition. Moreover, we have shown that our game formulation outperforms



(a) Variation of utilities against proportionality price factor Ω_2.



(b) Variation of utilities against proportionality price factor Ω_3.

FIGURE 10. Variation of utilities of under-loaded (λ_j = 5000 jobs/s) and overloaded (λ_i = 9990 jobs/s) cloudlets (with μ_{ii} = 10000 jobs/s) against proportionality price factors Ω_2 and Ω_3.

some of the recently proposed game-theoretic load balancing frameworks, especially in high load conditions.

APPENDIX A
PROOF OF THEOREM 1

Proof: At first, we observe that the solution space of $\mathbf{VI}(\Psi, F)$ defined as $\Psi = (\Phi \cap \aleph) \subseteq [0,1]^{N\times N}$, is *closed and bounded*, i.e., *compact*, where

$$\aleph = \left\{\boldsymbol{\varphi} \in [0,1]^{N\times N} : U_i^N \geq U_i^0, \forall i \in \mathcal{C}\right\}. \tag{27}$$

Note that, due to constraint (5), we can remove diagonal strategy elements as $\varphi_{ii} = (1 - \sum_{j=1, j\neq i}^{N} \varphi_{ij})$ and by using this expression, we re-write the utility function of each cloudlet as in (7). Now, to prove the *concavity* of the utility functions subject to the condition $[\mu_{ii} - (1 - \sum_{j\neq i}\varphi_{ij})\lambda_i - \sum_{j\neq i}\varphi_{ji}\lambda_j] \geq 0, \forall i, j \neq i \in \mathcal{C}$, at first we compute the *first-order partial derivative* (8) and set it to 0. Next, we derive the *second-order partial derivatives* as shown in (11)-(13). Clearly, the diagonal elements $\frac{\partial^2 U_i^N}{\partial \varphi_{ij}^2}$, and the non-diagonal elements $\frac{\partial^2 U_i^N}{\partial \varphi_{ij}\partial \varphi_{ik}}$ are such that the *Hessian matrix* for $U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}), \forall i \in \mathcal{C}$ is *negative semi-definite*. Hence, the utility functions $U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}), \forall i \in \mathcal{C}$ are jointly concave over $\boldsymbol{\varphi}$. Therefore, the solution space of the game $\Gamma = \langle \mathcal{C}, (\Theta_i)_{i\in\mathcal{C}}, (U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}))_{i\in\mathcal{C}}\rangle$ created by (5) and the *jointly concave shared constraints* (6) is also a *convex* set. Hence, for each competing cloudlet $i \in \mathcal{C}$:

i. the (non-empty) strategy space Θ_i is compact and convex;
ii. the utility function $U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i})$ is concave and continuously differentiable in $\boldsymbol{\varphi}_i \in \Theta_i$ for every fixed $\boldsymbol{\varphi}_{-i}$.

The above conditions are the sufficient conditions to imply that the game Γ is equivalent to the $\mathbf{VI}(\Psi, F)$, where $F = \nabla_{\boldsymbol{\varphi}_i}(U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}))_{i\in\mathcal{C}}$ [38], [43]. This theorem also implies that there exists at least one pure-strategy NE of the game Γ if Ψ_i is non-empty. ■

APPENDIX B
PROOF OF THEOREM 2

Proof: The best response function of cloudlet $i \in \mathcal{C}$ can be represented by (8), i.e., $\frac{\partial U_i^N}{\partial \varphi_{ij}} = 0$. However, the expressions are implicitly coupled among all $\varphi_{ij}, \forall i, j \neq i \in \mathcal{C}$ and presents analytically difficulties to verify the monotonicity of the best-response functions of the competing cloudlets. In this situation, we consider the utility function $U_i^2(\varphi_i, \varphi_{-i}), \forall i \in \mathcal{C}$ between $N = 2$ competing cloudlets first. To show the *monotonicity* of $\varphi_1 = g(\varphi_2)$ and $\varphi_2 = h(\varphi_1)$, we evaluate the expressions for second-order partial derivatives as follows:

$$\frac{\partial^2 U_1^2}{\partial \varphi_1^2} = -2\Omega_3 \frac{\lambda_1^2}{\mu_{11}}\left\{\frac{(\mu_{11} - \varphi_2\lambda_2)}{(\mu_{11} - (1-\varphi_1)\lambda_1 - \varphi_2\lambda_2)^3} + \frac{(\mu_2 - (1-\varphi_2)\lambda_2)}{(\mu_{22} - (1-\varphi_2)\lambda_2 - \varphi_1\lambda_1)^3}\right\} < 0, \tag{28}$$

$$\frac{\partial^2 U_2^2}{\partial \varphi_2^2} = -2\Omega_3 \frac{\lambda_2^2}{\mu_{22}}\left\{\frac{(\mu_{11} - (1-\varphi_1)\lambda_1)}{(\mu_{11} - (1-\varphi_1)\lambda_1 - \varphi_2\lambda_2)^3} + \frac{(\mu_{22} - \varphi_1\lambda_1)}{(\mu_{22} - (1-\varphi_2)\lambda_2 - \varphi_1\lambda_1)^3}\right\} < 0, \tag{29}$$

$$\frac{\partial^2 U_1^2}{\partial \varphi_2 \partial \varphi_1} = \Omega_3 \frac{\lambda_1\lambda_2}{\mu_{11}}\left[\frac{\mu_{11} + (1-\varphi_1)\lambda_1 - \varphi_2\lambda_2}{(\mu_{11} - (1-\varphi_1)\lambda_1 - \varphi_2\lambda_2)^3} + \frac{\mu_{22} - (1-\varphi_2)\lambda_2 + \varphi_1\lambda_1}{(\mu_{22} - (1-\varphi_2)\lambda_2 - \varphi_1\lambda_1)^3}\right] > 0, \tag{30}$$

$$\frac{\partial^2 U_2^2}{\partial \varphi_1 \partial \varphi_2} = \Omega_3 \frac{\lambda_1\lambda_2}{\mu_{22}}\left[\frac{\mu_{11} - (1-\varphi_1)\lambda_1 + \varphi_2\lambda_2}{(\mu_{11} - (1-\varphi_1)\lambda_1 - \varphi_2\lambda_2)^3} + \frac{\mu_{22} + (1-\varphi_2)\lambda_2 - \varphi_1\lambda_1}{(\mu_{22} - (1-\varphi_2)\lambda_2 - \varphi_1\lambda_1)^3}\right] > 0. \tag{31}$$

Therefore, by using *implicit function theorem* [44] and (28)-(31), we evaluate expressions for $d\varphi_1/d\varphi_2$ and $d\varphi_2/d\varphi_1$ as follows:

$$\frac{d\varphi_1}{d\varphi_2} = \frac{dg(\varphi_2)}{d\varphi_2} = -\frac{\left(\frac{\partial^2 U_1^2}{\partial \varphi_2 \partial \varphi_1}\right)}{\left(\frac{\partial^2 U_1^2}{\partial \varphi_1^2}\right)} > 0, \tag{32}$$

$$\frac{d\varphi_2}{d\varphi_1} = \frac{dh(\varphi_1)}{d\varphi_1} = -\frac{\left(\frac{\partial^2 U_2^2}{\partial\varphi_1\partial\varphi_2}\right)}{\left(\frac{\partial^2 U_2^2}{\partial\varphi_2^2}\right)} > 0. \tag{33}$$

Therefore, the best-response functions of both the competing cloudlets are *strongly monotone*. As the utility functions of the competing cloudlets in the $N \geq 2$ cloudlet game $\Gamma = \langle \mathcal{C}, (\Theta_i)_{i\in\mathcal{C}}, (U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}))_{i\in\mathcal{C}} \rangle$ is a direct linear extension of the $N = 2$ cloudlet game, hence the best-response functions of all the competing cloudlets are also strongly monotone. This ensures that the game Γ has a unique pure-strategy NE, but this does not necessarily imply that the corresponding $\mathbf{VI}(\Psi, F)$ has a unique solution. However, as the complete solution space $\Psi \subseteq [0,1]^{N\times N}$ of $\mathbf{VI}(\Psi, F)$ is compact and convex, the uniqueness of the solution can be guaranteed. This proves that the game Γ represented by $\mathbf{VI}(\Psi, F)$, where $F = \nabla_{\boldsymbol{\varphi}_i}(U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}))_{i\in\mathcal{C}}$ attains a unique pure-strategy NE [38], [43]. ■

APPENDIX C
PROOF OF THEOREM 3

Proof: The definition of $F = \nabla_{\boldsymbol{\varphi}_i}(U_i^N(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i}))_{i\in\mathcal{C}}$ allows us to write the KKT conditions for $\mathbf{VI}(\Psi, F)$ as follows:

$$\begin{pmatrix} \nabla_{\boldsymbol{\varphi}_1} U_1^N \\ \vdots \\ \nabla_{\boldsymbol{\varphi}_N} U_N^N \end{pmatrix} + \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \\ \boldsymbol{\xi} \end{pmatrix}^T \begin{pmatrix} \nabla_{\boldsymbol{\varphi}_1}\boldsymbol{\varphi}_1 \\ \vdots \\ \nabla_{\boldsymbol{\varphi}_N}\boldsymbol{\varphi}_N \\ \nabla_{\boldsymbol{\varphi}_1}(1-\boldsymbol{\varphi}_1) \\ \vdots \\ \nabla_{\boldsymbol{\varphi}_N}(1-\boldsymbol{\varphi}_N) \\ \nabla_{\boldsymbol{\varphi}_1}(U_i^N - U_i^0)_{i\in\mathcal{C}} \\ \vdots \\ \nabla_{\boldsymbol{\varphi}_N}(U_i^N - U_i^0)_{i\in\mathcal{C}} \end{pmatrix} = 0, \tag{34}$$

$$\begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \\ \boldsymbol{\xi} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\varphi} \\ 1-\boldsymbol{\varphi} \\ (U_i^N - U_i^0)_{i\in\mathcal{C}} \end{pmatrix} = 0. \tag{35}$$

As the solution space Ψ is compact and convex, the KKT conditions (34)-(35) attains a solution. Therefore, we can conclude by directly comparing (34)-(35) with (22)-(25) that if there exists Lagrange multipliers $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\xi}^*)$ such that $\boldsymbol{\varphi}^*$ is a solution of $\mathbf{VI}(\Psi, F)$, then it is also a solution of the corresponding game Γ. ■

REFERENCES

[1] E. Wong, M. P. I. Dias, and L. Ruan, "Predictive resource allocation for tactile Internet capable passive optical LANs," *J. Lightw. Technol.*, vol. 35, no. 13, pp. 2629–2641, Jul. 1, 2017.

[2] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct.–Dec. 2009.

[3] A. Ceselli, M. Premoli, and S. Secci, "Mobile edge cloud network design optimization," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1818–1831, Jun. 2017.

[4] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Efficient algorithms for capacitated cloudlet placements," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 10, pp. 2866–2880, Oct. 2016.

[5] Q. Fan and N. Ansari, "Cost aware cloudlet placement for big data processing at the edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[6] B. P. Rimal, D. P. Van, and M. Maier, "Cloudlet fiber-wireless access for mobile-edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3601–3618, Jun. 2017.

[7] S. Mondal, G. Das, and E. Wong, "CCOMPASSION: A hybrid cloudlet placement framework over passive optical access network," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 1–9.

[8] S. Mondal, G. Das, and E. Wong, "Cost-optimal cloudlet placement frameworks over fiber-wireless access networks for low-latency applications," *J. Netw. Comput. Appl.*, vol. 138, pp. 27–38, Jul. 2019.

[9] R. I. Tinini, D. M. Batista, G. B. Figueiredo, M. Tornatore, and B. Mukherjee, "Low-latency and energy-efficient BBU placement and VPON formation in virtualized cloud-fog RAN," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 11, no. 4, pp. B37–B48, Apr. 2019.

[10] S. Mondal, G. Das, and E. Wong, "Efficient cost-optimization frameworks for hybrid cloudlet placement over fiber-wireless networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 11, no. 8, pp. 437–451, Aug. 2019.

[11] A. Yousefpour *et al.*, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *J. Syst. Archit.*, vol. 98, pp. 289–330, Sep. 2019.

[12] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing-based IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2146–2153, Jun. 2018.

[13] V. Cardellini *et al.*, "A game-theoretic approach to computation offloading in mobile cloud computing," *Math. Program.*, vol. 157, no. 2, pp. 421–449, Jun. 2016.

[14] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[15] B. P. Rimal and M. Maier, "Mobile data offloading in FiWi enhanced LTE-A heterogeneous networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 7, pp. 601–615, Jul. 2017.

[16] H. Cao and J. Cai, "Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 752–764, Jan. 2018.

[17] Y. Jiang, "A survey of task allocation and load balancing in distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 2, pp. 585–599, Feb. 2016.

[18] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[19] L. Liu, S. Chan, G. Han, M. Guizani, and M. Bandai, "Performance modeling of representative load sharing schemes for clustered servers in multiaccess edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4880–4888, Jun. 2019.

[20] Q. Fan and N. Ansari, "Towards workload balancing in fog computing empowered IoT," *IEEE Trans. Netw. Sci. Eng.*, early access, doi: 10.1109/TNSE.2018.2852762.

[21] R. Beraldi, A. Mtibaa, and H. Alnuweiri, "Cooperative load balancing scheme for edge computing resources," in *Proc. 2nd Int. Conf. Fog Mobile Edge Comput. (FMEC)*, May 2017, pp. 94–100.

[22] D. Zhang, Y. Ma, C. Zheng, Y. Zhang, X. S. Hu, and D. Wang, "Cooperative-competitive task allocation in edge computing for delay-sensitive social sensing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 243–259.

[23] Q. He *et al.*, "A game-theoretical approach for user allocation in edge computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 3, pp. 515–529, Mar. 2020.

[24] C. Liu, K. Li, and K. Li, "A game approach to multi-servers load balancing with load-dependent server availability consideration," *IEEE Trans. Cloud Comput.*, early access, doi: 10.1109/TCC.2018.2790404.

[25] L. Chen, S. Zhou, and J. Xu, "Computation peer offloading for energy-constrained mobile edge computing in small-cell networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1619–1632, Aug. 2018.

[26] S. Mondal, G. Das, and E. Wong, "Computation offloading in optical access cloudlet networks: A game-theoretic approach," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1481–1484, Jul. 2018.

[27] S. Penmatsa and A. T. Chronopoulos, "Game-theoretic static load balancing for distributed systems," *J. Parallel Distrib. Comput.*, vol. 71, no. 4, pp. 537–555, 2011.

[28] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
[29] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory* (OUP Catalogue). Oxford, U.K.: Oxford Univ. Press, 1995.
[30] X. Zhou, K. Wang, W. Jia, and M. Guo, "Reinforcement learning-based adaptive resource management of differentiated services in GEO-distributed data centers," in *Proc. IEEE/ACM 25th Int. Symp. Quality Service (IWQoS)*, Jun. 2017, pp. 1–6.
[31] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. New York, NY, USA: Cambridge Univ. Press, 2008.
[32] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5141–5154, Dec. 2018.
[33] C. Bhar, N. Chatur, A. Mukhopadhyay, G. Das, and D. Datta, "Designing a green optical network unit using ARMA-based traffic prediction for quality of service-aware traffic," *Photon. Netw. Commun.*, vol. 32, no. 3, pp. 407–421, Dec. 2016.
[34] C. Jiang, Y. Chen, Q. Wang, and K. J. R. Liu, "Data-driven stochastic scheduling and dynamic auction in IaaS," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.
[35] D. P. Bertsekas and R. G. Gallager, *Delay Models in Data Networks*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992, ch. 3, pp. 149–270.
[36] IBM Cloud Pricing. Accessed: Feb. 2020. [Online]. Available: https://www.ibm.com/cloud/pricing
[37] J. Rosen, "Existence and uniqueness of equilibrium points for concave N-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965.
[38] F. Facchinei, A. Fischer, and V. Piccialli, "On generalized Nash games and variational inequalities," *Oper. Res. Lett.*, vol. 35, no. 2, pp. 159–164, Mar. 2007.
[39] G. Scutari, F. Facchinei, J. S. Pang, and D. P. Palomar, "Real and complex monotone communication games," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 4197–4231, Jul. 2014.
[40] T. Roughgarden, "Intrinsic robustness of the price of anarchy," *J. ACM*, vol. 62, no. 5, pp. 1–32, Nov. 2015.
[41] C. J. Zarowski, *An Introduction to Numerical Analysis for Electrical and Computer Engineers*. Hoboken, NJ, USA: Wiley 2004.
[42] J. Wallenius, P. C. Fishburn, S. Zionts, J. S. Dyer, R. E. Steuer, and K. Deb, "Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead," *Manag. Sci.*, vol. 54, no. 7, pp. 1336–1349, 2008.
[43] G. Scutari, D. P. Palomar, F. Facchinei, and J. S. Pang, "Convex optimization, game theory, and variational inequality theory," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 35–49, May 2010.
[44] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. New York, NY, USA: McGraw-Hill, 1976.

SOURAV MONDAL (Student Member, IEEE) received the B.Tech. degree in electronics and communication engineering from Kalyani Govt. Engineering College, West Bengal University of Technology in 2012, and the M.Tech. degree in telecommunication systems engineering from the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology Kharagpur in 2014. He is currently pursuing the doctoral degree with the Department of Electrical and Electronic Engineering, University of Melbourne and has started to work on Cloudlet based edge-computing for low-latency applications over optical access networks.

GOUTAM DAS (Member, IEEE) received the M.Tech. degree from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 2001, and the Ph.D. degree from the University of Melbourne, Australia, in 2008. He has also worked as a Senior Research Engineer with General Electric R&D Center from 2001 to 2004. He has also worked as a Postdoctoral Fellow with Ghent University, Belgium, from 2009 to 2011. He is currently working as an Assistant Professor with the Indian Institute of Technology Kharagpur. His research interest is in the area of both optical as well as wireless networking.

ELAINE WONG (Senior Member, IEEE) received the B.E. and Ph.D. degrees from the University of Melbourne, Melbourne, VIC, Australia, where she is currently a Professor. Her research interests include energy-efficient optical and wireless networks, optical-wireless integration, broadband applications of vertical-cavity surface-emitting lasers, wireless sensor body area networks, and emerging optical and wireless technologies for tactile Internet. She has coauthored more than 150 journal and conference publications. She has served on the editorial board for the JOURNAL OF LIGHTWAVE TECHNOLOGY and the *Journal of Optical Communications and Networking*.