Efficient Resource Allocation for On-Demand Mobile-Edge Cloud Computing

Xu Chen , *Member, IEEE*, Wenzhong Li, *Member, IEEE*, Sanglu Lu, *Member, IEEE*, Zhi Zhou, and Xiaoming Fu, *Senior Member, IEEE*

Abstract—Mobile-edge cloud computing is a new paradigm to provide cloud computing capabilities at the edge of pervasive radio access networks in close proximity to mobile users. Aiming at provisioning flexible on-demand mobile-edge cloud service, in this paper we propose a comprehensive framework consisting of a resource-efficient computation offloading mechanism for users and a joint communication and computation (JCC) resource allocation mechanism for network operator. Specifically, we first study the resource-efficient computation offloading problem for a user, in order to reduce user's resource occupation by determining its optimal communication and computation resource profile with minimum resource occupation and meanwhile satisfying the QoS constraint. We then tackle the critical problem of user admission control for JCC resource allocation, in order to properly select the set of users for resource demand satisfaction. We show the admission control problem is NP-hard, and hence develop an efficient approximation solution of a low complexity by carefully designing the user ranking criteria and rigourously derive its performance guarantee. To prevent the manipulation that some users may untruthfully report their valuations in acquiring mobile-edge cloud service, we further resort to the powerful tool of critical value approach to design truthful pricing scheme for JCC resource allocation. Extensive performance evaluation demonstrates that the proposed schemes can achieve superior performance for on-demand mobile-edge cloud computing.

Index Terms—Computation offloading, edge computing, joint communication and computation optimization, on-demand resource allocation.

I. INTRODUCTION

A S SMARTPHONES are gaining enormous popularity, more and more new mobile applications such as face

X. Chen and Z. Zhou are with the School of Data and Computer Science, Sun Yat-sen University and Guangdong Key Laboratory for Big Data Analysis and Simulation of Public Opinion, Guangzhou 510006, China (e-mail: chenxu35@mail.sysu.edu.cn; zhouzhi9@mail.sysu.edu.cn).

W. Li and S. Lu are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China (e-mail: lwz@nju.edu.cn; sanglu@nju.edu.cn).

X. Fu is with the Institute of Computer Science, University of Goettingen, Goettingen 37083, Germany (e-mail: fu@cs.uni-goettingen.de).
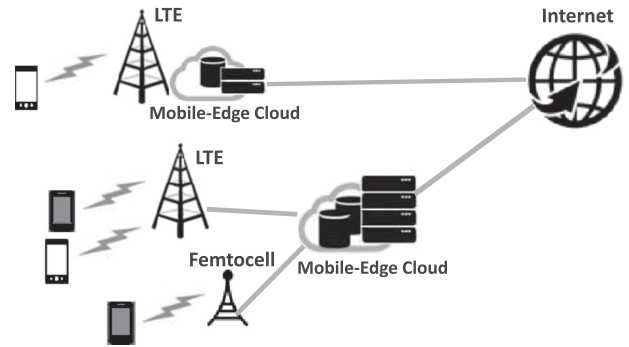
Fig. 1. An illustration of mobile-edge cloud computing.

recognition, natural language processing, interactive gaming, and augmented reality are emerging and attract great attention [1]–[3]. This kind of mobile applications are typically resource-hungry, demanding intensive computation and real-time responsiveness. Due to the physical size constraint, however, mobile devices are in general resource-constrained, having limited computation resources. The tension between resource-hungry applications and resource-constrained mobile devices hence poses a significant challenge for the future mobile platform development.

Mobile cloud computing is envisioned as a promising approach to address such a challenge. By offloading the computation via wireless access to the resource-rich cloud infrastructure, mobile cloud computing can augment the capabilities of mobile devices for resource-hungry applications. Currently, one common approach for mobile cloud computing is to offload the computation-intensive tasks to the remote public cloud infrastructure (e.g., Amazon EC2 and Windows Azure), in order to utilize the powerful computing and processing capabilities by the public clouds. As a matter of fact, the current public cloud architecture - built around static Internet-based installments of cloud resources not integrated with wireless networks - is already starting to show its limits in terms of computation-intensive mobile application support, since mobile users would experience long latency for data exchange with the public cloud through the wide area network (WAN), which risks to become the major impediment in satisfying the real-time interactive response requirement of mobile applications.

To address this challenge, a novel mobile cloud computing paradigm, called mobile-edge cloud computing, has been proposed [4]. As illustrated in Fig. 1, mobile-edge cloud

computing can provide cloud-computing capabilities at the edge of pervasive radio access networks (e.g., 3G, 4G, WiMax, femtocells) in close proximity to mobile users. In this case, the need for fast interactive response can be met by fast and low-latency connection (e.g., via fiber transmission) to resource-rich cloud computing infrastructures (called mobile-edge clouds) deployed by network operators (e.g., AT&T and T-Mobile) within the network edge and backhaul/core networks. By endowing ubiquitous radio access networks with powerful computing capabilities, mobile-edge cloud computing is envisioned to provide pervasive and agile computation augmenting services for mobile users at anytime and anywhere [4].

In this paper, we aim at devising an efficient mobile-edge cloud computing framework that can provide rich flexibility in meeting different mobile users' demands. To this end, in this paper we propose a comprehensive framework consisting of a resource-efficient computation offloading mechanism for the users and a joint communication and computation (JCC) resource allocation mechanism for the network operator. Specifically, we first address the resource-efficient computation offloading problem, in order to reduce a user's resource occupancy. We then study the admission control problem and design the pricing scheme for JCC resource allocation, by jointly taking into account the objective of system-wide performance optimization as well as the practical constraints such as computational complexity for practical implementation and truthfulness for preventing manipulation.

The main results and contributions of this paper are as follows:

• *We address the resource-efficient computation offloading problem for each individual user subject to the QoS constraint:* Specifically, by leveraging the structural property of user's task graph, we propose an efficient delay-aware task graph partition algorithm for computation offloading. Building on this, we derive the optimal communication and computation resource demanding profile for a user that minimizes the resource occupancy and meanwhile satisfies the completion time constraint.

• *We study the admission control problem for selecting the proper set of users in JCC resource demand satisfaction:* We show the admission control problem involving joint communication and computation resource allocation is NP-hard. We hence develop an efficient approximation solution by carefully designing the user ranking criteria, which has a low complexity to facilitate the practical implementation. We further derive the upper bound of the performance loss of the approximate admission control solution with respect to the global optimal solution. Numerical results demonstrate that the proposed approximate solution is very efficient, with at most 15% performance loss.

• *We investigate the truthful pricing problem for preventing manipulation by untruthful valuation reporting:* We borrow the powerful tool of critical value approach in auction theory, and rigorously show that the proposed pricing scheme is truthful such that no user has the incentive to lie about its valuation for acquiring the mobile-edge cloud service to complete its computational task. This would be very useful to prevent manipulations by untruthful valuation report, which would lead to inefficient allocation of resources and system performance degradation.

The rest of the paper is organized as follows. We first introduce the system model in Section II. We then study

resource-efficient computation offloading strategy for the users in Section III. We next discuss the admission control problem and pricing issue for JCC resource allocation in Sections IV and V, respectively. We conduct the performance evaluation in Section VI, discuss the related works in Section VII, and finally conclude in Section VIII.

II. SYSTEM MODEL

We consider that there are a set of K wireless base-stations (e.g., macrocell/femtocell base-stations) $\mathcal{K} = \{1, 2, ..., K\}$ and a set of B mobile-edge clouds $\mathcal{L} = \{1, 2, ..., L\}$ in the system. For a base-station $k \in \mathcal{K}$, there are M_k orthogonal wireless subchannels dedicated for supporting on-demand mobile-edge cloud service, and each subchannel has a bandwidth of w. Moreover, a base-station k is directly connected to the closest mobile-edge cloud $l \in \mathcal{L}$ in proximity, which has a total computation resource capacity of B_l. Note that as illustrated in Fig. 1, our model allows that multiple nearby base-stations share the same mobile-edge cloud in proximity.

A set of N mobile users $\mathcal{N} = \{1, 2, ..., N\}$ would like to acquire the mobile-edge cloud service from the network operator to complete their computation-intensive tasks. In the following we denote the base-station and the mobile-edge cloud with which a user n is associated as $k(n)$ and $l(n)$, respectively. Subject to the QoS requirement of the application, a user $n \in \mathcal{N}$ also has a maximum allowable completion time $\mathcal{T}_n$ for its task, and possesses a valuation v_n, i.e., the maximum amount of monetary cost that the user is willing to pay for acquiring the mobile-edge cloud service to complete its task.

In the following parts, we will first study the resource-efficient computation offloading problem for each user, in order to reduce user's resource occupation by determining its optimal communication and computation resource profile with minimum resource occupation, and meanwhile satisfying the completion time constraint. We will then design the efficient admission control mechanism for the network operator in order to select the proper set of users for JCC resource allocation subject to the resource capacity constraints. We will further develop a pricing scheme to prevent the manipulation such that some users may untruthfully report their valuations in acquiring mobile-edge cloud service, which would lead to inefficient allocation of resources and system performance degradation.

Note that for ease of exposition, in this paper we consider two most critical resource types for mobile applications, i.e., communication and computation resources. Our model can be easily extended to the cases with more resource types such as storage resource. Also, similar to many existing studies in wireless resource auction (e.g., [5]–[7]), as an initial thrust and to enable tractable analysis, in this paper we consider a static setting such that users are stationary. The more general case that users may dynamically depart and leave the mobile-edge cloud system is very challenging and will be addressed in a future work.

III. RESOURCE-EFFICIENT COMPUTATION OFFLOADING

In this section, we consider the resource-efficient computation offloading problem of each individual user to determine the optimal joint communication and computation resource profile
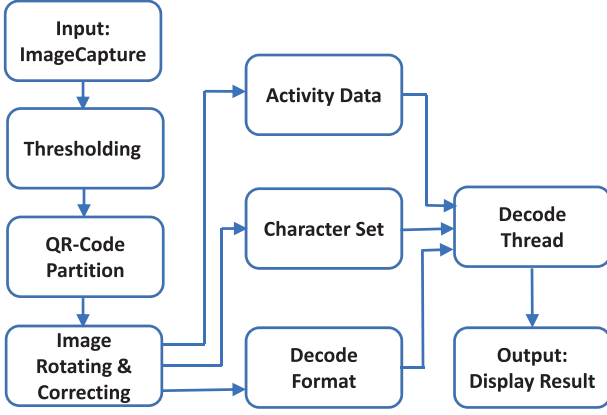
Fig. 2. Task graph of QR-code recognition application [6].

Algorithm 1: Topology Sorting of Task Graph.

set the sorted node list $\widetilde{\mathcal{V}}_n = \emptyset$
set $\mathcal{S}$ as the set of all nodes with no incoming edges
while $\mathcal{S} \neq \emptyset$ **do**
    remove a node $i \in \mathcal{S}$
    add node i to the end of the list $\widetilde{\mathcal{V}}_n$
    for *each node j such that $(i,j) \in \mathcal{E}_n$* **do**
        remove edge (i,j) from the task graph $\mathcal{G}_n(\mathcal{V}_n, \mathcal{E}_n)$ **if**
        node j has no other incoming edges **then**
            add node i to the set $\mathcal{S}$
        end
    end
end
return the list $\widetilde{\mathcal{V}}_n$

---

of a user for the cost-effective auction bidding, in order to minimize the resource occupancy and meanwhile satisfy the required completion time constraint.

A. Task Graph Model

To proceed, we first introduce the task graph model to describe the processing procedure of the computational task of a user n. Similar to many studies (e.g., [2], [6], [8]–[11] and references therein), we describe the computation task profile of a user n by a directed acyclic graph $\mathcal{G}_n(\mathcal{V}_n, \mathcal{E}_n)$, with the end node as the output component to be executed at the user's device. Fig. 2 illustrates an example of the task graph of QR-code recognition application [6]. Specifically, in the task graph $\mathcal{G}_n(\mathcal{V}_n, \mathcal{E}_n)$, the nodes represent the task components and the directed edges stand for the data dependencies. For a node $i \in \mathcal{V}_n$ in the task graph $\mathcal{G}_n(\mathcal{V}_n, \mathcal{E}_n)$, we denote C_i as the total computational instructions (i.e., CPU cycles) to execute the corresponding task component. For a subsequential node $j \in \mathcal{V}_n$ that invokes the output data from node i (i.e., $(i,j) \in \mathcal{E}_n$), we denote D_{ij} as the amount of data that input from node i to node j. Note that in practice many applications can be presented by such a directed acyclic task graph model[1] [2], [6], [8]–[11], e.g., face recognition, natural language processing, video streaming, virtual argumentation, data processing applications such as MapReduce.

B. Computation and Communication Models

We then introduce both the communication and computation models. For a given task component i of user n on its task graph (i.e., $i \in \mathcal{V}_n$), the user can execute it either locally at its own device or remotely at the mobile-edge cloud. For the local computing approach, we have the time for executing the given task component i as $T_{i,n}^m = \frac{C_i}{F_n^0}$, where F_n^0 denotes the computation capability (CPU cycles per second) of user n's device. For the cloud computing approach, a virtual machine is associated to user n for its task execution in the mobile-edge cloud. To provide flexibility in the mobile-edge cloud service provision, we consider that there is a set of S different virtual machine types $\mathcal{S} = \{1, 2, ..., S\}$ available and the computation

[1]We will consider the cyclic task graph case in a future work. One possible solution is to transform this case to the acyclic case by grouping the nodes in a cycle as a "supper" node.

capability of a virtual machine type $s \in \mathcal{S}$ is denoted by F_s^c. Let s_n be the virtual machine type chosen by user n for executing its whole task. Accordingly, we can then compute the time by the cloud computing approach for executing the given task component i on the virtual machine as $T_{i,n}^c = \frac{C_i}{F_{s_n}^c}$.

Now suppose that two task components $i, j \in \mathcal{V}_n$ with $(i,j) \in \mathcal{E}_n$ are executed at different locations (one at user's device and the other at the mobile-edge cloud). Then a total amount D_{ij} of data needs to be transferred between these two components via wireless connection. We hence introduce the communication model for wireless data transmission. Let $q_n \leq M_{k(n)}$ denote the number of subchannels demanded by user n. We can then compute the data rate for computation offloading of user n as $R_n = q_n w \log_2(1 + \frac{\eta_n h_{n,k(n)}}{\varpi_n})$, where η_n is the transmission power of user n which is determined by the base station according to some power control algorithm such as [10], [12], $h_{n,k(n)}$ is the channel gain between the user n and the base station, and ϖ_n is the background noise. Based on the communication model above, the total data transmission time for computation offloading between task components i and j is given as $T_{ij,n}^{tx} = \frac{D_{ij}}{R_n}$.

C. Delay-Aware Task Graph Partition

We next consider the delay-aware task graph partition problem for a fixed resource allocation. That is, given that a user n's subchannel demand q_n for wireless communication and the virtual machine type selection s_n for cloud computation are fixed, we would like to determine the optimal execution location (i.e., device or cloud) for each task component in its task graph, in order to minimize the total delay of executing the whole task.

To address the delay-aware task graph partition problem, we leverage the structural property of the task graph $\mathcal{G}_n(\mathcal{V}_n, \mathcal{E}_n)$. Since the graph $\mathcal{G}_n(\mathcal{V}_n, \mathcal{E}_n)$ is directed acyclic, it is known that we can carry out a topology sorting of the graph, i.e., ordering of the nodes such that for every directed edge $(i,j) \in \mathcal{E}_n$ from node i to node j, i comes before j in the ordering [13].

We introduce the classic Kahn Algorithm [13] for the topology sorting of the task graph in Algorithm 1, which has a computational complexity of $\mathcal{O}(|\mathcal{V}_n| + |\mathcal{E}_n|)$ (please refer to [13] for the details). After the topology sorting, we then obtain an ordering of the nodes as $\widetilde{\mathcal{V}}_n = (a_1, a_2, ..., a_{|\mathcal{V}_n|})$ such that $a_1 <$

$a_2 < \cdots < a_{|\mathcal{V}_n|}$. We also denote the set of nodes to which there is an edge directed from node i as $\Delta_n(i) = \{j : (i,j) \in \mathcal{E}_n\}$. According to the definition of topology sorting, we know that if $j \in \Delta_n(i)$, then $i < j$ in the ordering $\widetilde{\mathcal{V}_n}$. Note that since no edge is directed from the end node (i.e., the output component) and for any other node there always exists some edge directed from it (otherwise the corresponding task component is useless since it does not provide any input to either other intermediate components or the output component), the last node $a_{|\mathcal{V}_n|}$ in the ordering $\widetilde{\mathcal{V}_n}$ must be the output component.

Based on the node ordering by topology sorting, we then solve the delay-aware task graph partition problem by the principle of backward induction [13]. That is, we first determine the optimal execution location for the last node in the ordering $\widetilde{\mathcal{V}_n}$, and based on which we can determine the optimal location for the nodes by moving backward. Specifically, we denote the execution location of the task component i as $y_{i,n} \in \{m,c\}$, with $y_{i,n} = c$ denoting the component will be executed at the cloud and $y_{i,n} = m$ denoting the component will be executed at the device. Moreover, we denote the minimum delay starting from executing the task component i till the end of the whole task as $Z_n(i)$.

For the last node $a_{|\mathcal{V}_n|}$ (i.e., the output component) in the ordering $\widetilde{\mathcal{V}_n}$, we have $Z_n(a_{|\mathcal{V}_n|}) = T^m_{a_{|\mathcal{V}_n|},n}$ and the optimal execution location is $y^*_{a_{|\mathcal{V}_n|},n} = m$, due to the fact that the final result will be returned to the user's device. Then we move backward to consider the remaining nodes $a_i < a_{|\mathcal{V}_n|}$ in the ordering $\widetilde{\mathcal{V}_n}$ in a recursive manner. According to the definition of $Z_n(a_i)$, we know that the delay starting from executing the task component a_i till the end of the whole task depends on that of the bottleneck node $j \in \Delta_n(a_i)$ with maximum delay that requires the input data from node i. Thus, we have the following:

$$Z_n(a_i) = \min_{y \in \{m,c\}} \max_{j \in \Delta_n(a_i)} \{T^y_{a_i,n} + T^{tx}_{a_i j,n} \mathbf{1}_{\{y^*_{j,n} \neq y\}} + Z_n(j)\},$$
(1)

and

$$y^*_{a_i,n} = \arg\min_{y \in \{m,c\}} \max_{j \in \Delta_n(a_i)} \{T^y_{a_i,n} + T^{tx}_{a_i j,n} \mathbf{1}_{\{y^*_{j,n} \neq y\}} + Z_n(j)\},$$
(2)

where $\mathbf{1}_{\{A\}}$ is an indicator function such that $\mathbf{1}_{\{A\}} = 1$ if the event A is true and $\mathbf{1}_{\{A\}} = 0$ otherwise. Here $\mathbf{1}_{\{y^*_{j,n} \neq y\}} = 1$ implies the wireless data transmission is required since components i and j are executed at different locations.

By the back induction, for all the remaining nodes $a_i < a_{|\mathcal{V}_n|}$, we can recursively compute the values of $Z_n(a_i)$ and the optimal execution locations $y^*_{a_i,n}$ according to (1) and (2), respectively. We describe the procedure for the delay-aware task graph partition in Algorithm 2. The complexity of the backward induction is $\mathcal{O}(|\mathcal{V}_n|)$ and the complexity of topology sorting is $\mathcal{O}(|\mathcal{V}_n| + |\mathcal{E}_n|)$. We hence know then complexity of the delay-aware task graph partition algorithm is at most $\mathcal{O}(|\mathcal{V}_n|^2)$ since $|\mathcal{E}_n| \leq |\mathcal{V}_n|^2$.

D. Resource-Efficient Demanding Strategy

In Section III-C, we have solved the delay-aware task graph partition problem given the fixed subchannel demand q_n and

---

Algorithm 2: Delay-Aware Task Graph Partition.

conduct the topology sorting of $\mathcal{G}_n(\mathcal{V}_n, \mathcal{E}_n)$ and obtain the sorted node list $\widetilde{\mathcal{V}_n}$
set $Z_n(a_{|\mathcal{V}_n|}) = T^m_{a_{|\mathcal{V}_n|},n}$ and $y^*_{a_{|\mathcal{V}_n|},n} = m$ for the end node $a_{|\mathcal{V}_n|}$
for $i = |\mathcal{V}_n| - 1, ..., 1$ **do**
$\quad$ **compute** $Z_n(a_i)$ and $y^*_{a_i,n}$ according to (1) and (2), respectively
end
return $\{Z_n(a_i)\}_{a_i \in \mathcal{V}_n}$ and $\{y^*_{a_i,n}\}_{a_i \in \mathcal{V}_n}$

---

virtual machine type selection s_n of a user n. Building on this, we then determine the resource-efficient demanding strategy (i.e., the optimal subchannel demand q_n and virtual machine type selection s_n) of user n to minimize the resource occupancy, while guaranteeing its task can be completed within the completion time deadline $\mathcal{T}_n$.

First of all, we define the resource occupancy function for a user n as the sum of the normalized occupancy ratios of the user for both communication and computation resources, i.e.,

$$\Phi_n(q_n, s_n) = \frac{q_n}{M_{k(n)}} + \frac{F^c_{s_n}}{B_{l(n)}}.$$
(3)

Recall that $M_{k(n)}$ and $B_{l(n)}$ are the total number of available subchannels and computation capacity at user n's associated base-station and mobile-edge cloud, respectively. The key motivation of minimizing the resource occupancy is two-fold. On one hand, in the operator's admission control mechanism design for later, the resource occupancy serves as an important indicator for accepting a user's request and determining its payment. Thus, from the user's point of view, a smaller resource occupancy implies a higher chance of getting admitted into the mobile-edge cloud service and a lower service payment as well. On the other hand, from the perspective of the network operator, it helps to increase the resource utilization efficiency and support more users in the mobile-edge cloud service.

Then, the resource-efficient demanding strategy can be obtained by solving the following optimization problem:

$$\min_{q_n, s_n} \quad \Phi_n(q_n, s_n)$$
(4)

$$\text{s.t.} \quad Z_n(i) \leq \mathcal{T}_n, \forall i \in \mathcal{V}_n.$$
(5)

The objective is to minimize the resource occupancy, and the constraint guarantees that the whole task (i.e., all the task components) will be completed within the deadline $\mathcal{T}_n$. Building upon the delay-aware task graph partition algorithm in Algorithm 2, this problem can be solved as follows: first, we rank all the communication and computation resource profiles (q_n, s_n) of a user according to the corresponding resource occupancy function value $\Phi_n(q_n, s_n)$ (with ties randomly broken). Then, according to the ranking, we sequentially select a resource profile (q_n, s_n) and utilize Algorithm 2 to compute the value of $Z_n(i)$ for any $i \in \mathcal{V}_n$. Once the constraint in (5) is satisfied, then we stop and find the optimal resource profile (q^*_n, s^*_n) having the minimum value of $\Phi_n(q_n, s_n)$. The key idea is that, by the backward induction based on (1) and (2), Algorithm 2 finds

the optimal policy of the task graph partition that minimizes the total executing delay for given a fixed communication and computation resource allocation profile (q_n, s_n). So, the optimal resource allocation profile for the delay constrained optimization problem in (4) can be obtained by exhaustive searching over all the feasible resource allocation profiles. To do it in an efficient way, here we first rank the objective function value $\Phi_n(q_n, s_n)$ (3) for all the feasible resource allocation profiles in an increasing manner, and then sequentially search for the optimal resource profile with the minimum value $\Phi_n(q_n, s_n)$ and meanwhile the delay constraint in (5) being satisfied.

Note that the resource-efficient demanding strategy can be locally computed by each user based on its individual application information. Since there are at most $M_{k(n)}$ subchannels and S different virtual machine types, in the worst case a user n only has to check at most $M_{k(n)}S$ resource profiles, then the total complexity of computing the resource-efficient demanding strategy in (4) is at most $\mathcal{O}(M_{k(n)}S|\mathcal{V}_n|^2)$, which is computationally feasible.

IV. ADMISSION CONTROL FOR JCC RESOURCE ALLOCATION

After receiving users' resource demanding profiles, a key challenge would be that the total available communication and computation resources are limited in practice and the network operator may not satisfy all resource demands of the users. We hence have to design a user admission control mechanism for the network operator to select the proper set of requesting users to serve, in order to optimize the system-wide performance subject to both communication and computation capacity constraints.

A. Problem Formulation

We first formally state the admission control problem for JCC resource allocation as follows.

Definition 1: (Admission Control Problem) Given the resource demanding profiles of all the users, the operator would like to solve the following admission control problem:

$$\max_{\mathcal{C} \subseteq \mathcal{N}} \quad \sum_{n \in \mathcal{C}} v_n \tag{6}$$

$$\text{s.t.} \sum_{n : n \in \mathcal{C}, k(n)=i} q_n^* \leq M_i, \forall i \in \mathcal{K}, \tag{7}$$

$$\sum_{n : n \in \mathcal{C}, l(n)=j} F_{s_n^*}^c \leq B_j, \forall j \in \mathcal{L}. \tag{8}$$

The objective in (6) of the admission control problem is to select the proper set of users $\mathcal{C}$ to serve, in order to maximize the social welfare (i.e., system-wide performance) which is the sum of the accepted users' valuations. In other words, we would like to efficiently allocate the resources to the set of users that value those resources most. The first constraint in (7) means that for a base-station i, the total requested communication resources (i.e., subchannels) by the selected users should not exceed its capacity. The second constraint in (8) represents that for a mobile-edge cloud j, the total requested computation resources by the selected users should not exceed its total available computation

capacity. In this paper we aim at designing efficient resource allocation scheme for optimizing the system-wide performance (i.e., social welfare maximization), and the case of maximizing the revenue of the operator will be considered in a future work. Note that since the valuation of a user characterizes its willingness to pay, maximizing the sum of the valuations would also have a positive effect on increasing the revenue obtained by the operator.

Definition 2: (Computational Efficiency) A mechanism is computationally efficient if the computing procedure of the mechanism terminates in polynomial time.

The computational efficiency of a devised algorithm for JCC resource allocation is very important in practice. Any optimal algorithm with high complexity would require a large amount of time overhead to compute the optimal solution, which leads to high latency and slow responsiveness and hence is useless in reality.

B. Computational Complexity of Admission Control Problem

We first consider the computational complexity of the admission control problem. As mentioned above, to make the JCC resource allocation useful in practice, we need to develop a computationally efficient algorithm for the admission control problem. Unfortunately, we show that it is NP-hard to compute an optimal solution for the admission control problem.

Theorem 1: The admission control problem for the JCC resource allocation is NP-hard.

Proof: We consider the special case with only one base station having M subchannels available and one mobile-edge cloud having B computation capacity. In this case, the admission control problem becomes

$$\max_{\mathcal{C} \subseteq \mathcal{N}} \quad \sum_{n \in \mathcal{C}} v_n \tag{9}$$

$$\text{s.t.} \sum_{n : n \in \mathcal{C}} q_n^* \leq M, \tag{10}$$

$$\sum_{n : n \in \mathcal{C}} F_{s_n^*}^c \leq B. \tag{11}$$

Actually, the problem above is a two-dimensional knapsack problem, by regarding the communication and computation resources as two deimensions. It is known that the two-dimensional knapsack problem is strongly NP-hard [14]. Thus, as a generalized extension of the two-dimensional knapsack problem, the admission control problem in our case is also NP-hard. ∎

The NP-hardness of the admission control problem implies that we have to develop approximation algorithms for designing a fast admission control mechanism.

C. Approximate Admission Control Algorithm

We then propose an approximate approach to solve the admission control problem. The key idea is to first design the proper ranking metric to order the demanding profiles of all the users, and then select the set of accepted users in a sequential manner.

Algorithm 3: Admission Control for JCC Resource Allocation.

initialize $x_1 = ... = x_N = 0$
for $n \in \mathcal{N}$ **do**
    compute $\gamma_n = \frac{\lambda_n}{\Phi_n(q_n^*, s_n^*)}$
end
sort all the users in $\mathcal{N}$ in the decreasing order of γ_n
for $n = 1, 2, ..., N$ *in the sorted set* $\mathcal{N}$ **do**
    if $\sum_{i<n:k(i)=k(n)} q_i^* x_i + q_n^* \le M_{k(n)}$ *and*
    $\sum_{i<n:k(i)=k(n)} F_{s_i^*}^c x_i + F_{s_n^*}^c \le B_{l(n)}$ **then**
       set $x_n = 1$
    end
end
return $\{x_n\}_{n \in \mathcal{N}}$

Specifically, given the set of demanding profiles by all users $\{\varphi_n = <(q_n^*, s_n^*), v_n>\}_{n \in \mathcal{N}}$, the operator first orders the bids according to the following performance metric (with tie randomly broken):

$$\gamma_n = \frac{v_n}{\Phi_n(q_n^*, s_n^*)}. \qquad (12)$$

Recall that v_n is user's valuation and $\Phi_n(q_n^*, s_n^*)$ is the resource occupancy function in (3). The physical meaning of the performance metric γ_n indicates the user's valuation per unit resource occupancy. Intuitively, if a user has a higher valuation and consumes fewer resources, then the user should have a higher chance to get accepted in the JCC resource allocation.

After ranking all users, the operator then sequentially selects the set of accepted users one by one. For a candidate user n, the operator checks, by selecting this user as an accepted user, whether the total requested communication and computation resources exceed the capacity of its associated base-station and the mobile-edge cloud. If the capacity constraint is not violated, then the operator will accept this user as an accepted user; otherwise, the operator will reject this user. We summarize the whole procedure for determining the accepted users in Algorithm 3.

D. Performance Analysis

For the approximate admission control in Algorithm 3, it has the complexity of $\mathcal{O}(N)$ to compute the ranking metrics of all users, $\mathcal{O}(N \log N)$ to sort the users, $\mathcal{O}(N)$ to select the accepted users. Thus the complexity of Algorithm 3 is $\mathcal{O}(N \log N)$. Hence, the approximate admission control is computationally efficient for JCC resource allocation.

We next analyze the worst-case performance of the proposed admission control algorithm, with respect to the optimal solution that maximizes the social welfare in (6). Let $\mathcal{C}$ and $\mathcal{C}^*$ denote the set of accepted users by the admission control algorithm in Algorithm 3 and the optimal solution in (6), respectively. We also denote the social welfare by the admission control algorithm and the optimal solution as $W = \sum_{n \in \mathcal{C}} v_n$ and $W^* = \sum_{n \in \mathcal{C}^*} v_n$, respectively. We define the approximation ratio $0 \le \rho \le 1$ of the admission control algorithm such that $W \ge \rho W^*$.

We first consider the special case that the total communication resource reserved for mobile-edge cloud service at each base station is homogeneous and each mobile user demands the same communication resource amount, i.e., $M_k = M$ for any $k \in \mathcal{K}$ and $q_n^* = q$ for any $n \in \mathcal{N}$. This could correspond to the situation that all the base-stations reserve the same communication resource for mobile-edge could computing service and each user is allowed to utilize one channel for computation offloading in practice. For this special case, we have the following result.

Theorem 2: For the case that $M_k = M$ for any base-station $k \in \mathcal{K}$ and $q_n^* = q$ for any user $n \in \mathcal{N}$, the admission control algorithm in JCC resource allocation can approximate the optimal solution with a ratio of $\rho = \frac{1}{2}$.

Proof: For this special case, the admission control problem can be reformulated as

$$\max_{x_n} \sum_{n=1}^{N} v_n x_n$$

$$\text{s.t.} \sum_{n:k(n)=i} x_n \le \frac{M}{q}, \forall i \in \mathcal{K},$$

$$\sum_{n:l(n)=j} F_{s_n^*}^c \le B_j, \forall j \in \mathcal{L},$$

$$x_n \in \{0, 1\}, \forall n \in \mathcal{N}.$$

Since a base-station is connected to only one mobile-edge cloud, in this case we can decompose the problem above by only considering the sub-problem for each mobile-edge cloud as follows. Here we denote the set of base-stations and users connecting to base-station j as $\mathcal{K}_j$ and $\mathcal{N}_j$, respectively.

$$\max_{x_n} \sum_{n \in \mathcal{N}_j} v_n x_n$$

$$\text{s.t.} \sum_{n:k(n)=i} x_n \le \frac{M}{q}, \forall i \in \mathcal{K}_j,$$

$$\sum_{n:l(n)=j} F_{s_n^*}^c \le B_j,$$

$$x_n \in \{0, 1\}, \forall n \in \mathcal{N}_j. \qquad (13)$$

Now suppose we relax the problem above by allowing the variable x_n taking fractional value, i.e., $x_n \in [0, 1]$ as follows.

$$\max_{x_n} \sum_{n \in \mathcal{N}_j} v_n x_n \qquad (14)$$

$$\text{s.t.} \sum_{n:k(n)=i} x_n \le \frac{M}{q}, \forall i \in \mathcal{K}_j, \qquad (15)$$

$$\sum_{n:l(n)=j} F_{s_n^*}^c x_n \le B_j,$$

$$0 \le x_n \le 1, \forall n \in \mathcal{N}_j. \qquad (16)$$

For the relaxed problem above, we can easily solve it by the greedy manner using $r_n = \frac{v_n}{F_{s_n^*}^c}$ as the ranking metric. Specifically, we first rank all the users in the set $\mathcal{N}_j$ accordingly to the ranking metric $r_n = \frac{v_n}{F_{s_n^*}^c}$ and then sequentially add a user into the selected set (i.e., $x_n = 1$) if both conditions (15) and (16) are

satisfied. When a user violates condition (15), we set $x_n = 0$. When a user who first violates (16), we set $x_n = \alpha \in [0, 1)$ as a fractional value to ensures the condition (15) is satisfied and the algorithm ends. We can prove that this greedy algorithm can find the optimal solution to problem (14) by contradiction. This is because, if there exists a user n of a higher ranking metric r_n that is selected by the greedy algorithm but not included in the final solution, we can swap it with another user m of lower ranking metric r_m that is not picked by the greedy algorithm but is included in the solution to improve the solution. In particular, if there exists such a user m that belongs to the same base-station with user n, then we swap user n with user m in the solution and the conditions (15) and (16) are still satisfied. If not, we we swap user n with user m from another base-station. Since in the greedy algorithm user n is selected and the conditions (15) and (16) are satisfied, then after the swap the the conditions are still satisfied. As a result, we see that the solution can be improved and this contradicts with the optimal assumption.

Without loss generality, we denote the optimal solution for problem (14) as $x_1 = \cdots = x_d = 1$, $x_{d+1} = \alpha$ and $x_n = 0$ for any $n > d + 1$. Since $q_n^* = q$ and $M_k = M$, then the ranking metric in the admission control algorithm becomes $r_n = \frac{v_n}{q/M + F_{s_n^*}^c / B_j}$. This is equivalent to using the ranking metric $r_n = \frac{v_n}{F_{s_n^*}^c}$ in the admission control algorithm. As a result, the admission control algorithm can obtain the solution as $x_1 = \cdots = x_d = 1$ and $x_n = 0$ for any $n > d$. Let W_F and W_G denotes the values by the two solutions $(1, ..., 1, \alpha, 0, ..., 0)$ and $(1, ..., 1, 0, 0, ..., 0)$ above, and W_{OPT} denotes the optimal solution to the original problem (13). Then, we have

$$W_{OPT} \le W_F \le W_G + \alpha v_{d+1} \le 2W_G.$$

Thus, we have proved that $W_G \ge \frac{1}{2}W_{OPT}$. ∎

The key idea of the proof for Theorem (2) above is that we first formulate the special case admission control problem as an integer programming problem, and then show that the proposed approximate admission control algorithm can obtain the optimal solution for its relaxed fractional programming problem. By exploring the connection between the original integer programming problem and the relaxed fractional programming problem, we can finally show that in the worst case the approximate admission control algorithm can achieve approximation ratio of 1/2 for the original problem.

For the general case that each base station may have heterogenous total communication resource and each mobile user may demand the different amount of communication resource, we can show the following result.

Theorem 3: For the general case, the admission control algorithm in JCC resource allocation can approximate the optimal solution with a ratio of $\rho = \frac{A_{\max} + B_{\max}}{2A_{\max}B_{\max}}$, where $A_{\max} = \max_{k \in \mathcal{K}}\{M_k\}$ and $B_{\max} = \max_{l \in \mathcal{L}, s \in \mathcal{S}}\{\frac{B_l}{F_s^c}\}$.

Proof: Based on the ranking metric γ_n, we first define the set of users Ω_n of a user n such that if $i \in \Omega_n$ then $i \ge n$ and $i \in \mathcal{C}^*$ but $i \notin \mathcal{C}$ due to that user n blocks user i being accepted in the JCC resource allocation. In the following we also define that $\widetilde{\Omega}_n = \Omega_n \cup \{n\}$. According to the admission control algorithm, we know that for any user $i \in \widetilde{\Omega}_n$, we have

$\gamma_i \le \gamma_n$, $k(n) = k(i)$ and $l(n) = l(i)$, which implies that

$$v_i \le v_n \frac{\Phi_i(q_i^*, s_i^*)}{\Phi_n(q_n^*, s_n^*)} = v_n \left(\frac{q_i^*}{M_{k(n)}} + \frac{F_{s_i^*}^c}{B_{l(n)}} \right) /$$
$$\left(\frac{q_n^*}{M_{k(n)}} + \frac{F_{s_n^*}^c}{B_{l(n)}} \right).$$

By summing over all the users $i \in \widetilde{\Omega}_n$, we have

$$\sum_{i \in \widetilde{\Omega}_n} v_i = v_n / \left(\frac{q_n^*}{M_{k(n)}} + \frac{F_{s_n^*}^c}{B_{l(n)}} \right) \sum_{i \in \widetilde{\Omega}_n} \left(\frac{q_i^*}{M_{k(n)}} + \frac{F_{s_i^*}^c}{B_{l(n)}} \right).$$

Since

$$\sum_{i \in \widetilde{\Omega}_n} \left(\frac{q_i^*}{M_{k(n)}} + \frac{F_{s_i^*}^c}{B_{l(n)}} \right) \le 2 \text{ and } \frac{q_n^*}{M_{k(n)}} + \frac{F_{s_n^*}^c}{B_{l(n)}}$$
$$\ge \frac{1}{A_{\max}} + \frac{1}{B_{\max}},$$

we then have

$$v_n \ge \frac{A_{\max} + B_{\max}}{2A_{\max}B_{\max}} \sum_{i \in \widetilde{\Omega}_n} v_i.$$

Furthermore, we know that $\mathcal{C}^* \subseteq \bigcup_{n \in \mathcal{C}} \widetilde{\Omega}_n$. This implies that

$$\sum_{n \in \mathcal{C}} v_n \ge \frac{A_{\max} + B_{\max}}{2A_{\max}B_{\max}} \sum_{n \in \mathcal{C}} \sum_{i \in \widetilde{\Omega}_n} v_i$$
$$\ge \frac{A_{\max} + B_{\max}}{2A_{\max}B_{\max}} \sum_{i \in \bigcup_{n \in \mathcal{C}} \widetilde{\Omega}_n} v_i$$
$$\ge \frac{A_{\max} + B_{\max}}{2A_{\max}B_{\max}} \sum_{i \in \mathcal{C}^*} v_i.$$

Thus, we have $W \ge \frac{A_{\max} + B_{\max}}{2A_{\max}B_{\max}} W^*$, which completes the proof. ∎

Theorem 3 describes the performance guarantee of the admission control algorithm in the worst case (which rarely happens in practice). Numerical results demonstrate that in practice the proposed algorithm is very efficient, with at most 14.3% performance loss, compared to the optimal solution.

V. TRUTHFUL PRICING FOR JCC RESOURCE ALLOCATION

As discussed above, the valuation v_n of a user n measures the importance or the utility that the user n can achieve for completing its task within the delay constraint. It plays a critical role in the admission control for JCC resource allocation. As a result, it would risk that some users may untruthfully report their valuations v_n in order to increasing their chances in getting accepted in JCC resource allocation. These manipulations by the users would lead to inefficient allocation of resources and system-wide performance degradation. Thus, it is highly desirable to prevent such manipulations by designing truthful pricing scheme such that all users have incentives to report their true valuations.

Formally, if a user n pays a monetary cost p_n for its demanded resources, the user receives the net payoff of $v_n - p_n$. Then the

truthful pricing problem for JCC resource allocation is defined as follows.

Definition 3: (Truthful Pricing Problem) For a user $n \in \mathcal{N}$, let $\varphi_n = <(q_n^*, s_n^*), v_n>$ and $\widetilde{\varphi}_n = <(q_n^*, s_n^*), \widetilde{v}_n>$ denote the demand report profiles with the truthful valuation v_n and untruthful valuation $\widetilde{v}_n$, respectively. We also denote the payments of user i under the truthful demand report φ_n and the untruthful report $\widetilde{\varphi}_n$ as $p_n(\varphi_n)$ and $p_n(\widetilde{\varphi}_n)$, respectively. Then net payoffs of user n for the truthful report and the untruthful report are $v_n - p_n(\varphi_n)$ and $v_n - p_n(\widetilde{\varphi}_n)$, respectively. The truthful pricing problem is to design a payment scheme such that $v_n - p_n(\varphi_n) \geq v_n - p_n(\widetilde{\varphi}_n)$, i.e., no user can benefit by reporting its valuation untruthfully.

A. Truthful Pricing Scheme

We next consider the truthful pricing scheme design. Here we borrow the critical value approach in auction theory [15] for our design. Let λ_n denote user n's reported valuation. Intuitively, a critical value θ_n of a user n is the minimum value that user n should pay in order to be accepted in admission control for JCC resource allocation. That is, if $\lambda_n > \theta_n$, then user n is accepted; if $\lambda_n < \theta_n$, then user n is rejected. And accordingly the truthful pricing scheme based the critical value is that if a user n is accepted, then user pay θ_n; otherwise, it pays 0.

The key challenge of the critical value approach is determining the critical value. For our problem, we show that determining the critical value θ_n of a user n is equivalent to identifying its critical user, whose demanding profile is the first profile following user n that has been rejected in admission control but would have been granted if user n is absent in the JCC resource allocation. Note that if a user does not have a critical user, it has a critical value of zero.

To proceed, in order to find the critical user of an accepting user n, we first assume user n's demanding profile is not in the ranked user list by the admission control algorithm. Then we identify the first user i (following user n in the ranked user list) in the remaining set that if that user is selected as an accepted user (i.e., satisfy the resource constraints) it would make some resource constraint violated when we also add user n in the accepted list. Thus, this user i is indeed the critical user of user n, since user i will not be accepted in admission control when user n has already been accepted. Accordingly, we can then compute the critical value of the user n as

$$\theta_n = \frac{\lambda_i \Phi_n(q_n^*, s_n^*)}{\Phi_i(q_i^*, s_i^*)}. \tag{17}$$

The procedure for obtaining the truthful pricing of JCC resource auction based on critical value is summarized in Algorithm 4. In the following we will show that the proposed payment scheme will guarantee that the proposed pricing scheme is indeed truthful.

B. Truthfulness Analysis

We first show that the proposed pricing scheme for JCC resource allocation is truthful. According to the critical value approach for auction mechanism design [15], a pricing scheme is truthful if the following conditions hold: (1) the winner

---

Algorithm 4: Pricing Scheme for JCC Resource Allocation.

initialize $p_1 = ... = p_N = 0$
for $n = 1, 2, ..., N$, *sorted in the decreasing order of* γ_n **do**
  if $x_n = 1$ **then**
    compute $\beta_1 = \sum_{j<n:k(j)=k(n)} q_j^* x_j$ and $\beta_2 = \sum_{j<n:l(j)=l(n)} F_{s_j^*}^c x_j$
    for $i = n + 1, ..., N$ **do**
      if $k(i) = k(n)$ **then**
        if $\beta_1 + q_i^* \leq M_{k(n)}$ *and* $\beta_2 + F_{s_n^*}^c \leq B_{l(n)}$ **then**
          compute $\beta_1 = \beta_1 + q_i^*$ and $\beta_2 = \beta_2 + F_{s_i^*}^c$
          if *either* $\beta_1 + q_n^* > M_{k(n)}$ *or* $\beta_2 + F_{s_n^*}^c > B_{l(n)}$ **then**
            set $p_n = \theta_n = \frac{\lambda_i \Phi_n(q_n^*, s_n^*)}{\Phi_i(q_i^*, s_i^*)}$
            break
          end
        end
      end
    end
  end
end
return $\{p_n\}_{n \in \mathcal{N}}$

---

determination algorithm (i.e., the admission control algorithm for our problem) is monotone; (2) each winning user (i.e., each accepted user for our problem) is paid its critical value. To proceed, we formally define these two conditions.

Definition 4: (Monotonicity) An admission control algorithm is monotone, if for a user n when a report $\varphi_n = <(q_n^*, s_n^*), \lambda_n>$ is accepted, then given the reports of the other users φ_{-n} are fixed, the report $\widetilde{\varphi}_n = <(\widetilde{q}_n, \widetilde{s}_n), \widetilde{\lambda}_n>$ satisfying $\widetilde{q}_n \geq q_n^*$, $\widetilde{s}_n \geq s_n^*$ and $\widetilde{\lambda}_n \geq \lambda_n$ by user n would also be accepted.

Intuitively, an admission control algorithm is monotone if, increasing the valuation and/or decreasing the size of the desired communication and computation resources in the bid will not cause an accepted user to lose given the reports of the other users are fixed.

Definition 5: (Critical Value) For a user n in the accepted user set, given the reports of the other users φ_{-n} are fixed, there is a critical value θ_n such that if the user n declares a valuation $\lambda_n > \theta_n$ in its report, then it must be accepted; if the user n declares a valuation $\lambda_n < \theta_n$, then it will be rejected.

We then prove that the proposed pricing scheme is truthful by showing that it satisfies the two conditions above.

Theorem 4: The proposed pricing scheme in Algorithm 4 for JCC resource allocation is truthful.

Proof: We first check the condition of monotonicity. Recall that we use the performance metric $\gamma_n = \frac{\lambda_n}{\Phi_n(q_n^*, s_n^*)}$ to rank the users in the admission control algorithm. Assume that a user n with a report $\varphi_n = <(q_n^*, s_n^*), \lambda_n>$ is an accepted user. Then given the reports of the other users φ_{-n} are fixed, for the user n with the report $\widetilde{\varphi}_n = <(\widetilde{q}_n, \widetilde{s}_n), \widetilde{\lambda}_n>$ satisfying $\widetilde{q}_n \geq q_n^*$, $\widetilde{s}_n \geq s_n^*$ and $\widetilde{\lambda}_n \geq \lambda_n$, we know that $\widetilde{\gamma}_n = \frac{\widetilde{\lambda}_n}{\Phi_n(\widetilde{q}_n, \widetilde{s}_n)} \geq \gamma_n = \frac{\lambda_n}{\Phi_n(q_n^*, s_n^*)}$. Then the report $\widetilde{\varphi}_n$ must also be an accepted report according to the admission control algorithm. Thus, the condition of monotonicity is satisfied.

We then check the condition of critical value. Recall that the critical value of an accepted user n is given as $\theta_n = \frac{\lambda_i \Phi_n(q_n^*, s_n^*)}{\Phi_i(q_i^*, s_i^*)}$ where user i is the critical user of user n, whose report is the first report following user n that has been rejected but would have been granted if user n is absent in accepted user set. Thus, if user n declares a valuation $\lambda_n > \theta_n$ in its report, then user n has a higher priority than the critical user i in the admission control since $\gamma_n = \frac{\lambda_n}{\Phi_n(q_n^*, s_n^*)} > \gamma_i = \frac{\lambda_i}{\Phi_i(q_i^*, s_i^*)}$. In this case the set of accepted users does not change and user n must still be accepted. If user declares a valuation $\lambda_n < \theta_n$ in its report, then we have $\gamma_n < \gamma_i$, i.e., user i has a lower priority than the critical user i in the admission control. In this case, according to Algorithm 4 for determining the critical user, the user i will be accepted, and if we further add user n in the accepted user set the resource capacity constraint is violated. As a result, user n can not be accepted. Thus, the condition of critical value also holds. Combining the arguments above, we hence know that the proposed pricing scheme for JCC resource allocation is truthful. ∎

Hence, by adopting the truthful pricing scheme in Algorithm 4, all users have incentives to report their true valuations to the network operator. This will be very useful to prevent manipulations by untruthful reports and guarantee the system efficiency for mobile-edge cloud computing.

VI. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed on-demand JCC resource allocation mechanisms through numerical studies. We consider that there are $N = 400$ randomly-scattered mobile users requesting mobile-edge cloud service. There are multiple wireless base-stations, each of which has $M_k = 15$ subchannels (each with a bandwidth $w = 1$ Mhz) for supporting on-demand mobile-edge cloud service. The transmission power of a mobile user is $\eta_n = 100$ mW and the background noise $\varpi_n = -100$ dBm [12]. According to the physical interference model, we set the channel gain $h_{n,k(n)} = d_{n,k(n)}^{-\alpha}$ where $d_{n,k(n)}$ is the distance between user n and its associated base-station $k(n)$ and $\alpha = 4$ is the path loss factor [12], [16].

For the computation task, we consider two types of applications – face recognition [2] and high-resolution QR-code recognition [6], and use the task graphs therein. The task completion deadline of each user is randomly determined from the set $\{300, 500, 1000, 2000, 5000\}$ ms, and the computation capability of a user is randomly assigned from the set $\{0.5, 0.8, 1.0\}$ GHz [5]. For the cloud computing, there are multiple mobile-edge clouds, and each of which offers three types of virtual machines with the computation capabilities of $5, 10, 20$ GHz, respectively [17], [18]. The total computation capability of each mobile-edge cloud server is randomly assigned from the set $\{50, 100, 200\}$ GHz [5]. To reduce the connection latency, we assume that each base-station is connected to the closest mobile-edge cloud and each user is associated with the closest base-station. The truthful valuation v_n of each user for purchasing the mobile-edge cloud service to complete the task is randomly assigned from the set $\{1, 2, ..., 20\}$ dollars.

We first evaluate the performance of our proposed resource-efficient offloading algorithm (i.e., delay-aware task graph
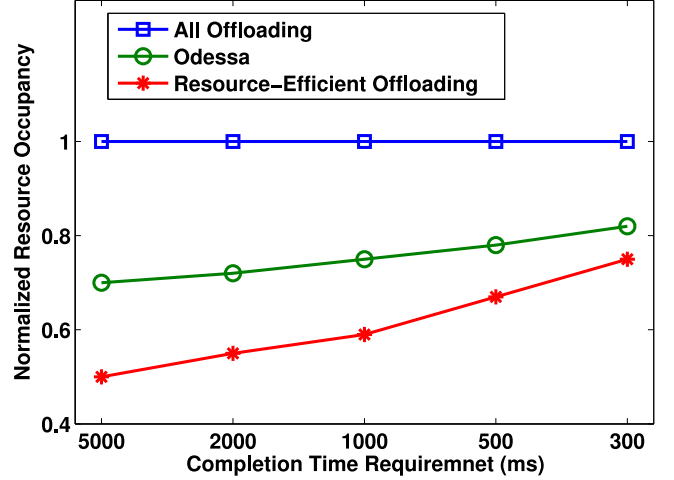

Fig. 3.   Resource occupancy by different computation offloading schemes.

partition in Algorithm 2 in terms of the resource occupancy. As the benchmark, we also implement the following computation offloading schemes: 1) all offloading solution – all task components in the task graph will be offloaded; 2) Odessa scheme [19] – a task component is decided to be executed locally or offloaded remotely in a greedy manner. For each offloading scheme we find the minimum resource occupancy $\Phi_n(q_n, s_n)$ that satisfies the given completion time requirement. The results are shown in Fig. 3 using face recognition [2] as a study case, wherein we use the all offloading solution as the baseline to normalize the resource occupancy performance by the other two schemes. We see that the proposed delay-aware task graph partition in Algorithm 2 can reduce up-to 50% resource occupancy over the all offloading scheme. Compared to Odessa scheme, our algorithm can achieve up-to 22% resource occupancy reduction. This demonstrates the proposed delay-aware task graph partition algorithm has superior resource efficiency.

We then evaluate the proposed approximate admission control in JCC resource allocation. As the benchmark, we consider the following solutions: (1) Optimal solution: we compute the (near) optimal solution of the admission control problem using Cross Entropy method, which is an advanced randomized searching technique and has been shown to be efficient in finding near-optimal solutions to complex combinatorial optimization problems [20]; (2) Coarse greedy solution: instead of using the ranking criterion in (12), we compute the greedy solution according to the coarse ranking criterion of λ_n, i.e., the claimed valuation in the bid; (3) Random selection: we sequentially and randomly select a new user to add into the accepted user set until the capacity constraints are violated.

In the following, we use the optimal solution by Cross Entropy method as the baseline to normalize the system utility (i.e., social welfare) of other solutions. We implement the simulations with different number of users, and mobile-edge clouds in Fig. 4 and 5, respectively. In both cases, we observe that the proposed JCC solution achieves at least 34.7% and 88.3% performance improvement over the solutions of coarse greedy and random selection, respectively. Compared with the optimal solution, we see that the performance loss of the JCC solution is at most
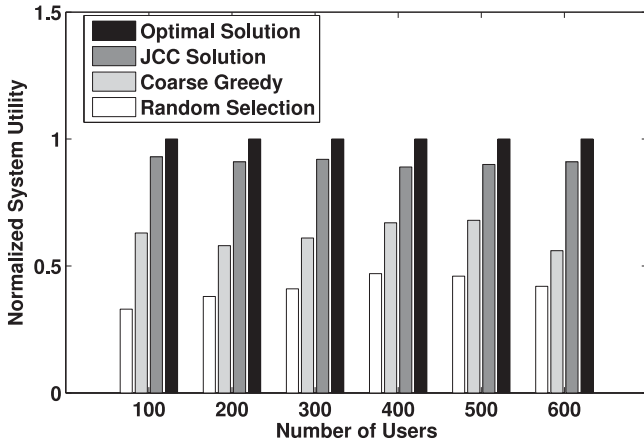
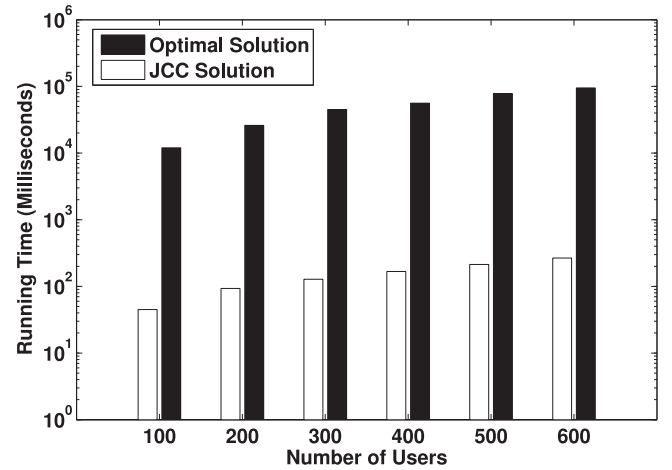Fig. 4.   System performance with different number of users.
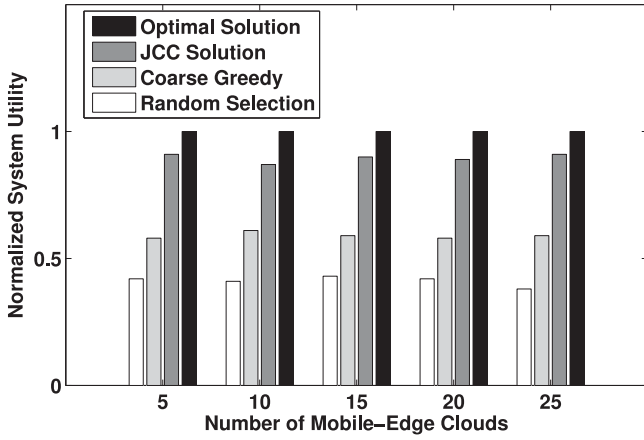

Fig. 6.   Running time of two comparing schemes.

Fig. 5.   System performance with different number of mobile-edge clouds.

Fig. 7.   User net utility with different claimed valuations (truthful valuation $v_n = 8$).

14.3%. This demonstrates of the efficiency of the proposed JCC solution.

We further investigate the running time of the proposed admission control algorithm. We measure the running time in the computing environment of a 64-bit Windows PC with 2.5 GHz Quad core CPU and 16 GB memory. The results are shown in Fig. 6. We see that the JCC mechanism is computationally efficient (with a running time of several milliseconds), with a factor of up to 1000 speed-up compared with the optimal solution by Cross Entropy method. This also demonstrates the proposed admission control algorithm is useful for practical implementation.

We next evaluate the truthfulness of the proposed pricing scheme for JCC resource allocation. We randomly select two users from the system and allow them to claim different valuations in their reports. We show the obtained user utility for these two users in Fig. 7 and 8, respectively. We see that by claimed valuations that are different from the truthful valuations, no users can improve their net utility (i.e., true valuation minus payment), under some cases the utility would be worsen. This demonstrates the proposed pricing scheme for JCC resource allocation is truthful such that no user has the incentive to lie about its valuation.
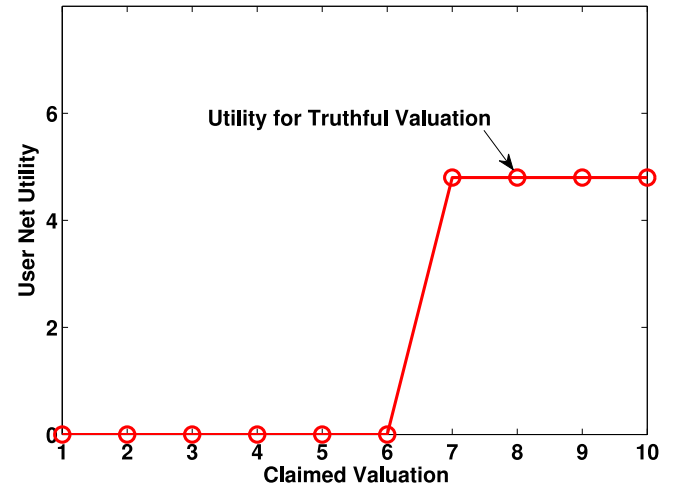
VII. RELATED WORK

Many previous works in mobile cloud computing have focused on the computation offloading problem (e.g., [5]–[7], [21], [22] and references therein). Wen *et al.* in [5] presented an efficient offloading policy by jointly configuring the clock frequency in the mobile device and scheduling the data transmission to minimize the energy consumption. Yang *et al.* in [6] studied the scenario that multiple users share the wireless network bandwidth, and solved the problem of minimizing the delay performance by a centralized heuristic genetic algorithm. Chen *et al.* in [7], [23] proposed a game theoretic approach for designing decentralized multi-user computation offloading mechanism. Note that the computation offloading scheme in our paper is different from the studies above, since the existing works aim at optimizing user's energy efficiency or delay performance, while our objective is to optimize the user's offloading decision in terms of resource utilization efficiency. Moreover, existing works above do not consider the JCC resource allocation. Sardellitti *et al.* in [10] proposed an optimization framework for the energy-efficient JCC resource allocation, by
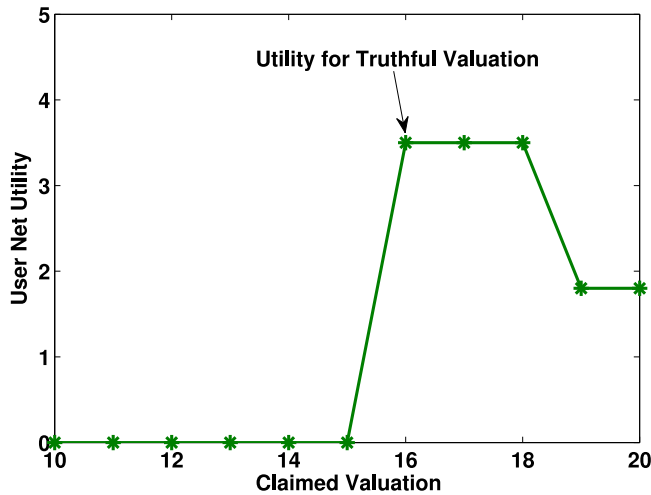
Fig. 8. User net utility with different claimed valuations (truthful valuation $v_n = 16$).

assuming the network operator knows the complete information of all users' applications. Li *et al.* in [24] studied the computation offloading issue for deep learning applications. Tao *et al.* in [25] investigated the performance guaranteed computation offloading for mobile-edge cloud computing. Chen *et al.* in [26] considered to exploit the D2D communications to assist the mobile edge computing. Along a different line, in this paper we introduce a novel paradigm of on-demand mobile-edge cloud service in order to efficiently allocate the JCC resources to those users value them most. Moreover, in our setting each user can locally decide the communication and computation resource profile for computation offloading without the need of reporting the application information to the network operator. This is very useful for reducing the information exchange overhead and protecting user's privacy without exposing the sensitive user specific application information.

There are existing studies in designing truthful pricing scheme for wireless resource auction (e.g., [27]–[29]) and cloud resource auction (e.g., [30]–[32]), which consider either network communication or cloud computation resource allocation in a separate manner. While, in this paper we investigate the joint communication and computation resource allocation for mobile-edge cloud computing – the nexus between cloud computing and wireless networking. Moreover, the truthful pricing scheme in our study builds upon the approximate admission control algorithm using the ranking metric in (12) regarding to the resource occupancy function (3), which is due to user's decision in the computation offloading. For the performance analysis we derive the approximation ratio for both the special and general cases. The analysis and the approximation ratios are completely different from the auction winner selection schemes in existing studies (e.g., [27]–[29], [31], [32]).

VIII. CONCLUSION

Aiming at provisioning flexible mobile-edge cloud service, in this paper we proposed a comprehensive framework consisting of a resource-efficient computation offloading

mechanism for the users and a joint communication and computation (JCC) resource allocation mechanism for the network operator. We first solved the resource-efficient computation offloading problem for each individual user, and derive the optimal communication and computation resource demanding profile that minimizes the resource occupancy and meanwhile satisfies the delay constraint. We tackled the admission control problem for JCC resource allocation, and developed an efficient approximation solution of a low complexity. We also addressed the truthful pricing problem by resorting to the powerful tool of critical value approach. Extensive performance evaluation demonstrates that the proposed mechanisms can achieve superior performance for on-demand mobile-edge cloud computing.

For the future work, we are going to consider the more general case that users may dynamically depart and leave the mobile-edge cloud system. We will take into account users' mobility patterns and devise efficient online resource allocation algorithms to cope with such system dynamics.

REFERENCES

[1] K. Kumar and Y. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *IEEE Comput.*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[2] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Proc. IEEE Symp. Comput. Commun.*, 2012, pp. 59–66.

[3] X. Chen, B. Proulx, X. Gong, and J. Zhang, "Exploiting social ties for cooperative D2D communications: A mobile social networking case," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1471–1484, Oct. 2015.

[4] *Mobile-Edge Computing–Introductory Technical White Paper* European Telecommunications Standards Institute, Sophia Antipolis, France, Sep. 2014.

[5] Y. Wen, W. Zhang, and H. Luo, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," in *Proc. IEEE Int. Conf. Comput.*, 2012, pp. 2716–2720.

[6] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *ACM SIGMETRICS Performance Eval. Rev.*, vol. 40, no. 4, pp. 23–32, 2013.

[7] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[8] Y. Zhang, H. Liu, L. Jiao, and X. Fu, "To offload or not to offload: An efficient code partition algorithm for mobile cloud computing," in *Proc. IEEE 1st Int. Conf. Cloud Netw.*, 2012, pp. 80–86.

[9] Y.-H. Kao and B. Krishnamachari, "Optimizing mobile computational offloading with delay constraints," in *Proc. IEEE Global Commun. Conf.*, 2014, pp. 2289–2294.

[10] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[11] X. Chen, Q. Shi, L. Yang, and J. Xu, "Thriftyedge: Resource-efficient edge computing for intelligent IoT applications," *IEEE Netw.*, vol. 32, no. 1, pp. 61–65, Jan./Feb. 2018.

[12] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 291–303, Feb. 2002.

[13] C. Stein, T. Cormen, R. Rivest, and C. Leiserson, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.

[14] A. Caprara and M. Monaci, "On the two-dimensional knapsack problem," *Operations Res. Lett.*, vol. 32, no. 1, pp. 5–14, 2004.

[15] P. R. Milgrom, *Putting Auction Theory to Work*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[16] Z. Li, F. Xiao, S. Wang, T. Pei, and J. Li, "Achievable rate maximization for cognitive hybrid satellite-terrestrial networks with AF-relays," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 304–313, Feb. 2018.

[17] Z. Zhou, M. Dong, K. Ota, G. Wang, and L. T. Yang, "Energy-efficient resource allocation for D2D communications underlaying cloud-ran-based lte-a networks," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 428–438, Jun. 2016.

[18] Z. Li, B. Chang, S. Wang, A. Liu, F. Zeng, and G. Luo, "Dynamic compressive wide-band spectrum sensing based on channel energy reconstruction in cognitive internet of things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2598–2607, Jun. 2018.

[19] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan, "Odessa: Enabling interactive perception applications on mobile devices," in *Proc. ACM 9th Int. Conf. Mobile Syst., Appl., Serv.*, 2011, pp. 43–56.

[20] R. Y. Rubinstein and D. P. Kroese, *The Cross-entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning.* Berlin, Germany: Springer-Verlag, 2004.

[21] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[22] X. Chen, Z. Zhou, W. Wu, D. Wu, and J. Zhang, "Socially-motivated cooperative mobile edge computing," *IEEE Netw.*, to be published, doi: 10.1109/MNET.2018.1700354.

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

[24] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the internet of things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan./Feb. 2018.

[25] X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, "Performance guaranteed computation offloading for mobile-edge cloud computing," *IEEE Wireless Commun. Lett.*, vol. 6, no. 6, pp. 774–777, Dec. 2017.

[26] X. Chen, L. Pu, L. Gao, W. Wu, and D. Wu, "Exploiting massive D2D collaboration for energy-efficient mobile edge computing," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 64–71, Aug. 2017.

[27] X. Zhou, S. Gandhi, S. Suri, and H. Zheng, "eBay in the sky: Strategy-proof wireless spectrum auctions," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw.*, 2008, pp. 2–13.

[28] L. Gao, Y. Xu, and X. Wang, "Map: Multiauctioneer progressive auction for dynamic spectrum access," *IEEE Trans. Mobile Comput.*, vol. 10, no. 8, pp. 1144–1161, Aug. 2011.

[29] W. Dong, S. Rallapalli, L. Qiu, K. Ramakrishnan, and Y. Zhang, "Double auctions for dynamic spectrum allocation," in *Proc. IEEE Int. Conf. Comput.*, 2014, pp. 709–717.

[30] Y. Liu, A. Liu, S. Guo, Z. Li, Y.-J. Choi, and H. Sekiya, "Context-aware collect data with energy efficient in cyber–physical cloud systems," *Future Generation Comput. Syst.*, to be published, doi: 10.1016/j.future.2017.05.029.

[31] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*, vol. 1, Cambridge, U.K.: Cambridge Univ. Press, 2007.

[32] H. Zhang, B. Li, H. Jiang, F. Liu, A. V. Vasilakos, and J. Liu, "A framework for truthful online auctions in cloud computing with heterogeneous user demands," in *Proc. IEEE Int. Conf. Comput.*, 2013, pp. 1510–1518.

Xu Chen (M'12) received the Ph.D. degree in information engineering from the Chinese University of Hong Kong, Ma Liu Shui, Hong Kong, in 2012, and worked as a Postdoctoral Research Associate with Arizona State University, Tempe, AZ, USA from 2012 to 2014 and a Humboldt Scholar Fellow with the Institute of Computer Science of University of Goettingen, Goettingen, Germany from 2014 to 2016. He is a Full Professor with Sun Yat-sen University, Guangzhou, China, and the Vice Director with National and Local Joint Engineering Laboratory of Digital Home Interactive Applications. He was the recipient of the prestigious Humboldt research fellowship awarded by Alexander von Humboldt Foundation of Germany, 2014 Hong Kong Young Scientist Runner-up Award, 2016 Thousand Talents Plan Award for Young Professionals of China, 2017 IEEE Communication Society Asia-Pacific Outstanding Young Researcher Award, 2017 IEEE ComSoc Young Professional Best Paper Award, Honorable Mention Award of 2010 IEEE international conference on Intelligence and Security Informatics, Best Paper Runner-up Award of 2014 IEEE International Conference on Computer Communications, and Best Paper Award of 2017 IEEE International Conference on Communications. He is currently an Associate Editor of IEEE INTERNET OF THINGS JOURNAL and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) Series on Network Softwarization and Enablers.

Wenzhong Li (M'08) received the B.S. and Ph.D degrees from Nanjing University, Nanjing, China, both in computer science. He was an Alexander von Humboldt Scholar Fellow with the University of Goettingen, Germany. He is currently an Associate Professor with the Department of Computer Science, Nanjing University. He was a Program Cochair of MobiArch 2013 and Registration Chair of ICNP 2013. He was the TPC member of several international conferences and the reviewer of many journals. He is the principle investigator of two funding from National Natural Science Foundation of China, and the Coprinciple investigator of a China–Europe international research staff exchange program. He is the member of ACM and China Computer Federation. He was also the recipient of the Best Paper Award of ICC 2009.

Sanglu Lu (M'04) received the B.S., M.S., and Ph.D. degrees from Nanjing University, Nanjing, China, in 1992, 1995, and 1997, respectively, all in computer science. She is currently a Professor with the Department of Computer Science and Technology and the Deputy Director of State Key Laboratory for Novel Software Technology, Nanjing, China. She has published more than 100 papers in referred journals and conferences in the above areas. Her research interests include distributed computing, pervasive computing, and wireless networks. She is a member of ACM.

Zhi Zhou received the B.S., M.E. and Ph.D. degrees in 2012, 2014 and 2017, respectively, all from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. He is currently a Research Associate Fellow with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. In 2016, he has been a Visiting Scholar with the University of Göttingen. His research interests include edge computing, cloud computing, and datacenter energy management.

Xiaoming Fu (SM'09) received the Ph.D. degree from Tsinghua University, Beijing, China, in 2000. He is a full Professor of computer science with the University of Göttingen, Göttingen, Germany. He was then a Research Staff with TU Berlin before joining Göttingen in 2002, where he has been a Professor and Head of computer networks group since 2007. His research interests lie in networked systems and applications, including mobile and cloud computing, information-centric networking, social networks and big data analysis. He is an IET Fellow, he currently serves on the editorial boards of IEEE Communications Magazine, IEEE Transactions on Network and Service Management, and Computer Communications. He served as Vice Chair (2010–2012) of the IEEE Technical Committee of Computer Communications and Chair (2011–2013) of the IEEE Technical Committee on Internet, and has been the coordinator of EU FP7 MobileCloud, GreenICN, and CleanSky projects, as well as H2020 ICN2020 projects.