Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8888074, 14 pages
https://doi.org/10.1155/2020/8888074

Research Article

New Application Task Offloading Algorithms for Edge, Fog, and Cloud Computing Paradigms

Sungwook Kim

Department of Computer Science, Sogang University, 35 Baekbeom-ro, Sinsu-dong, Mapo-gu, Seoul 04107, Republic of Korea

Correspondence should be addressed to Sungwook Kim; swkim01@sogang.ac.kr

Received 15 March 2020; Revised 15 July 2020; Accepted 21 August 2020; Published 6 October 2020

Academic Editor: Miguel Garcia-Pineda

In the last few years, we have seen an exponential increase in the number of computation-intensive applications, which have resulted in the popularity of fog and cloud computing paradigms among smart-chip-embedded mobile devices. These devices can partially offload computation tasks either using the fog system or using the cloud system. In this study, we design a new task offloading scheme by considering the challenges of future edge, fog and cloud computing paradigms. To provide an effective solution toward an appropriate task offloading problem, we focus on two cooperative bargaining game solutions—Tempered Aspirations Bargaining Solution (TABS) and Gupta-Livne Bargaining Solution (GLBS). To maximize the application service quality, a proper bargaining solution should be properly selected. In the proposed scheme, the TABS method is used for time-sensitive offloading services, and the GLBS method is applied to ensure computation-oriented offloading services. The primary advantage of our bargaining-based approach is to provide an axiom-based strategic solution for the task offloading problem while dynamically responding to the current network environments. Extensive simulation studies are conducted to demonstrate the effectiveness of the proposed scheme, and the superior performance over existing schemes is observed. Finally, we show prime directions for future work and potential research issues.

1. Introduction

Currently, billions of smart devices connect to the Internet in the form of the Internet of Things (IoT). IoT is a worldwide network based on standard communication protocols and a novel paradigm with access to wireless communication systems. It applies various technologies to provide the promising fifth generation (5G) service applications. Meanwhile, the evolution of 5G networks is becoming a major driving force for the growth of IoT. For the connection of billions of smart devices, 5G-based IoT infrastructure is expected to have extended coverage, higher throughput, lower latency, and connection density of massive bandwidth. However, the management of such different kinds of control criteria is cumbersome and challenging for traditional network infrastructures that rely on conventional computing paradigms [1, 2].

Despite the advance in the capacity of smart devices, mobile hardware is still resource-poor compared to the system server hardware. Constrained by battery life, storage limitation, computation capacity, and wireless bandwidth scarcity, the resource-poor mobile devices encounter the difficulty of supporting content-rich or computation-intensive applications such as real-time image processing for video games, augmented reality, and location-based services. Cloud computing is introduced as a promising paradigm to overcome the above difficulty. By employing this cloud computing method, the computing, data storage, and mass information processing can be offloaded to the cloud servers while ensuring the reliability and availability of the application services. This new paradigm is termed as the Cloud of Things (CoT), which helps in creating an extended portfolio of future network architecture [3, 4].

CoT offers an efficient computing model where system resources can be shared as services through IoT. However, connecting to the remote cloud server causes communication latency, and the cloud cannot easily respond in real time to frequent network dynamics; it turns down the expected

advantages of CoT. Usually, mobile devices can no longer afford to wait for the varying response time of a cloud-based computation service, especially with stringent demands on tolerated delay. Therefore, the rising tide is driving toward a new technology. Fog computing is a solution to subdue the shortcomings of cloud computing. It is a highly distributed platform with fog computing nodes, such as cloudlets, located at the edge of the Internet. As a mobility-enhanced small-scale cloud datacenter, the main purpose of cloudlets is arbitrating resource-intensive and interactive mobile applications with lower latency. It is a new architectural structure, called Fog-of-Things (FoT), which extends the CoT paradigm to leverage recent developments in future networks [5, 6].

Initially, IoT devices had simply developed to collect and send data for analysis, but lacked system elements to perform complex computations on-site. However, recent advancements in embedded systems-on-a-chip have significantly increased the number of intelligent devices that possess some resources to partially run computation-intensive applications [2]. This trend has extended the potential of IoT, and paves a way to develop a new paradigm, called the Edge-of-Things (EoT). Actually, there is a high possibility that CoT and FoT paradigms will encounter more challenges in relation to network dynamics, resulting in a high overhead in the network response time, leading to time latency and traffic burden. In order to avoid these problems while achieving an efficient resource utilization, the EoT paradigm may become necessary in future network services [7].

While FoT and EoT paradigms have some similarities, there is a major difference. First, both paradigms involve pushing intelligence and processing capabilities down closer to where services originate. Therefore, they share similar objectives (i) to reduce the amount of data sent to the cloud, (ii) to decrease network and Internet latency, and (iii) to improve system response time in remote mission-critical applications. However, there is a key difference between FoT and EoT; it is exactly where intelligence and computing power is placed. FoT pushes intelligence down to the local area network level of the network architecture, processing data in a fog node or IoT gateway. This approach can achieve a number of benefits including on-demand service, resource pooling, and virtualization. Metaphorically speaking, fog computing sits between physical things and cloud computing, just like in nature, where fog exists between the ground and clouds. Contrary to FoT, EoT pushes the intelligence, processing power, and communication capabilities of an edge gateway or appliance directly into devices. To ensure Quality of Experience (QoE) in terms of latency, bandwidth, and security, the applications running on the EoT paradigm will perform actions locally before connecting to the cloud, thus reducing network overhead issues as well as security and privacy issues. Therefore, EoT can bring new benefits such as early data resolution; responsive management on the edge; and improved latency, robustness, and security. However, due to cost and energy consumption issues, edge devices typically have limited capacities [7, 8].

Fortunately, CoT, FoT, and EoT paradigms are not incompatible in nature; in fact, they compensate each other's limitations. More importantly, the future network concept is the convergence of CoT, FoT, and EoT paradigms; it has inspired us to seek a joint solution to maximize the performance of future networks. In this study, we propose a new task offloading control scheme by considering the merits of CoT, FoT, and EoT paradigms. Based on the combined design of different paradigm operations, our integrated approach can obtain a synergy effect while attaining an appropriate performance balance. However, it is an extremely challenging work to combine the CoT, FoT, and EoT paradigms into a holistic scheme. Therefore, a new solution concept is required.

Since the 1950s, game theory has been used to study strategic interactions. Whenever the choices made by two or more individuals have an effect on each other's gains or losses, and hence their actions, the interaction between them is game-theoretic in nature. In recent years, there has been a remarkable increase in work at the interface of game theory and many academic research fields from economics to computer science. Especially, game theory has been playing an increasingly visible role in network management, in areas such as resource management, routing mechanism, power control, and traffic modeling. There is a major reason for this; the Internet calls for analysis and design of systems that span multiple entities with diverging information and interests. Game theory, for all its limitations, is by far the most developed theory of such interactions [9].

1.1. Motivation. The aim of this study is to propose a novel task offloading control scheme for a hierarchical future network system. To tackle the task offloading problem in mixed edge-fog-cloud computing, we employ the CoT, FoT, and EoT paradigms, and jointly consider the combination of mobile devices, cloudlets, and a cloud system. They need to coexist and synthetically complement each other to meet the diverse requirements of future networks. To investigate the strategic interactions among cloud, fog, and edge computing paradigms, we formulate mobile device/cloudlet/cloud-connected cooperative games, and adopt the Tempered Aspirations Bargaining Solution (TABS) and the Gupta-Livne Bargaining Solution (GLBS). Both are based on bargaining solution guidelines, and each individual mobile device and its corresponding cloudlet and cloud server work cooperatively to negotiate their conflicting interests while guaranteeing fairness and efficiency.

The main challenge of our game-based task offloading approach is to retain generality for future networks. Definitely, future networks will adopt new computing paradigms, and a three-layer hierarchical network system can be extended complicatedly. Therefore, CoT, FoT, and EoT paradigms could be replaced by new computing fashions. To adapt to these dynamics, our proposed task offloading control scheme is not fixed to specific computing paradigms but is designed to be dynamic and flexible and can adaptively respond to new future network infrastructures. This is the main advantage of our proposed scheme over the traditional task offloading scheme.

1.2. Major Contributions. To fulfill the promised advantages of three-layer hierarchical network platforms, several technical issues and challenges should be addressed. In this study, our work addresses the task offloading problem by adopting TABS and GLBS. To model the interactions among mobile devices, cloudlets, and a cloud system, we design a new cooperative bargaining game process. Using two different bargaining solutions, the proposed scheme effectively allocates the hierarchical network resources in a fair-efficient manner. With self-adaptability and real-time effectiveness, a well-balanced solution can be obtained while leveraging the full synergy of the CoT, FoT, and EoT paradigms. In summary, the contributions of this paper are as follows:

(i) By employing CoT, FoT, and EoT paradigms: motivated by the future IoT environments, we assume a three-layer hierarchical network system by employing the CoT, FoT, and EoT paradigms. Depending on the different computing characteristics, they work together toward an appropriate network performance

(ii) Computation-intensive task offloading based on GLBS: according to GLBS, a computation-intensive task is offloaded to fog and cloud servers. This approach can investigate the potential benefit gained from its delay-tolerant characteristics

(iii) Time-sensitive task offloading based on TABS: based on TABS, the time-sensitive task is offloaded to fog and cloud servers. This approach can maximize the expected payoff obtained from its delay-sensitive characteristics

(iv) Jointly designed to leverage the synergistic and complementary features: we explore the interaction of GLBS and TABS methods to balance contradictory requirements. The main idea of our approach lies in its responsiveness to the reciprocal combination of different bargaining solutions

(v) Reciprocal negotiation and self-adaptability: from the viewpoint of practical operations, the main features of our bargaining-based task offloading scheme are reciprocal negotiation and self-adaptability. Under dynamic hierarchical network environments, these characteristics are generic and applicable for real-world operations while ensuring a fair-efficient solution

(vi) Performance analysis: the major challenge of our proposed scheme is to strike the appropriate performance fairly and efficiently. A numerical simulation study shows that a timely effective solution is dynamically obtained based on the jointly bargaining solutions

Beyond the feasible combination of optimality and practicality, the possible advantages of our approach include adaptability, flexibility, and responsiveness to current network system conditions. To the best of our knowledge, little research has been conducted on bargaining-based task offloading algorithms for future hierarchical network systems.

1.3. Organization. The remainder of this article is organized as follows. In Section 2, some related researches about cloud and fog computing-based task offloading problems are discussed. In Section 3, we provide a three-layer hierarchical network system infrastructure for the task offloading problem and formulate two cooperative bargaining game models for different kinds of application services. Then, we design our proposed scheme aiming at maximizing the system performance. We also provide the primary steps of the proposed scheme for readers' convenience. In Section 4, we evaluate the performance of our proposed scheme through extensive simulations. Finally, concluding remarks are drawn in Section 5 with future work.

2. Related Work

Cloud, fog, and edge computing mechanisms, which are kinds of Internet-based paradigms, have attracted great attention with a large quantity of literatures. In [10], the Fair and Energy-Minimized Task Offloading (FEMTO) algorithm is proposed based on a fairness scheduling metric, taking three important characteristics into consideration, which include the task offloading energy consumption, the fog node's historical average energy, and fog node priority. Based on the fairness scheduling metric, the FEMTO algorithm determines the task offloading solution including the target fog node, the terminal node transmission power, and the subtask size in a fair and energy-minimized manner. Finally, extensive simulations are carried out in a fog-enabled IoT network to investigate the performance of the proposed FEMTO algorithm [10].

The article [11] studies the problem of dynamic offloading and resource allocation with prediction in a fog computing system with multiple tiers. By formulating it as a stochastic network optimization problem, the Predictive Offloading and Resource Allocation (PORA) algorithm is developed. The PORA algorithm exploits predictive offloading to minimize power consumption with queue stability guarantee. Theoretical analysis and simulation results show that the PORA algorithm incurs near-optimal power consumptions with a guarantee of queue stability. Furthermore, it requires only a mild value of predictive information to achieve a notable latency reduction, even with the prediction errors [11].

Yousefpour et al. introduced a general framework for IoT-fog-cloud applications and proposed a delay-minimizing collaboration and offloading policy for fog-capable devices that is aimed at reducing the service delay for IoT applications [12]. The authors developed an analytical model to evaluate their policy and showed how the proposed framework helps to reduce IoT service delay. In contrast to the existing schemes, their proposed policy considers IoT-to-cloud and fog-to-cloud interactions and also employs fog-to-fog communications to reduce the service delay by sharing load. For load sharing, it considers not only the queue length

but also different request types that have various processing times [12].

The authors in [13] designed a more efficient and secure cloud storage based on fog computing. By offloading part of the computing and storage work to the fog servers, the Reed-Solomon code is also introduced to protect the privacy of users. Therefore, data privacy can be guaranteed. To decrease the communication cost and reduce latency, they developed a differential synchronization algorithm, which provides a feasible solution but increases the workload on the users' devices and the cloud server. By offloading part of the work to the fog server, the efficiency of the entire process can be improved. Finally, the experiment results show that their architecture is feasible and has better performance than the other methods [13].

The Joint User equipment and Fog Optimization (JUFO) scheme is designed to minimize the energy consumption of the user's equipment and fog system based on the priority distribution of cloud tasks while maintaining service time constraints [14]. It is based on the popularity distribution of cloud tasks and energy consumption model. A network system consisting of a user's equipment, a fog server, and a remote cloud server is considered, where the user's equipment sends requests for cloud services, and the fog server and the remote cloud server process the requested service. In order to solve the optimization problem, the energy consumption and service time of each network component are mathematically modeled. The advantage of the JUFO scheme comes from using the profile of each cloud task in the optimized fog server offloading control scheme. Simulation results show that the JUFO scheme can provide a significant savings in energy consumption while supporting real-time service requirements in regions with burdening workloads [14].

The authors in [15] proposed the Joint Radio and Computational Resource Allocation (JRCRA) scheme. The JRCRA scheme investigates a joint radio and computational resource allocation problem to optimize the system performance and improve user satisfaction. By communicating with the users, cloud providers try to find suitable fog nodes for offloading users' computation tasks, together with the assignment of a radio spectrum, to satisfy users' requirements. With the objective of optimizing the users' satisfaction, they formulate this joint resource allocation as a mix integer nonlinear programming problem. Therefore, the interactions among the IoT users, service providers, and fog nodes have been modeled based on the matching game framework, and the transmission quality, service latency, and maximum power requirement have been effectively addressed. Through the simulation results, they conform that their proposed approach achieves the distributive, close-to-optimal performance from both the users' perspective and the system's view [15].

The Hierarchical Fog-Cloud Computation Offloading (HFCCO) scheme in [16] focuses on the allocation of fog computing resources to the IoT users in a hierarchical computing paradigm including fog and remote cloud computing services. The major goal of this scheme is to determine the offloading decision for each task arriving to the IoT users, where each user is interested in maximizing its own QoE. Utilizing a potential game model, the HFCCO scheme proves the existence of a pure Nash Equilibrium (NE) and develops an algorithm to obtain NE. To mitigate the time complexity of obtaining NE, a near-optimal resource allocation algorithm is also provided and shows that it reaches ε -NE in polynomial time. Numerical analysis shows that the IoT users can obtain a higher QoE, and the computation time of delay-sensitive IoT applications is reduced significantly when utilizing the computing resources of fog nodes. These results demonstrate the ability of fog nodes in providing low-latency computing services in IoT systems [16].

In [17], the Fog-Cloud Optimal Workload Allocation (FCOWA) scheme is proposed for the tradeoff between power consumption and transmission delay in the fog-cloud computing system. To provide a systematic framework of computation and communication codesign in the fog-cloud computing system, the FCOWA scheme models the power consumption function and delay function of each part of the fog-cloud computing system and formulates the workload allocation problem. This problem can be decomposed into three subproblems of three corresponding subsystems, which are solved via existing optimization techniques. Extensive simulations show that the fog computing mechanism can significantly improve the performance of the cloud computing mechanism while sacrificing modest computation resources to save communication bandwidth and reduce transmission latency [17].

Chen et al. developed a novel traffic-flow prediction algorithm that is based on long short-term memory with an attention mechanism to train mobile-traffic data in a single-site mode [18]. The proposed algorithm is capable of effectively predicting the peak value of the traffic flow. This predicted peak value is sent to a remote cloud. At the remote cloud, resources are dispatched and allocated dynamically based on traffic adaptation using a cognitive engine and an intelligent mobile-traffic module to balance the network load. For a multisite case, they also presented an intelligent IoT-based mobile-traffic prediction-and-control architecture capable of dynamically dispatching communication and computing resources. With the support of the cognitive engine and mobile-traffic control modules, the mobile-traffic flow for the entire network is predicted and controlled intelligently [18].

The paper [19] proposes an intelligent task offloading scheme, called the iTask-Offloading scheme, for a cloud-edge collaborative system. The architecture of iTask-Offloading includes the local device layer, the edge cloud layer, the remote cloud layer, and the cognitive engine; it can not only recognize the resources from the local device, the edge cloud, and the remote cloud, but it also understands the task of intelligent application. The iTask-Offloading scheme is designed to combine the cognitive engine with the traditional cloud-edge collaborative system, and provides fine-grained task offloadings for the separability of intelligent application tasks to enable personalized task offloading. Finally, a real testbed is built to show that the iTask-Offloading scheme has less latency than traditional cloud computing [19].

In [20], the authors proposed a new Edge Cognitive Computing (ECC) architecture that deploys cognitive computing at the edge of the network to provide dynamic and elastic storage and computing services. In addition, they proposed an ECC-based dynamic cognitive service-migration mechanism that considers both the elastic allocation of the cognitive computing services and user mobility, to provide a mobility-aware dynamic service-adjustment scheme. Finally, they developed an ECC-based test platform to evaluate system performance; the results effectively demonstrate that edge cognitive computing realizes the cognitive information cycle for human-centered reasonable resource distribution and optimization [20].

Chen and Hao investigated the task offloading problem in an ultradense network aiming to minimize the delay while saving the battery life of a user's equipment [21]. Specifically, they formulated a task offloading problem as a mixed integer nonlinear program and transformed this optimization problem into two subproblems, i.e., a task placement subproblem and a resource allocation subproblem. Based on the solution of the two subproblems, they proposed an efficient offloading scheme. Simulation results have shown that their proposed scheme is more efficient compared to the random and uniform computation offloading schemes [21].

The paper [22] proposes a new mobile cloudlet-assisted service mode named Opportunistic task Scheduling over Co-located Clouds (OSCC), which achieves flexible cost-delay tradeoffs between the conventional remote cloud service mode and the mobile cloudlet service mode. Then, this work performs detailed analytic studies for the OSCC mode and solves the energy minimization problem by compromising between the remote cloud mode, the mobile cloudlet mode, and the OSCC mode. In addition, this study introduces two different kinds of task allocation schemes, i.e., dynamic allocation and static allocation. Under both the mobile cloudlet mode and the OSCC mode, dynamic allocation exhibits lower cost than static allocation [22].

3. The Bargaining-Game-Based Task Offloading Algorithms

In this section, we describe the three-layer hierarchical network architecture based on the CoT, FoT, and EoT paradigms. It presents the different emerging technologies, which can be combined to approximate the optimal system performance. According to the cooperative game approach, we can get an effective bargaining solution while adapting the fast changing future network environments.

3.1. Hierarchical Network Architecture for Task Offloading Services. In this study, we consider a future network system with a hierarchical computing structure and discuss the functional capabilities of different computing paradigms with their physical properties. The main objective of the hierarchical architecture is to provide a better QoE for end users. Edge devices may either perform their tasks locally or offload them to computing servers, which are the cloudlets in close proximity and the remote cloud server. In our proposed scheme, we address the task offload problem according to cooperative bargaining models, which are formulated by cooperation, coordination, and collaboration of the device, the cloudlet, and the cloud server.

As shown in Figure 1, we assume a three-layer hierarchical network system comprised of multiple IoT devices, such as smart phones, surveillance cameras, personal digital assistants, laptops, and on-board units, denoted as the set of EoT devices $\mathbb{D} = \{\mathscr{D}_1, \mathscr{D}_2, \cdots, \mathscr{D}_n\}$. $\mathscr{D}_{1\le i\le n}$ generates different application service requests $\mathbb{A}^{\mathscr{D}_i} = \{\mathscr{A}_1^{\mathscr{D}_i}, \mathscr{A}_2^{\mathscr{D}_i}, \mathscr{A}_3^{\mathscr{D}_i}, \cdots\}$ and may offload certain amounts of computing tasks to the fog nodes, denoted as the set of cloudlets $\mathbb{F} = \{\mathscr{CL}_1, \mathscr{CL}_2 \cdots, \mathscr{CL}_m\}$, and one cloud server ($\mathbb{C}$). $\mathscr{D}_{1\le i\le n}$, $\mathscr{CL}_{1\le j\le m}$, and $\mathbb{C}$ have their computation power capacities, i.e., $\mathfrak{P}^{\mathscr{D}_i}$, $\mathfrak{P}^{\mathscr{CL}_j}$, and $\mathfrak{P}^{\mathbb{C}}$, respectively, which can be consumed by a monotonic increasing function of the computation amount. Whereas in reality, the $\mathfrak{P}^{\mathscr{D}_i}$, $\mathfrak{P}^{\mathscr{CL}_j}$, and $\mathfrak{P}^{\mathbb{C}}$ resources are limited and raced. When a lot of computation-intensive applications are executed, these resources will become exhausted rapidly. Due to the resource scarcity, it is impossible to guarantee all applications' needs. To maximize the overall system performance, it is necessary to effectively utilize these computation resources for different application requests.

Despite the obvious advantages of using offloading services to process IoT applications, the future network system still suffers from the degraded QoE from service delays. Different application services not only require different computation intensities, but also have different delay sensitivities. Since the future network system covers a large geographical area from the edge device ($\mathscr{D}$) to the central cloud ($\mathbb{C}$), the communication delay should be taken into account. According to the required QoE, various application services can be categorized into two classes: computation-intensive applications and delay-sensitive applications. To make offloading decisions, we must consider the required QoE. Therefore, resource management strategy becomes a key factor in enhancing the future network system performance while ensuring service constraints.

To tackle the future network task offloading problem, we adopt two cooperative bargaining solutions: Tempered Aspirations Bargaining Solution (TABS) and Gupta-Livne Bargaining Solution (GLBS) [23]. Each individual mobile device offloads its application task ($\mathscr{A}$) while partitioning the computation amount ($\Gamma^{\mathscr{A}}$) into three parts, i.e., $\mathfrak{P}_{\mathscr{A}}^{\mathscr{D}}$, $\mathfrak{P}_{\mathscr{A}}^{\mathscr{CL}}$, and $\mathfrak{P}_{\mathscr{A}}^{\mathbb{C}}$; they are assigned to its own device $\mathscr{D}$, the corresponding $\mathscr{CL}$, and $\mathbb{C}$, respectively. To adaptively partition its $\Gamma^{\mathscr{A}}$, the main ideas of TABS and GLBS are applied. Based on two bargaining solutions, we can take various benefits in a fair-efficient way.

3.2. Tempered Aspirations and Gupta-Livne Bargaining Solutions. Let N be the set of potential bargainers, and $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_{++}$ are denoted as the sets of all, nonnegative and positive real numbers, respectively. $\mathbb{R}^n$ is the n-fold Cartesian product of $\mathbb{R}$. We use conventional notation for comparison of vectors: $\mathbf{x} \ge \mathbf{y}$ means that $x_{1\le i\le n} \ge y_{1\le i\le n}$, $\mathbf{x} > \mathbf{y}$ indicates

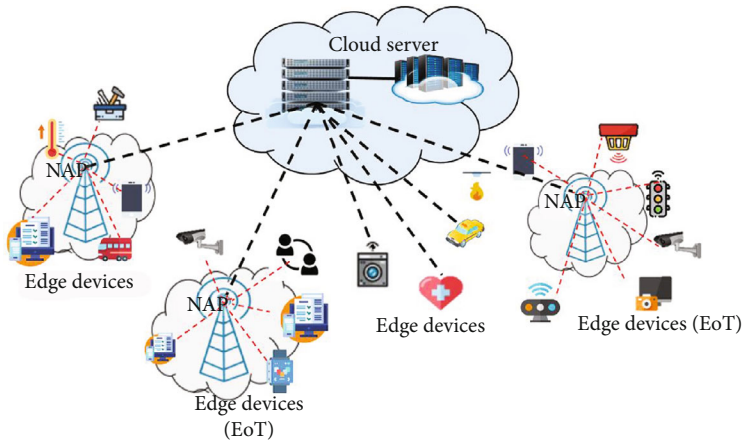


FIGURE 1: The infrastructure of the three-layer hierarchical network.

that $\mathbf{x} \geq \mathbf{y}$, and $\mathbf{x} \neq \mathbf{y}$ and $\mathbf{x} \gg \mathbf{y}$ means $x_{1\leq i\leq n} > y_{1\leq i\leq n}$. Let $co(A)$ denote the convex hull of set A in $\mathbb{R}^n$; it is mathematically expressed as $co(A) = \{\mathscr{Z} \in \mathbb{R}^n \mid \mathscr{Z} = (\alpha \times \mathbf{x}) + ((1-\alpha) \times \mathbf{y}), \mathbf{x}, \mathbf{y} \in A \text{ and } \alpha \in [0,1]\}$. Let $cch(A)$ denote the convex and comprehensive hull of A, $cch(A) = \{\mathbf{y} \in \mathbb{R}^n \mid \mathbf{y} \leq \mathscr{Z}, \mathscr{Z} \in co(A)\}$. If N has more than one member, for every $\mathbf{x} \in \mathbb{R}^n$ and every $i \in N$, define $\mathbf{x}_{-i} = \mathbf{x}_{N\setminus\{i\}}$ [23, 24]. A disagreement point (d) is a vector $d = (d_1, \cdots, d_n)$ that is expected to be the result if bargainers cannot reach an agreement. A bargaining problem for N is a pair (S, d) such that S is a bargaining set for N, $d \in S$, and there exists an $\mathbf{x} \in S$ satisfying $\mathbf{x} \gg d$. Let the aspiration vector $a(S, \mathbf{x})$ be defined by [23].

$$a_i(S, \mathbf{x}) = \max \{t \in \mathbb{R} \mid (t, \mathbf{x}_{-i}) \in S\} \text{ for every } i \in N. \quad (1)$$

The ideal point of the problem (S, d) represents bargainers' expectations before bargaining negotiation and it is defined by $a(S, d)$. Denote the family of all bargaining problems for N by Σ^N. The reference point $r \in S$ satisfies $r \geq d$. A solution concept on Σ^N is a function ϕ that associates with each triple $(S, d, r) \in \Sigma^N$, and a unique outcome of ϕ is denoted as $\phi(S, d, r) \in S$ [23].

In 2011, P.V. Balakrishnana et al. proposed a new bargaining solution, called Tempered Aspirations Bargaining Solution (TABS). With the reference point (r), *TABS* is defined for every $(S, d, r) \in \Sigma^N$ as [23]:

$$\begin{aligned} \text{TABS}(S, d, r) &= [\lambda^* \times a(S, r)] + [(1 - \lambda^*) \times \mathrm{d}] \\ \text{s.t.} \quad \lambda^* &= \max \{\lambda \in [0,1] \mid ([\lambda \times a(S, r)] \\ &\quad + [(1-\lambda) \times \mathrm{d}]) \in \mathrm{S}\}. \end{aligned} \quad (2)$$

If a bargaining problem is translated so that the disagreement point is at the origin, TABS is the only point along the frontier of S proportional to the aspirations vector $a(S, r)$. TABS can be axiomatically characterized by using the following axioms: Weak Pareto-Optimality, Symmetry, Scale Invariance axioms, r-Restricted$S$-Monotonicity, Irrelevance of Trivial Reference Points, and S-Continuity [23].

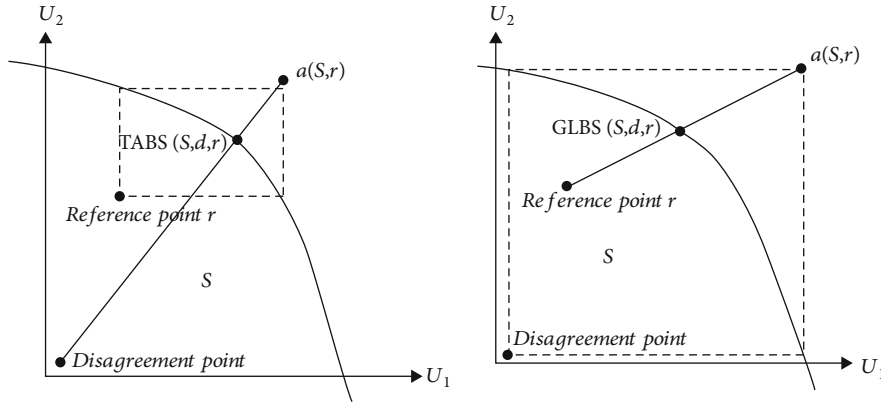
(i) Weak Pareto-Optimality (WPO): for every bargaining set S, define its Pareto-optimal set as $\text{PO}(S) = \{\mathbf{y} \in S \mid \mathbf{x} > \mathbf{y} \text{ implies } \mathbf{x} \notin S\}$. Similarly, its weak Pareto-optimal set is defined as $\text{WPO}(S) = \{\mathbf{y} \in S \mid \mathbf{x} \gg \mathbf{y} \text{ implies } \mathbf{x} \notin S\}$. For every $(S, d, r) \in \Sigma^N$, $\phi(S, d, r) \in \text{WPO}(S)$

(ii) Symmetry (SYM): let $\Pi(N)$ be the set of permutations of set N. For every $\pi \in \Pi(N)$ and every $\mathbf{x} \in \mathbb{R}^N$, define $\pi(\mathbf{x}) \in \mathbb{R}^N$ as the vector such that for every $i \in N$, $(\pi(\mathbf{x}))_{\pi(i)} = x_i$. For every $X \subseteq \mathbb{R}^N$ define $\pi(X) = \{\pi(\mathbf{x}) \mid \mathbf{x} \in X\}$. A problem $(S, d, r) \in \Sigma^N$ is said to be symmetric if, for every $\pi \in \Pi(N)$, $\pi(S) = S$, $\pi(d) = d$, and $\pi(r) = r$. Therefore, for every $(S, d, r) \in \Sigma^N$, if (S, d, r) is symmetric then, for every $i, j \in N$, $\phi(S, d, r) = \phi_j(S, d, r)$

(iii) Scale Invariance (SC.INV): let $\mathscr{L}$ be the family of vectors of functions $(L_i)_{i\in N}$ such that for every $i \in N$, there exist $m_i \in \mathbb{R}_{++}$ and $b_i \in \mathbb{R}$ satisfying, for every $t \in \mathbb{R}$, $L_i(t) = m_i \times t + b_i$. For every $(S, d, r) \in \Sigma^N$ and every $L \in \mathscr{L}$, $\phi(L(S), L(d), L(r)) = L(\phi(S, d, r))$

(iv) r-Restricted S-Monotonicity (r-REST.S-MON): in the presence of a reference point, the ideal point, $a(S, d)$, is substituted by the vector of aspirations, $a(S, r)$. For every (S, d, r) and $(S', d', r') \in \Sigma^N$, $(d, r) = (d', r')$, $S \subseteq S'$, and $a(S, r) = a(S', r')$, imply $\phi(S, d, r) \leq \phi(S', d', r')$

(v) Irrelevance of Trivial Reference Points (ITR): whenever introducing a reference point does not change the bargainers' initial aspirations $a(S, d)$, the reference point might as well be replaced by the disagreement point. For every $(S, d, r) \in \Sigma^N$, $a(S, r) = a(S, d)$, imply $\phi(S, d, r) = \phi(S, d, d)$



(a) TABS (b) GLBS

FIGURE 2: Tempered aspirations and Gupta-Livne bargaining solutions.

(vi) S-Continuity (S-CONT): when a convergence of sets is evaluated using the Hausdorff topology, for every $\{(S_k, d, r)\}_k \subset \Sigma^N$, such that $\lim_{k\to\infty} S_k = S$ and $(S, d, r) \in \Sigma^N$, $\lim_{k\to\infty} \phi(S_k, d, r) = \phi(S, d, r)$

In 1988, Gupta and Livne proposed another bargaining solution, called the Gupta-Livne Bargaining Solution (GLBS). The solution is "dual" to the TABS in the sense that it exchanges the roles played by the reference and disagreement points. In the Gupta-Livne approach, the disagreement point d has no role to play as a threat in the bargain. It serves only to form the aspirations of the players through the construction of the ideal aspiration point. For every $(S, d, r) \in \Sigma^N$, the GLBS is defined as follows [23]:

$$\begin{aligned} \mathrm{GLBS}(S, d, r) &= [\lambda^* \times a(S, d)] + [(1 - \lambda^*) \times r] \\ \text{s.t.} \quad \lambda^* &= \max \{\lambda \in [0, 1] \mid ([\lambda \times a(S, d)] \\ &\quad + [(1 - \lambda) \times r]) \in S\}. \end{aligned} \tag{3}$$

Gupta and Livne characterize their solution using the already familiar WPO, SYM, and SC.INV, plus the following Relevant Domain, d-REST.S-MON, and LIM.d-SENS axioms [23]:

(i) Relevant Domain (RD): for every $(S, d, r) \in \Sigma^N$, $\phi(S, d, r) = \phi(cch(\{x \in S | x \geq d\}), d, r)$. This property states that the outcome of the negotiation is only affected by points that weakly Pareto-dominate the disagreement point

(ii) dS-Monotonicity (d-REST. S-MON): for every (S, d, r), $(S', d', r') \in \Sigma^N$, $(d, r) = (d', r')$, $S \subseteq S'$, and $a(S, d) = a(S', d')$, imply $\phi(S, d, r) \leq \phi(S', d', r')$. Therefore, as the bargaining set S grows, the corresponding aspirations must remain fixed in order to preserve monotonicity. Originally proposed for standard bargaining problems by Roth, the d-REST. S-MON axiom can be seen as dual to r-REST. S-MON

(iii) Limited$d$-Sensitivity (LIM.d-SENS): for every (S, d, r), $(S', d', r') \in \Sigma^N$, $(S, r) = (S', r')$, and $a(S, d) = a(S', d')$, imply $\phi(S, d, r) = \phi(S', d', r')$. This axiom was originally labeled limited sensitivity to changes in the disagreement point. It says that if the disagreement point changes without altering the corresponding aspirations, then the outcome of the negotiation is the same

The main difference of TABS and GLBS is the role of disagreement point d. In TABS, d is used as a reference vector from which proportional payoffs are measured. However, in GLBS, it is used to set the ideal aspiration point. Both solution concepts are illustrated in Figure 2 [23].

3.3. The Proposed Application Task Offloading Algorithms. In this study, we design two bargaining games for task offloading services. First, the idea of TABS is adopted to implement the time-sensitive application offloading algorithm. To fair-efficiently offload the task computation, the offload decision process for $\mathscr{A}_k^{\mathscr{D}_i}$ is formulated as a cooperative bargaining game $\mathbb{G}_{\mathrm{TABS}} = \{\{\mathscr{D}_i, \mathscr{CL}_j, \mathbb{C}\}, \{\mathfrak{P}^{\mathscr{D}_i}, \mathfrak{P}^{\mathscr{CL}_j}, \mathfrak{P}^{\mathbb{C}}\}, \mathscr{A}_k^{\mathscr{D}_i}, \Gamma^{\mathscr{A}_k}, \{\mathbb{S}_{\mathscr{D}_i}, \mathbb{S}_{\mathscr{CL}_j}, \mathbb{S}_{\mathbb{C}}\}, \{U_{\mathscr{D}_i}, U_{\mathscr{CL}_j}, U_{\mathbb{C}}\}, r_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S, \mathbf{x}), a_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S, \mathbf{x})\}$:

(i) Players: in $\mathbb{G}_{\mathrm{TABS}}$, a smart device $\mathscr{D}_i \in \mathbb{D}$, the corresponding cloudlet $\mathscr{CL}_j \in \mathbb{F}$, and the cloud server $\mathbb{C}$ are assumed as game players $\{\mathscr{D}_i, \mathscr{CL}_j, \mathbb{C}\}$ to process the task offloading service

(ii) Computation powers of players: $\mathscr{D}_i$, $\mathscr{CL}_j$, and $\mathbb{C}$ have $\mathfrak{P}^{\mathscr{D}_i}$, $\mathfrak{P}^{\mathscr{CL}_j}$, and $\mathfrak{P}^{\mathbb{C}}$ computation powers, respectively; they are assumed as total CPU capacities of game players

(iii) Application task and computation amount: the application $\mathscr{A}_k^{\mathscr{D}_i}$ is generated from the $\mathscr{D}_i$, and total computation amount of $\mathscr{A}_k^{\mathscr{D}_i}$ is $\Gamma^{\mathscr{A}_k}$

(iv) Strategies: each player has a finite computation capacity. The set of strategies for each player consists of its discrete computation power levels. Let $\mathbb{S}_{\mathscr{D}_i} = \{\mathfrak{C}_{1\le k\le r}^{\mathscr{D}_i} \le \mathfrak{P}^{\mathscr{D}_i} \mid \{\mathfrak{C}_1^{\mathscr{D}_i}, \cdots, \mathfrak{C}_r^{\mathscr{D}_i}\}\}$ be $\mathscr{D}_i$'s strategy set, $\mathbb{S}_{\mathscr{CL}_j} = \{\mathfrak{C}_{1\le k\le e}^{\mathscr{CL}_j} \le \mathfrak{P}^{\mathscr{CL}_j} \mid \{\mathfrak{C}_1^{\mathscr{CL}_j}, \cdots, \mathfrak{C}_e^{\mathscr{CL}_j}\}\}$ be $\mathscr{CL}_j$'s strategy set, and $\mathbb{S}_{\mathbb{C}} = \{\mathfrak{C}_{1\le k\le l}^{\mathbb{C}} \le \mathfrak{P}^{\mathbb{C}} \mid \{\mathfrak{C}_1^{\mathbb{C}}, \cdots, \mathfrak{C}_l^{\mathbb{C}}\}\}$ be $\mathbb{C}$'s strategy set

(v) Utility functions: $\mathscr{D}_i$, $\mathscr{CL}_j$, and $\mathbb{C}$ players have their own utility functions $U_{\mathscr{D}_i}^{\mathscr{A}_k}$, $U_{\mathscr{CL}_j}^{\mathscr{A}_k}$, and $U_{\mathbb{C}}^{\mathscr{A}_k}$, respectively, to process the offload service of task $\mathscr{A}_k^{\mathscr{D}_i}$. Each utility function maps the player's satisfaction to a real number, which represents the resulting payoff in the game $\mathbb{G}_{\text{TABS}}$

(vi) Reference point: the reference point of $\mathscr{D}_i$, $\mathscr{CL}_j$, and $\mathbb{C}$ is denoted as $r_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x})$; it satisfies two features, namely, (a) $r_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}} \in S/\text{WPO}(S)$, and (b) $r_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}} > d$

(vii) Aspiration point: the aspiration point of $\mathscr{D}_i$, $\mathscr{CL}_j$, and $\mathbb{C}$ is denoted as $a_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x})$; it is defined based on the reference point

To quantify service satisfaction, the utility functions of players in TABS can be derived as follows:

$$\begin{cases} U_{\mathscr{D}_i}^{\mathscr{A}_k}\left(\chi_{\mathscr{D}_i}^{\mathscr{A}_k}\right) = \dfrac{\psi_{\mathscr{D}_i}}{\eta_{\mathscr{D}_i}} \times \log\left(1 + \left(\eta_{\mathscr{D}_i} \times \dfrac{\chi_{\mathscr{D}_i}^{\mathscr{A}_k}}{\Gamma^{\mathscr{A}_k}}\right)\right), \\ U_{\mathscr{CL}_j}^{\mathscr{A}_k}\left(\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}\right) = \dfrac{\psi_{\mathscr{CL}_j}}{\eta_{\mathscr{CL}_j}} \times \log\left(1 + \left(\eta_{\mathscr{CL}_j} \times \dfrac{\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}}{\Gamma^{\mathscr{A}_k}}\right)\right), \\ U_{\mathbb{C}}^{\mathscr{A}_k}\left(\chi_{\mathbb{C}}^{\mathscr{A}_k}\right) = \dfrac{\psi_{\mathbb{C}}}{\eta_{\mathbb{C}}} \times \log\left(1 + \left(\eta_{\mathbb{C}} \times \dfrac{\chi_{\mathbb{C}}^{\mathscr{A}_k}}{\Gamma^{\mathscr{A}_k}}\right)\right), \end{cases}$$
$$\text{s.t.} \quad \eta_{\mathscr{D}_i} = \frac{\beta_C^{\mathscr{D}_i}}{\mathfrak{P}^{\mathscr{D}_i}}$$
$$\eta_{\mathscr{CL}_j} = \frac{\beta_C^{\mathscr{CL}_j}}{\mathfrak{P}^{\mathscr{CL}_j}}$$
$$\eta_{\mathbb{C}} = \frac{\beta_C^{\mathbb{C}}}{\mathfrak{P}^{\mathbb{C}}}$$
$$\Gamma^{\mathscr{A}_k} = \left(\chi_{\mathscr{D}_i}^{\mathscr{A}_k} + \chi_{\mathscr{CL}_j}^{\mathscr{A}_k} + \chi_{\mathbb{C}}^{\mathscr{A}_k}\right). \tag{4}$$

where $\chi_{\mathscr{D}_i}^{\mathscr{A}_k}$, $\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}$, and $\chi_{\mathbb{C}}^{\mathscr{A}_k}$ are assigned computation amounts to $\mathscr{D}_i$, $\mathscr{CL}_j$, and $\mathbb{C}$, respectively. $\psi_{\mathscr{D}_i}$, $\psi_{\mathscr{CL}_j}$, and $\psi_{\mathbb{C}}$ are coefficient parameters to represent the QoE of $\mathscr{D}_i$, $\mathscr{CL}_j$, and $\mathbb{C}$ computation services, respectively. $\beta_C^{\mathscr{D}_i}$, $\beta_C^{\mathscr{CL}_j}$, and $\beta_C^{\mathbb{C}}$ are the current computation loads of $\mathscr{D}_i$, $\mathscr{CL}_j$, and $\mathbb{C}$, respectively. In the developed bargaining game, each player is a member of a team willing to compromise with other players. According to their utility functions and expected payoffs, team players make a collective decision to gain a total optimal solution. In $\mathbb{G}_{\text{TABS}}$, the reference point, i.e., $r_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x})$, is defined as follows:

$$r_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x}) = \left(U_{\mathscr{D}_i}^r, U_{\mathscr{CL}_j}^r, U_{\mathbb{C}}^r\right)$$
$$\text{s.t.} \quad U_{\mathscr{D}_i}^r = \frac{U_{\mathscr{D}_i}^{\mathscr{A}_k}\left(m^{\mathscr{A}_k}\right)}{\varphi_{\mathscr{D}}}$$
$$U_{\mathscr{CL}_j}^r = \frac{U_{\mathscr{CL}_j}^{\mathscr{A}_k}\left(m^{\mathscr{A}_k}\right)}{\varphi_{\mathscr{CL}}}$$
$$U_{\mathbb{C}}^r\left(\chi_{\mathbb{C}}^{\mathscr{A}_k}\right) = \frac{U_{\mathbb{C}}^{\mathscr{A}_k}\left(m^{\mathscr{A}_k}\right)}{\varphi_{\mathbb{C}}}, \tag{5}$$

where $\varphi_{\mathscr{D}}$, $\varphi_{\mathscr{CL}}$, and $\varphi_{\mathbb{C}}$ are the control factors to decide the reference point values of $\mathscr{D}_i$, $\mathscr{CL}_j$, and $\mathbb{C}$, respectively. $m^{\mathscr{A}_k}$ is the minimum computation capacity for the $\mathscr{A}_k$ task offloading service. In $\mathbb{G}_{TABS}$, the aspiration point of TABS, i.e., $a_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x})$, is defined as follows:

$$a_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x}) = \left(a_{\mathscr{D}_i}(S,\mathbf{x}), a_{\mathscr{CL}_j}(S,\mathbf{x}), a_{\mathbb{C}}(S,\mathbf{x})\right)$$
$$\text{s.t.} \quad \begin{cases} a_{\mathscr{D}_i}(S,\mathbf{x}) = \max\left\{t \in \mathbb{R} \mid \left(t, U_{\mathscr{CL}_j}^r, U_{\mathbb{C}}^r\right) \in S\right\} \\ a_{\mathscr{CL}_j}(S,\mathbf{x}) = \max\left\{t \in \mathbb{R} \mid \left(t, U_{\mathscr{D}_i}^r, U_{\mathbb{C}}^r\right) \in S\right\} \\ a_{\mathbb{C}}(S,\mathbf{x}) = \max\left\{t \in \mathbb{R} \mid \left(t, U_{\mathscr{D}_i}^r, U_{\mathscr{CL}_j}^r\right) \in S\right\}. \end{cases} \tag{6}$$

Based on the disagreement point d as a starting point, the line (**L**) forward of the aspiration point $a_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x})$ is defined as follows:

$$\mathbf{L} = \left\{\mathbf{U} \middle| \frac{U_{\mathscr{D}_i}^{\mathscr{A}_k}\left(\chi_{\mathscr{D}_i}^{\mathscr{A}_k}\right)}{a_{\mathscr{D}_i}(S,\mathbf{x})} = \frac{U_{\mathscr{CL}_j}^{\mathscr{A}_k}\left(\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}\right)}{a_{\mathscr{CL}_j}(S,\mathbf{x})} = \frac{U_{\mathbb{C}}^{\mathscr{A}_k}\left(\chi_{\mathbb{C}}^{\mathscr{A}_k}\right)}{a_{\mathbb{C}}(S,\mathbf{x})}\right\}. \tag{7}$$

Simply, we can think that TABS is a weak Pareto-optimal solution located in S as well as in line **L** in (7). Geometrically, TABS is the intersection point $(U_{\mathscr{D}_i}^{\mathscr{A}_k}(\chi_{\mathscr{D}_i}^*), U_{\mathscr{CL}_j}^{\mathscr{A}_k}(\chi_{\mathscr{CL}_j}^*), U_{\mathbb{C}}^{\mathscr{A}_k}(\chi_{\mathbb{C}}^*))$ between the bargaining set S and line **L**. Therefore, TABS must satisfy

$$\left(\frac{U_{\mathscr{D}_i}^{\mathscr{A}_k}\left(\chi_{\mathscr{D}_i}^{*}\right)}{a_{\mathscr{D}_i}(S,\mathbf{x})}\right)=\left(\frac{U_{\mathscr{CL}_j}^{\mathscr{A}_k}\left(\chi_{\mathscr{CL}_j}^{*}\right)}{a_{\mathscr{CL}_j}(S,\mathbf{x})}\right)=\left(\frac{U_{\mathbb{C}}^{\mathscr{A}_k}(\chi_{\mathbb{C}}^{*})}{a_{\mathbb{C}}(S,\mathbf{x})}\right). \quad (8)$$

Second, the idea of GLBS is adopted to develop the computation-intensive application offloading algorithm. To adaptively offload the delay-tolerant task computation, the offload decision process for $\mathscr{A}_k^{\mathscr{D}_i}$ is formulated as another cooperative game model $\mathbb{G}_{\text{GLBS}}=\{\{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}\},\{\mathfrak{P}^{\mathscr{D}_i},\mathfrak{P}^{\mathscr{CL}_j},\mathfrak{P}^{\mathbb{C}}\},\mathscr{A}_k^{\mathscr{D}_i},\Gamma^{\mathscr{A}_k},\{\mathbb{S}_{\mathscr{D}_i},\mathbb{S}_{\mathscr{CL}_j},\mathbb{S}_{\mathbb{C}}\},\{U_{\mathscr{D}_i},U_{\mathscr{CL}_j},U_{\mathbb{C}}\},r_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x}),a_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x})\}$. In the $\mathbb{G}_{\text{GLBS}}$ game, only utility functions and aspiration points are defined differently, and the other game elements are the same as $\mathbb{G}_{\text{TABS}}$. In $\mathbb{G}_{\text{GLBS}}$, $a_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x})$ is dynamically calculated according to (1), and $\mathscr{D}_i$, $\mathscr{CL}_j$, and $\mathbb{C}$'s utility functions for the task $\mathscr{A}_k^{\mathscr{D}_i}$ can be derived as follows:

$$U_{\mathscr{D}_i}^{\mathscr{A}_k}\left(\chi_{\mathscr{D}_i}^{\mathscr{A}_k}\right)=\frac{\psi_{\mathscr{D}_i}}{\eta_{\mathscr{D}_i}}\times\log\left(1+\left(\eta_{\mathscr{D}_i}\times\frac{\chi_{\mathscr{D}_i}^{\mathscr{A}_k}}{\Gamma^{\mathscr{A}_k}}\right)\right)\times\mathscr{F}_{\mathscr{D}_i}^{\mathscr{A}_k}\left(\chi_{\mathscr{D}_i}^{\mathscr{A}_k}\right)$$

$$\text{s.t.}\quad \mathscr{F}_{\mathscr{D}_i}^{\mathscr{A}_k}\left(\chi_{\mathscr{D}_i}^{\mathscr{A}_k}\right)=\begin{cases}\dfrac{\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}}{\left(1+\left(\omega_{\mathscr{D}_i}^{\mathscr{A}_k}/\sigma\right)\right)\times\Gamma^{\mathscr{A}_k}}, & \text{if}\left(\omega_{\mathscr{D}_i}^{\mathscr{A}_k}\times\dfrac{\chi_{\mathscr{D}_i}^{\mathscr{A}_k}}{m^{\mathscr{A}_k}}\right)\le\left(\dfrac{\chi_{\mathscr{D}_i}^{\mathscr{A}_k}}{\Gamma^{\mathscr{A}_k}}\times T^{\mathscr{A}_k}\right),\\ 0, & \text{otherwise,}\end{cases} \quad (9)$$

$$U_{\mathscr{CL}_j}^{\mathscr{A}_k}\left(\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}\right)=\frac{\psi_{\mathscr{CL}_j}}{\eta_{\mathscr{CL}_j}}\times\log\left(1+\left(\eta_{\mathscr{CL}_j}\times\frac{\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}}{\Gamma^{\mathscr{A}_k}}\right)\right)\times\mathscr{F}_{\mathscr{CL}_j}^{\mathscr{A}_k}\left(\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}\right)$$

$$\text{s.t.}\quad \mathscr{F}_{\mathscr{CL}_j}^{\mathscr{A}_k}\left(\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}\right)=\begin{cases}\dfrac{\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}}{\left(1+\left(\left(\left(\xi_{\mathscr{CL}_j}^{\mathscr{D}_i}+\omega_{\mathscr{CL}_j}^{\mathscr{A}_k}\right)+\omega_{\mathscr{D}_i}^{\mathscr{A}_k}\right)/\sigma\right)\right)\times\Gamma^{\mathscr{A}_k}}, & \text{if}\left(\left(\left(\xi_{\mathscr{CL}_j}^{\mathscr{D}_i}+\omega_{\mathscr{CL}_j}^{\mathscr{A}_k}\right)+\omega_{\mathscr{D}_i}^{\mathscr{A}_k}\right)\times\dfrac{\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}}{m^{\mathscr{A}_k}}\right)\le\left(\dfrac{\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}}{\Gamma^{\mathscr{A}_k}}\times T^{\mathscr{A}_k}\right),\\ 0, & \text{otherwise,}\end{cases} \quad (10)$$

$$U_{\mathbb{C}}^{\mathscr{A}_k}\left(\chi_{\mathbb{C}}^{\mathscr{A}_k}\right)=\frac{\psi_{\mathbb{C}}}{\eta_{\mathbb{C}}}\times\log\left(1+\left(\eta_{\mathbb{C}}\times\frac{\chi_{\mathbb{C}}^{\mathscr{A}_k}}{\Gamma^{\mathscr{A}_k}}\right)\right)\times\mathscr{F}_{\mathbb{C}}^{\mathscr{A}_k}\left(\chi_{\mathbb{C}}^{\mathscr{A}_k}\right)$$

$$\text{s.t.}\quad \mathscr{F}_{\mathbb{C}}^{\mathscr{A}_k}\left(\chi_{\mathbb{C}}^{\mathscr{A}_k}\right)=\begin{cases}\dfrac{\chi_{\mathbb{C}}^{\mathscr{A}_k}}{\left(1+\left(\left(\left(\xi_{\mathbb{C}}^{\mathscr{D}_i}+\omega_{\mathbb{C}}^{\mathscr{A}_k}\right)+\omega_{\mathscr{D}_i}^{\mathscr{A}_k}\right)/\sigma\right)\right)\times\Gamma^{\mathscr{A}_k}}, & \text{if}\left(\left(\left(\xi_{\mathbb{C}}^{\mathscr{D}_i}+\omega_{\mathbb{C}}^{\mathscr{A}_k}\right)+\omega_{\mathscr{D}_i}^{\mathscr{A}_k}\right)\times\dfrac{\chi_{\mathbb{C}}^{\mathscr{A}_k}}{m^{\mathscr{A}_k}}\right)\le\left(\dfrac{\chi_{\mathbb{C}}^{\mathscr{A}_k}}{\Gamma^{\mathscr{A}_k}}\times T^{\mathscr{A}_k}\right),\\ 0, & \text{otherwise.}\end{cases} \quad (11)$$

where $\omega_{\mathscr{D}_i}^{\mathscr{A}_k}$, $\omega_{\mathscr{CL}_j}^{\mathscr{A}_k}$, and $\omega_{\mathbb{C}}^{\mathscr{A}_k}$ are computation delay factors of $\mathscr{D}_i$, $\mathscr{CL}_j$, and $\mathbb{C}$, respectively. $\xi_{\mathscr{CL}_j}^{\mathscr{D}_i}$ and $\xi_{\mathbb{C}}^{\mathscr{D}_i}$ are communication delay factors of $\mathscr{CL}_j$ and $\mathbb{C}$, respectively. σ is the system's basic time unit for the task offloading service. $T^{\mathscr{A}_k}$ is the time deadline of $\mathscr{A}_k$. Based on the reference point $r_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x})$ as a starting point, the line (**L**) forward of the aspiration point $a_{\mathscr{D}_i,\mathscr{CL}_j,\mathbb{C}}(S,\mathbf{x})$ is defined as follows:

$$\mathbf{L}=\left\{\mathbf{U}\left|\frac{U_{\mathscr{D}_i}^{\mathscr{A}_k}\left(\chi_{\mathscr{D}_i}^{\mathscr{A}_k}\right)-U_{\mathscr{D}_i}^{r}\left(\chi_{\mathscr{D}_i}^{\mathscr{A}_k}\right)}{a_{\mathscr{D}_i}(S,\mathbf{x})-U_{\mathscr{D}_i}^{r}\left(\chi_{\mathscr{D}_i}^{\mathscr{A}_k}\right)}=\frac{U_{\mathscr{CL}_j}^{\mathscr{A}_k}\left(\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}\right)-U_{\mathscr{CL}_j}^{r}\left(\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}\right)}{a_{\mathscr{CL}_j}(S,\mathbf{x})-U_{\mathscr{CL}_j}^{r}\left(\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}\right)}=\frac{U_{\mathbb{C}}^{\mathscr{A}_k}\left(\chi_{\mathbb{C}}^{\mathscr{A}_k}\right)-U_{\mathbb{C}}^{r}\left(\chi_{\mathbb{C}}^{\mathscr{A}_k}\right)}{a_{\mathbb{C}}(S,\mathbf{x})-U_{\mathbb{C}}^{r}\left(\chi_{\mathbb{C}}^{\mathscr{A}_k}\right)}\right\}. \quad (12)$$

Simply, we can think that GLBS is a weak Pareto-optimal solution located in S as well as in line $\mathbf{L}$ in (12). Geometrically, GLBS is the intersection point $(U_{\mathscr{D}_i}^{\mathscr{A}_k}(\chi_{\mathscr{D}_i}^*), U_{\mathscr{CL}_j}^{\mathscr{A}_k}(\chi_{\mathscr{CL}_j}^*), U_{\mathbb{C}}^{\mathscr{A}_k}(\chi_{\mathbb{C}}^*))$ between the bargaining set S and line $\mathbf{L}$. Therefore, GLBS must satisfy

$$\frac{\left(U_{\mathscr{D}_i}^{\mathscr{A}_k}\left(\chi_{\mathscr{D}_i}^*\right) - U_{\mathscr{D}_i}^r\left(\chi_{\mathscr{D}_i}^{\mathscr{A}_k}\right)\right)}{\left(a_{\mathscr{D}_i}(S,\mathbf{x}) - U_{\mathscr{D}_i}^r\left(\chi_{\mathscr{D}_i}^{\mathscr{A}_k}\right)\right)} = \frac{\left(U_{\mathscr{CL}_j}^{\mathscr{A}_k}\left(\chi_{\mathscr{CL}_j}^*\right) - U_{\mathscr{CL}_j}^r\left(\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}\right)\right)}{\left(a_{\mathscr{CL}_j}(S,\mathbf{x}) - U_{\mathscr{CL}_j}^r\left(\chi_{\mathscr{CL}_j}^{\mathscr{A}_k}\right)\right)} = \frac{\left(U_{\mathbb{C}}^{\mathscr{A}_k}(\chi_{\mathbb{C}}^*) - U_{\mathbb{C}}^r\left(\chi_{\mathbb{C}}^{\mathscr{A}_k}\right)\right)}{\left(a_{\mathbb{C}}(S,\mathbf{x}) - U_{\mathbb{C}}^r\left(\chi_{\mathbb{C}}^{\mathscr{A}_k}\right)\right)}. \tag{13}$$

3.4. Main Steps of Proposed Task Offloading Algorithm. In this study, we design a novel task offloading scheme for different kinds of applications, which can be categorized into two classes according to the required QoE: computation-intensive or time-sensitive applications. Different types of application services over future network systems not only require different QoE but also need different network control strategies. Based on different application characteristics, we dynamically select the most adaptable bargaining solution to address the task offloading problem. In the proposed scheme, the basic concepts of TABS and GLBS are adopted to distribute the computation amount of each application task. Computation-intensive but delay-tolerant applications can be ultimately executed without offloading services. Therefore, it is reasonable that the task offloading bargaining solution is measured based on the reference point as a starting point; GLBS is suitable for these services. For time-sensitive and delay-constrained applications, it is worthless if we cannot meet the time deadlines of applications. Therefore, it is appropriate that the task offloading bargaining solution is measured based on the disagreement point as a starting point; TABS is appropriate for these services. By a sophisticated combination of these two bargaining solutions, our cooperative game-based approach approximates a well-balanced performance among conflicting requirements. The primary steps of the proposed scheme are described as follows, and they are described by the following Figure 3:

Step 1. Control parameters and system factors are determined by the simulation scenario in Section 4 and Table 1.

Step 2. At each time period, individual mobile devices $\mathscr{D}$ generate application tasks; different kinds of applications are equally generated.

Step 3. If a computation-intensive application $\mathscr{A}$ is generated, the GLBS is used to process the task offloading service. According to (1), (3), (9)-(12), and (13), the computation amount $\Gamma^{\mathscr{A}}$ of an application task is effectively distributed to $\mathscr{D}$, $\mathscr{CL}$, and $\mathbb{C}$.

Step 4. If a time-sensitive application $\mathscr{A}$ is generated, TABS is used to process the task offloading service. According to (2), (4)–(7), and (8), the computation amount $\Gamma^{\mathscr{A}}$ of an application task is dynamically distributed to $\mathscr{D}$, $\mathscr{CL}$, and $\mathbb{C}$.

Step 5. Based on the interactive process, the current computation loads of a device, a cloudlet, and cloud server, i.e., $\beta_C^{\mathscr{D}}$, $\beta_C^{\mathscr{CL}}$, and $\beta_C^{\mathbb{C}}$, respectively, are monitored in a real-time online manner. This information is used to calculate utility functions of each game players.

Step 6. The system is constantly self-monitoring the current network situation. If a new task offloading service is requested, it can retrigger a new bargaining process; the system proceeds to Step 3 for the next game iteration.

4. Performance Evaluation

4.1. Simulation Setup. In this section, we evaluate the performance of our proposed protocol and compare it with that of the *JRCRA* [15], *HFCCO* [16], *FCOWA* [17] schemes. To ensure a fair comparison, the following assumptions and system scenario are used:

(i) The simulated hierarchical network system consists of 10 cloudlets ($m = 10$) and 100 mobile devices ($n = 100$).

(ii) In the offered application load situation, the arrival process for new application requests is the rate of the Poisson process (ρ). The offered range is varied from 0 to 3.0

(iii) Mobile devices are distributed randomly over the network coverage area, and we assume the absence of physical obstacles in the experiments

(iv) For mobile device, cloudlet, and cloud computation capacities, i.e., $\mathfrak{P}^{\mathscr{D}}$, $\mathfrak{P}^{\mathscr{CL}}$, and $\mathfrak{P}^{\mathbb{C}}$, we assume their CPU computing powers. They are 5 GHz, 100 GHz, and 1000 GHz per second, respectively

(v) Each mobile device selects its corresponding cloudlet at the closest distance for the task offloading service

(vi) We assume that 10% of $\mathfrak{P}^{\mathscr{D}}$, $\mathfrak{P}^{\mathscr{CL}}$, and $\mathfrak{P}^{\mathbb{C}}$ may be consumed to sustain the basic operations of a mobile device, a cloudlet, and a cloud server

(vii) Computation-intensive applications and time-sensitive applications are equally generated

(viii) To reduce the computation complexity, the computation amount is specified in terms of the basic computation unit, i.e., m, where one m is the minimum computation capacity (e.g., 100 MHz) for the offloading service. Therefore, for practical implementations, the computation amount distribution is negotiated discretely by the size of one m

(ix) The hierarchical network system performance measures obtained on the basis of 100 simulation

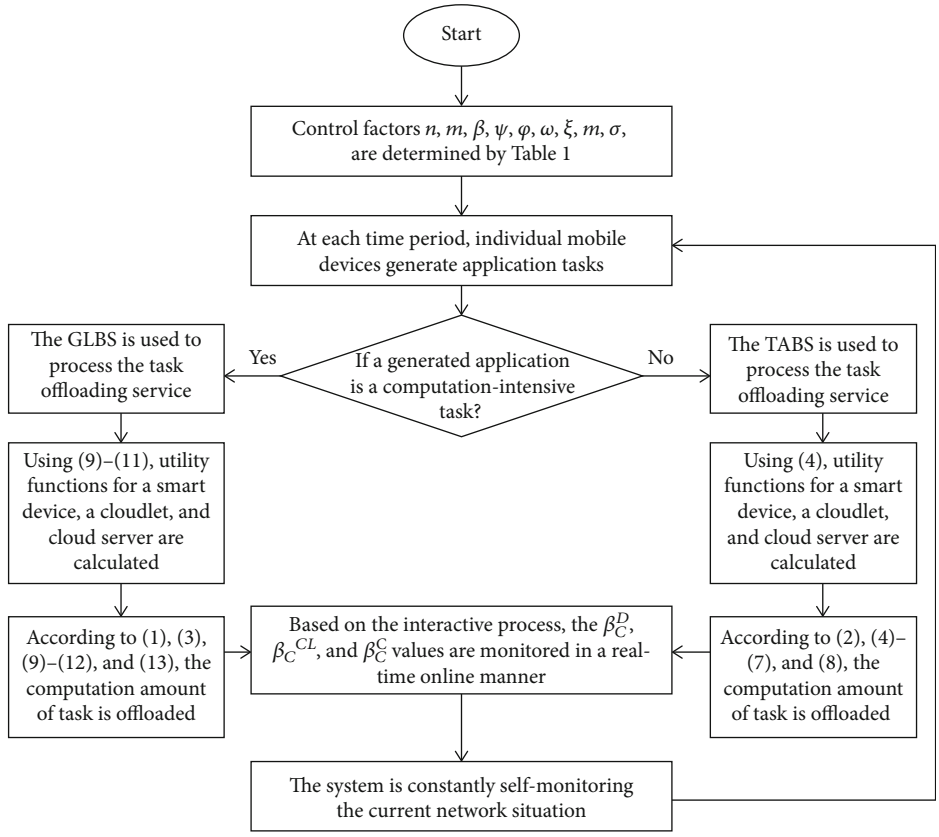


Figure 3: Flowchart of the proposed algorithm.

Table 1: System parameters used in the simulation experiments.

(a)

Parameter	Value	Description
n, m	100, 10	Total number of mobile devices and fog nodes
$\mathfrak{P}^{\mathscr{D}}$, $\mathfrak{P}^{\mathscr{CL}}$, $\mathfrak{P}^{\mathbb{C}}$	5, 100, 1000 GHz/s	The computation capacities of $\mathscr{D}$, $\mathscr{CL}$, and $\mathbb{C}$, respectively
$\psi_{\mathscr{D}}$, $\psi_{\mathscr{CL}}$, $\psi_{\mathbb{C}}$	1.5, 1.75, 2	Coefficient QoE parameters of $\mathscr{D}$, $\mathscr{CL}$, and $\mathbb{C}$, respectively
$\varphi_{\mathscr{D}}$, $\varphi_{\mathscr{CL}}$, $\varphi_{\mathbb{C}}$	0.2, 0.5, 1	Control factors to decide the reference point value $r_{\mathscr{D},\mathscr{CL},\mathbb{C}}(S,x)$
$\omega_{\mathscr{D}}^{\mathscr{A}}$, $\omega_{\mathscr{CL}}^{\mathscr{A}}$, $\omega_{\mathbb{C}}^{\mathscr{A}}$	125, 75, 50 msec	Computation delay factors of $\mathscr{D}$, $\mathscr{CL}$, and $\mathbb{C}$, respectively
$\xi_{\mathscr{CL}}^{\mathscr{D}}$, $\xi_{\mathbb{C}}^{\mathscr{D}}$	100, 200 msec	Communication delay factors of $\mathscr{CL}$ and $\mathbb{C}$, respectively
m	100 MHz	Basic computation unit for computation offloading service
σ	1 second	The system's basic time-unit for the task offloading service

(b)

Application type	Application task	Computation amount ($\Gamma^{\mathscr{A}_k}$)	Time deadline ($T^{\mathscr{A}_k}$)
Computation-intensive applications	$\mathscr{A}_k \in$ I	300 GHz	N/A
	$\mathscr{A}_k \in$ II	400 GHz	N/A
	$\mathscr{A}_k \in$ III	500 GHz	N/A
Time-sensitive applications	$\mathscr{A}_k \in$ IV	250 GHz	5 seconds
	$\mathscr{A}_k \in$ V	450 GHz	10 seconds
	$\mathscr{A}_k \in$ VI	900 GHz	15 seconds

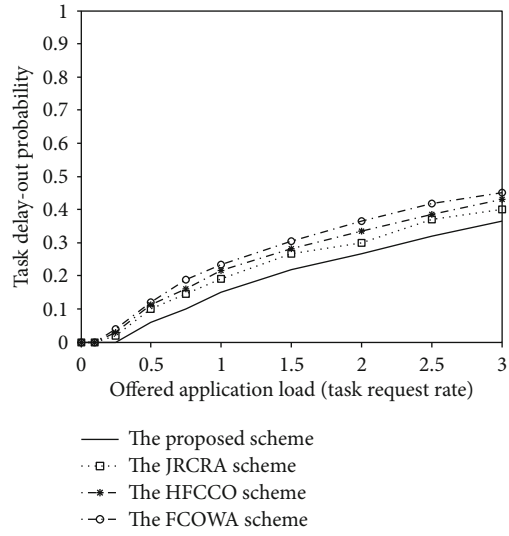


Figure 4: Task delay-out probability.

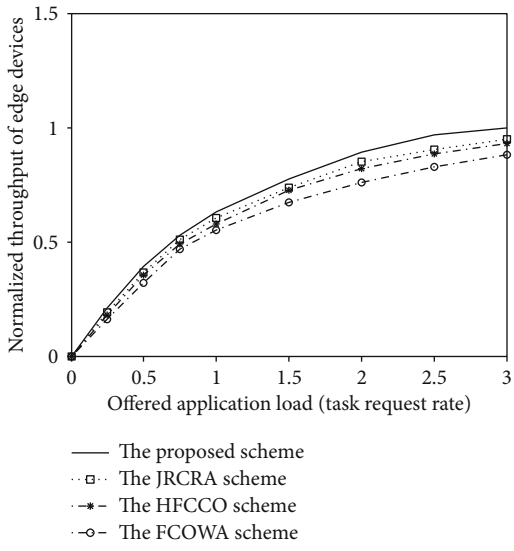


Figure 5: Normalized throughput of edge devices.

runs are plotted as functions of the Poisson process (ρ)

To demonstrate the validity of our proposed method, we measured the task delay-out probability, normalized throughput of edge devices, and fairness of edge devices for their payoffs. Table 1 shows the control parameters and system factors used in the simulation. These parameters and factors have their own characteristics.

5. Results and Discussion

In Figure 4, we evaluate the task delay-out probability under four methods. As a criterion of QoE assessment, the task delay-out probability is a measurement of how many application tasks fail to meet their time delay constraints. It is a key performance evaluation factor in the future network operation. The fail ratio of all schemes is increasing with the increase of the task request rate. It is reasonable since the higher task requests lead to the system resource exhaustion, thus making the task delay-out probability increases. However, we observe that there is a considerable performance excellence in the proposed scheme. Our bargaining-based approach can fair-efficiently share the future network system resource to improve the service quality. Therefore, we can maintain the stable performance superiority under different application load intensities.

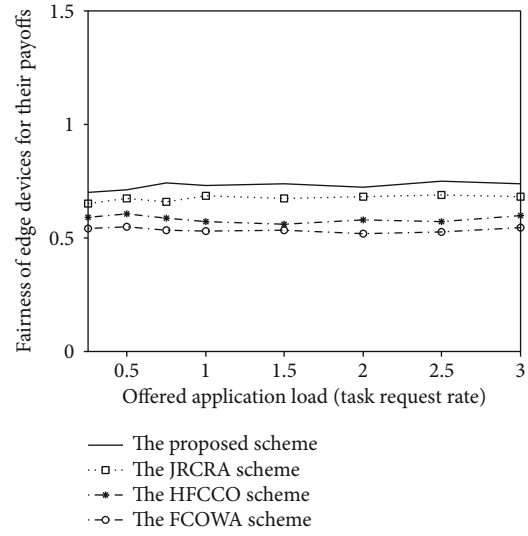


Figure 6: Fairness of edge devices for their payoffs.

Normalized throughput of edge devices, which is displayed in Figure 5, represents the resource efficiency of the hierarchical network system. This is another main criterion on the performance evaluation. As can be observed, the performance trend of all schemes is similar. Typically, a higher system throughput can increase the network capacity; it is more profitable for the system operator. In the proposed scheme, each smart device adaptively offloads its tasks to the fog node and cloud server based on a proper bargaining solution. Especially, we explore the reciprocal combination of the *GLBS* and *TABS* methods to balance contradictory requirements. Under dynamic network system environments, the possible advantages of our approach include adaptability, flexibility, and responsiveness to current network system conditions. Therefore, we can effectively manage the three-layer hierarchical network system resource while satisfying desirable features, which are defined as axioms of a selected bargaining solution. Due to this reason, we can actually distribute the system resource to increase the throughput of mobile edge devices than the existing *JRCRA*, *HFCCO*, and *FCOWA* schemes.

Figure 6 depicts the fairness among edge devices. Fairness is a prominent issue for the operation of traffic intensive networks, and it is analogous to the social welfare for the resource allocation problem. Especially, under heavy application load environments, fairness is a highly desirable property for different edge devices. To characterize the fairness notion, we follow Jain's fairness index [25], which has been

frequently used to measure the fairness in network management. In the proposed scheme, we adopt the basic idea of *TABS* and *GLBS*, and share the system resource fairly while satisfying their fair-oriented axioms. Therefore, in our proposed scheme, the actual outcome is fairly dealt out among individual edge devices. As shown in Figure 6, the profit-sharing fairness in our approach is distinctly better compared to the existing schemes, which are designed as lopsided and one-way methods and do not effectively consider the fairness issue.

The simulation results shown in Figures 4–6 demonstrate that the proposed scheme can attain an appropriate performance balance. In contrast, the *JRCRA* [15], *HFCCO* [16], and *FCOWA* [17] schemes cannot offer this outcome under widely different network application request situations.

6. Summary and Conclusions

In this paper, we investigate the application task offloading problem based on the edge, fog, and cloud computing paradigms. According to the 3-tier network hierarchy, i.e., mobile device-cloudlet-cloud infrastructure, the task offloading problem is formulated and addressed by using the cooperative bargaining game concept. Especially, we practically apply the *TABS* and *GLBS* methods to effectively offload the computation amount of each application task. By jointly considering the computation intensity and delay sensitivity, we adaptively select the most suitable bargaining method in an intelligent manner. For the evolution of the future network application services, our bargaining-game-based approach is attractive and appropriate to operate the real-world network system. The performance evaluations are presented to illustrate the effectiveness of the proposed scheme and demonstrate the superior performance over the existing *JRCRA*, *HFCCO*, and *FCOWA* schemes.

In the future, we would like to consider privacy issues such as the differential privacy during the task offloading operation. Further, we will investigate the mobile device mobility to excellently adapt the dynamic network environments. In that case, the required information exchange and communication overhead need to be carefully investigated. In addition, we will extend the scenario from one cloudlet fog node to multiple cloudlets fog nodes when an individual application task is offloaded. For this future work, interference management, control overhead and load balancing will be considered.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author, Sungwook Kim, declares that there are no competing interests regarding the publication of this paper.

Authors' Contributions

The author, Sungwook Kim, is the sole contributor to this research work.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2018-0-01799) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation), and was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A1A09081759).

References

[1] D. Wang, D. Chen, B. Song, N. Guizani, X. Yu, and X. Du, "From IoT to 5G I-IoT: the next generation IoT-based intelligent algorithms and 5G technologies," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 114–120, 2018.

[2] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob, and M. Imran, "The role of edge computing in internet of things," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 110–115, 2018.

[3] W. Li, Y. Zhao, S. Lu, and D. Chen, "Mechanisms and challenges on mobility-augmented service provisioning for mobile cloud computing," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 89–97, 2015.

[4] Y.-H. Son and K.-C. Lee, "Cloud of things based on linked data," in *2018 International Conference on Information Networking (ICOIN)*, Chiang Mai, Thailand, January 2018.

[5] S. M. A. Oteafy and H. S. Hassanein, "IoT in the fog: a roadmap for data-centric IoT development," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 157–163, 2018.

[6] M. Funk, "Designing the fog: towards an intranet of things," in *CHI Workshop on Interacting with Smart Objects*, pp. 31–38, Montreal, Canada, 2018.

[7] H. El-Sayed, S. Sankar, M. Prasad et al., "Edge of things: the big picture on the integration of edge, IoT and the cloud in a distributed computing environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2018.

[8] R. Yu, G. Xue, V. T. Kilari, and X. Zhang, "The fog of things paradigm: road toward on-demand internet of things," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 48–54, 2018.

[9] Y. Shoham, "Computer science and game theory," *Communications of the ACM*, vol. 51, no. 8, pp. 74–79, 2008.

[10] G. Zhang, F. Shen, Z. Liu, Y. Yang, K. Wang, and M.-T. Zhou, "FEMTO: fair and energy-minimized task offloading for fog-enabled IoT networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4388–4400, 2019.

[11] X. Gao, X. Huang, S. Bian, Z. Shao, and Y. Yang, "PORA: predictive offloading and resource allocation in dynamic fog computing systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 72–87, 2020.

[12] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 998–1010, 2018.

[13] T. Wang, J. Zhou, A. Liu, M. Z. A. Bhuiyan, G. Wang, and W. Jia, "Fog-based computing and storage offloading for data

synchronization in IoT," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4272–4282, 2019.

[14] J. Kim, T. Ha, W. Yoo, and J.-M. Chung, "Task popularity-based energy minimized computation offloading for fog computing wireless networks," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1200–1203, 2019.

[15] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint radio and computational resource allocation in IoT fog computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7475–7484, 2018.

[16] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: a computation offloading game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018.

[17] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing Towards balanced delay and power consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, 2016.

[18] M. Chen, Y. Miao, H. Gharavi, L. Hu, and I. Humar, "Intelligent traffic adaptive resource allocation for edge computing-based 5G networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 499–508, 2020.

[19] Y. Hao, Y. Jiang, T. Chen, D. Cao, and M. Chen, "iTaskOffloading: intelligent task offloading for a cloud-edge collaborative system," *IEEE Network*, vol. 33, no. 5, pp. 82–88, 2019.

[20] M. Chen, W. Li, G. Fortino, Y. Hao, L. Hu, and I. Humar, "A dynamic service migration mechanism in edge cognitive computing," *ACM Transactions on Internet Technology*, vol. 19, no. 2, pp. 1–15, 2019.

[21] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.

[22] M. Chen, Y. Hao, C.-F. Lai, D. Wu, Y. Li, and K. Hwang, "Opportunistic task scheduling over co-located clouds in mobile environment," *IEEE Transactions on Services Computing*, vol. 11, no. 3, pp. 549–561, 2018.

[23] P. V. S. Balakrishnan, J. C. Gómez, and R. V. Vohra, "The tempered aspirations solution for bargaining problems with a reference point," *Mathematical Social Sciences*, vol. 62, no. 3, pp. 144–150, 2011.

[24] M. A. Hinojosa, A. M. Mármol, and J. M. Zarzuelo, "Inequality averse multi-utilitarian bargaining solutions," *International Journal of Game Theory*, vol. 37, no. 4, pp. 597–618, 2008.

[25] M. Dianati, X. Shen, and S. Naik, "A new fairness index for radio resource allocation in wireless networks," in *IEEE Wireless Communications and Networking Conference, 2005*, pp. 712–715, New Orleans, LA, USA, March 2005.