An Online Incentive Mechanism for Collaborative Task Offloading in Mobile Edge Computing

Gang Li and Jun Cai, *Senior Member, IEEE*

***Abstract*— This paper discusses incentive mechanism design for collaborative task offloading in mobile edge computing (MEC). Different from most existing work in the literature that was based on offline settings, in this paper, an online truthful mechanism integrating computation and communication resource allocation is proposed. In our system model, upon the arrival of a smartphone user who requests task offloading, the base station (BS) needs to make a decision right away without knowing any future information on i) whether to accept or reject this task offloading request and ii) if accepted, who to execute the task (the BS itself or nearby smartphone users called collaborators). By considering each task's specific requirements in terms of data size, delay, and preference, we formulate a social-welfare-maximization problem, which integrates collaborator selection, communication and computation resource allocation, transmission and computation time scheduling, as well as pricing policy design. To solve this complicated problem, a novel online mechanism is proposed based on the primal-dual optimization framework. Theoretical analyses show that our mechanism can guarantee feasibility, truthfulness, and computational efficiency (competitive ratio of 3). We further use comprehensive simulations to validate our analyses and the properties of our proposed mechanism.**

***Index Terms*— MEC, collaborative offloading, online mechanism design, social welfare maximization.**

I. Introduction

As THE explosive development of smart devices and the advent of more and more new applications such as interactive gaming, video processing, and object recognition, traditional mobile cloud computing architectures become limited in meeting the requirements of these latency-critical applications due to potentially long propagation delay. Furthermore, improving network capacity along with higher data rates and lower latency is the essential requirement in the coming 5G networks [1], [2]. To address these issues, mobile edge computing (MEC), firstly proposed by European Telecommunications Standards Institute (ETSI) [2], is deemed as one of the key technologies for meeting these demands by deploying cloud servers at the edge of radio access networks. A recent trend of MEC is to integrate collaborative offloading framework [3]–[6], where the computational tasks from requesters, such as smartphone users and Ipads, can not only be executed by MEC servers, but also be offloaded to nearby available mobile terminals called collaborators. As a complementary to MEC systems, this collaborative offloading framework can achieve a win-win situation for both idle mobile terminals and network operators [7]. However, implementing collaborative offloading in MEC networks faces many critical issues.

- In reality, collaborative offloading will cost storage, computation and communication resources of both MEC servers and collaborators. From the economical perspective, MEC servers have no responsibility to execute tasks from requesters without any reimbursement. In addition, idle mobile terminals are commonly intelligent and selfish [8], [9], and may not be willing to serve as collaborators. Thus, it is imperative to design an incentive mechanism, which can encourage both idle smartphone users and MEC servers to participate in collaborative offloading by offering them certain rewards for compensating their resource consumption.
- In practice, computation tasks do not arrive at the same time. Thus, it is infeasible for the central controller (e.g., the base station (BS)) to collect knowledge of all tasks before making offloading decisions, which makes the traditional offline solutions infeasible and requests the design of online algorithms.
- Practical computation tasks commonly have some tolerance in delay [7], [10]. Thus, suitably scheduling transmission and computation time periods within these tasks' delay tolerance becomes possible to make full use of network dynamics in channel conditions and computation resources, and to achieve diversity along the time.

In the literature, most work [5]–[7], [11] focused on the offline computation and transmission resource allocation and task assignment, while ignoring the aforementioned practical issues. Note that even though online mechanisms were also proposed in the work [12]–[15] for communication systems and networks, they cannot be applied to our considered scenario, which considers joint computation and transmission resource allocation, and transmission and computation time scheduling. In this paper, we are going to design an online mechanism integrating task executor selection (i.e., the pairing of task executors and requesters), computation and transmission resource allocation, and transmission and computation time scheduling. Unfortunately, designing such mechanism is very challenging due to the following aspects.

Manuscript received May 16, 2019; revised August 25, 2019; accepted October 5, 2019. Date of publication October 18, 2019; date of current version January 8, 2020. The associate editor coordinating the review of this article and approving it for publication was D. Niyato. *(Corresponding author: Jun Cai.)*

The authors are with the Network Intelligence and Innovation Laboratory, Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H4B 1R6, Canada (e-mail: gang.li@mail.concordia.ca; jun.cai@concordia.ca).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2019.2947046

- In an online mechanism, the BS needs to make decisions right away for each arriving task without future task information. Therefore, maintaining a good performance compared to the optimal offline solution becomes very difficult.
- The online mechanism needs to jointly optimize task executor selection, transmission and computation resources allocation, and transmission and computation time scheduling. Such joint optimization problem falls in a typical scope of combinatorial optimization and mixed integer programming, which is extremely difficult to solve.
- The rational relationship between transmission and computation processes and the indetermination of transmission and computation resources allocation introduce nonlinear constraints, which further perplex the formulated optimization problem.
- In a multiple-collaborator task offloading system, task information, such as data size, maximal execution delay, and preference, needs to be known to the BS for optimally managing the network. However, in practice, this information is private and unknown to the BS so that the selfish and intelligent requesters can intentionally report false information so as to maximize their own benefits. This requests that the designed online mechanism should not only incentivize both collaborators and the BS, but also prevent requesters from misreporting their information.

To address all these challenges, in this paper, we propose an efficient online incentive mechanism for collaborative task offloading in MEC networks. In the considered system model, upon the arrival of a requester, it submits its private information to the central controller (i.e., the BS) to request a task offloading. After receiving the request, the BS makes decisions right away on task executor selection, time scheduling, resource allocation, and reward determination. With the objective of maximizing the total social welfare (the summation of utilities of all requesters and task executors), we formulate a complex optimization problem and design an online truthful mechanism based on the primal-dual framework. Specifically, we first convert the optimization problem to its dual form, and then by observing the dual constraints and its corresponding dual variables, we design two marginal price functions which are updated according to the current availability of resources. Based on these marginal price functions, we can decide the best task executor which has the maximal utility, and determine the optimal time scheduling and corresponding resource allocation.

The main contributions of this paper are summarized as follows.

- An online incentive mechanism integrating task executor selection, resource allocation, and time scheduling is proposed for collaborative task offloading in a MEC network. To the best of our knowledge, we are the first to jointly consider all these features for MEC networks.
- We theoretically prove that the proposed online mechanism owns the properties of feasibility, computation efficiency with a competitive ratio of 3, incentive compatibility and individual rationality.
- Numerical simulations have been conducted to justify our theoretical analyses and verify the effectiveness of our proposed online mechanism.

The remainder of this paper is organized as follows. In Section III, the system model is described, and the problem formulation is elaborated. In Section IV, the proposed online auction framework is presented. Numerical results are presented in Section V, followed by concluding remarks in Section VI. In the sequel of this paper, $|\mathcal{X}|$ denotes the cardinality of set $\mathcal{X}$, and $\lceil x \rceil$ means rounding up the value of x.

II. Related Works

In the literature, resource allocation, including radio resource (such as transmission power and channel allocation) and computation resource (such as CPU computation frequency in MEC server), is the most important and widely investigated research topic in MEC networks. The existing works can be divided into three categories based on their optimization objectives, even though the constraints may be variable in different scenarios: i) minimizing the task delay [16], [17], ii) minimizing overall energy consumption [18]–[20], and iii) maximizing the total revenue of the network or MEC server [21]–[24]. However, all these works did not consider the potential collaborative task offloading and online mechanism design.

For collaborative task offloading, different implementation methods have been proposed in [5]–[7] and [11]. In particular, a four-slot three nodes offloading scheme was proposed in [5] where the objective was to minimize the total energy consumption while satisfying the user's latency constraint. Authors in [6] developed E2COM and SDCOM controllers which were centralized and performed task scheduling (i.e. when and where to offload) using online task scheduling algorithm, while authors in [7] proposed an algorithm based on successive convex approximation and geometric programming. In [11], Lu *et al.* developed a heuristic algorithm to solve cooperative offloading at low computation cost based on probabilistic framework that provided the estimation of data delivery probability. However, all these works assumed that idle mobile terminals and MEC servers were always ready to compute the task of a requester. However, in reality, idle mobile terminals and MEC servers may not be willing to share their resources without receiving any remuneration.

Recently, studies have been done to investigate auction models for resource allocation in could computing systems, with quiet few of them [21]–[24] studying auction mechanisms for MEC. Authors in [21] proposed a three-hierarchically architecture in MEC, where an auction-based profit maximization problem was formulated. The work focused on maximizing the gained profit by jointly considering the revenue of serving the Virtual Machine (VM) demands, the electricity cost of running computing and network facilities, and the revenue lost due to network delay. A social welfare maximization problem was formulated in [22] to optimize continuous channel allocation under the constraint of a total number of channels

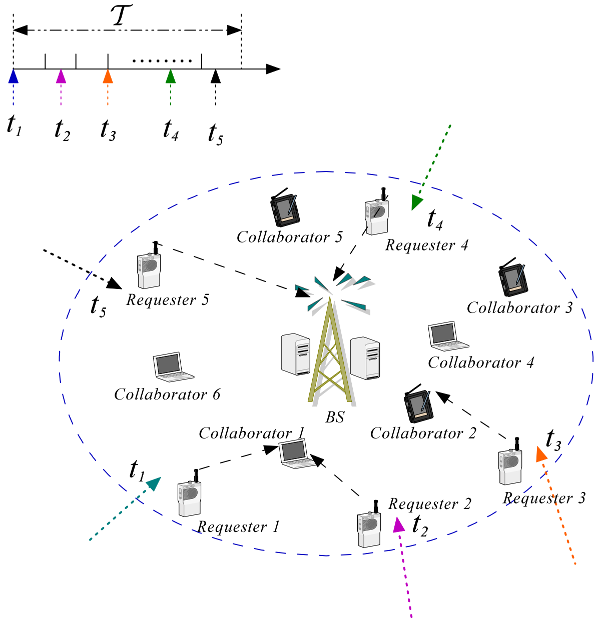

Fig. 1. The system model of collaborative task offloading in mobile edge computing where there are 5 requesters arriving at the network in an online fashion and submitting their requests.

and CPU resources. By revealing the competitions between different miners in blockchain network, the authors in [23] focused on the trading between the cloud/fog computing service provider and miners, and designed an auction mechanism to achieve optimal social welfare. Besides, a double auction model was proposed in [24], in which mobile devices were buyers and cloudlets were sellers. Note that all the aforementioned incentive mechanisms were designed for the non-cooperative task offloading and can only be used in a static setting (i.e., offline setting). In cooperative task offloading, [25] established a computation offloading reverse auction in vehicular networks where a VCG-based mechanism problem is formulated while satisfying the desirable economical properties of truthfulness and individual rationality.

Different from all existing works, by considering dynamic arrival of offloaded tasks, we aim to design an online mechanism, which can not only encourage more participators, but also optimize both communication and computation resource allocation in the collaborative task offloading scenario.

III. System Model

In this section, we describe the system under consideration, and model the interaction between the BS and arrived requesters as an online auction. After that, the corresponding offline optimization problem is formulated.

A. System model

We consider a mobile edge computing network as shown in Fig. 1. A similar system has been discussed in [3] and [7]. The system consists of a BS integrating edge servers and several smartphone users who can also provide computation services, called collaborators. These collaborators are recruited by the BS and are willing to provide computation resources if reimbursements are given. Time to time, there are smartphone users, called requesters, who request computing services. The requesters randomly arrive in a sequence, and we denote t_i as the arrival time instant of requester i. Note that the BS does not have any *a priori* information on requesters' arrival times.

Consider a time-slotted structure with a slot length of Δt. For each arrived requester $i \in \mathcal{U}$, where $\mathcal{U}$ denotes the set of all requesters, let $\mathcal{M}_i$ be the set of available collaborators that can provide computation services to it. Note that the set $\mathcal{M}_i$ could be available to requester i by applying the discovery approach [28]. We further define $\mathcal{N}_i = \mathcal{M}_i \cup 0$, where the index 0 represents the BS. Obviously, $\mathcal{N}_i$ consists of all collaborators and the BS that requester i can offload its task to. For the notation simplification, we will use the term "task executor" to denote any collaborator or the BS throughout this paper. We further let $\mathcal{M}$ be the set of all collaborators. Note that since user mobility cannot affect the offloading process in the coverage of one MEC server, we don't consider it in this paper. This is because if the collaborator moves away after the arrival of the requester's task, the task will fail to be transmitted to the collaborator, and no rewards can be obtained. So, there are no incentives for collaborators to move. For the movement of requester, if the requester moves around the corresponding collaborator, D2D technologies [29] can be applied to transmit the computational results and the reimbursements between requester and collaborator. Otherwise, results and reimbursements can also be received and transmitted through the cellular link.

The task from requester $i \in \mathcal{U}$ is denoted as $T_i = (s_i, \tau_i)$, where s_i is the size (in bits) of the offloaded task and τ_i is the maximal tolerance delay. Note that the task offloading to the collaborator can be done through a D2D link [29]. Each task i requires Q_i CPU cycles for execution and can be calculated by $Q_i = \kappa_i s_i$ [17] where κ_i is the CPU cycles coefficient. We also define the allocated CPU frequency at task executor j as $f_{i,j}$. Then, the required computation time at task executor j for task i equals

$$I^C_{i,j} = \frac{Q_i}{f_{i,j}} = d_{i,j} - a_{i,j}, \tag{1}$$

where $a_{i,j}$ and $d_{i,j}$ denote the starting and ending computation time instants, respectively. In addition, given that each requester i is allocated an orthogonal channel with bandwidth $\phi_{i,j}$ for task offloading to task executor j, the transmission rate from requester i to task executor j equals

$$r_{i,j} = \phi_{i,j} \log_2(1 + \gamma_{i,j}), \tag{2}$$

where $\gamma_{i,j} = \frac{\Lambda_{i,j}|h_{i,j}|^2}{\sigma^2}$ is the signal to noise ratio (SNR), σ^2 is the average power of background noise, and $\Lambda_{i,j}$ and $h_{i,j}$ are the transmission power and the channel gain between requester i and task executor j, respectively. Thus, the transmission time from requester i to task executor j can be calculated as

$$I^T_{i,j} = \frac{s_i}{r_{i,j}} = o_{i,j} - g_{i,j}, \tag{3}$$

where $g_{i,j}$ and $o_{i,j}$ denote the starting and ending transmission time instants, respectively. Note that since both the available computation and transmission resources are time-varying, both transmission time and computation time should be optimally determined for each offloading task. Therefore, $g_{i,j}$, $o_{i,j}$, $a_{i,j}$, and $d_{i,j}$ are decision variables. To meet the task delay requirement, we need

$$d_{i,j} - t_i \le \tau_i. \tag{4}$$

In (4), similar to studies in [30] and [31], we ignore the time for the task executor to send the computation result back to the requester because the data size of outcomes for many applications is commonly very small. In summary, the whole operation procedure of this system is described as follows.

Step 1. Upon the arrival of requester i, it submits multiple bids to the BS, denoted by $B_{i,j} = (T_i, t_i, v_{i,j}), j \in \mathcal{N}_i$, where $v_{i,j}$ is the valuation of requester i to task executor j, which represents its preference to offload the task to task executor j.[1]

Step 2. After collecting the bids from requester i, the BS makes a decision, denoted by a binary variable $x_{i,j}$, whether to accept this requester. $x_{i,j} = 1$ means requester i is accepted. Otherwise, $x_{i,j} = 0$. The BS further determines what are the optimal transmission and computation time instants for this task.

Step 3. The BS sends the optimal results obtained in Step 2 to requester i and notifies the selected task executor to prepare for the task execution.

Step 4. After the task is completed, requester i will be charged by $p_{i,j}$, which is another decision variable, and the task executor returns computation results to it.

Obviously, the interactions between task executors and requesters can be model as an online auction, where the BS is the auctioneer, requesters are buyers, and all the task executors are sellers. Requesters may strategically misreport their private information (i.e., $B_{i,j}$) in order to get more benefits. For example, requester i, who will lose in the auction, may submit false bid $B^{'}_{i,j}$, where $T^{'}_i = T_i, t^{'}_i = t_i$, and $v^{'}_{i,j} > v_{i,j}$. In this case, this requester have higher chance to win the auction than reporting truthfully. Thus, a truthful incentive mechanism is necessary for our considered system. Following the previous discussions, the utilities of requester i and the task executor j can be respectively expressed as

$$u_i = v_{i,j} - p_{i,j}, \tag{5}$$

$$u_j = p_{i,j} - e_{i,j} c_j, \tag{6}$$

where $e_{i,j} = Q_i \xi_j f^2_{i,j}$ and c_j are the energy consumption for executing the task i and the unit energy cost of task executor j, respectively, and ξ_j is the energy consumption coefficient [32]. For notational clarity, the commonly used abbreviations and notations are summarized in Table I.

TABLE I
COMMONLY USED NOTATIONS

Notation	Interpretation
t_i	Arriving time instant
Δt	A time slot length
$\mathcal{M}_i$	Set of available collaborators of requester i
$\mathcal{U}$	Set of all requesters
$\mathcal{N}_i$	Set includes $\mathcal{M}_i$ and the BS
τ_i	Maximal tolerance delay
s_i	Size of task
Q_i	Required CPU cycles for task i
$f_{i,j}$	Allocated CPU frequency for task i task executor j
$I^C_{i,j}$	Required computation time at task executor j for task i
$a_{i,j}$, $d_{i,j}$	Starting and ending computation time instants
$I^T_{i,j}$	Required transmission time from i to j
$g_{i,j}$, $o_{i,j}$	Starting and ending transmission time instants
$v_{i,j}$	Valuation of requester i to j
$x_{i,j}$	Binary variable
$p_{i,j}$	Payment from requester i to j
u_i	Utility of requester i
$e_{i,j}$	Energy consumption at task executor j
c_j	Unit cost per energy
$\ell^1_{i,j}$ and $\ell^2_{i,j}$	Sets of all the feasible transmission and computation time scheduling

B. Problem Formulation

Our target is to design an online auction which can satisfy the following properties.

- Incentive Compatibility (IC), which means no requesters can gain more utilities by misreporting their bids. The property IC sometimes is also called "truthfulness" in the literature;
- Individual Rationality (IR), which guarantees utilities of all requesters are no less than zero;
- Social Welfare Maximization (SWM). Here, social welfare is defined as a summation of all participators' utilities, and can be calculated as

$$SW = \sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{N}_i} w_{i,j} x_{i,j}, \tag{7}$$

  where $w_{i,j} = v_{i,j} - e_{i,j} c_j$.
- Competitive Ratio (CR), which is defined as the ratio of the results derived by the optimal offline solution given all future information over the corresponding online one. CR is a vital metric to measure the performance of the online mechanism. The larger CR is, the worse the online performance will be;
- Computational Efficiency (CE), which means the designed online mechanism should be run in polynomial time.

If the information about all tasks is known, we can formulate the corresponding offline optimization problem (MSW) as

$$\max_{\boldsymbol{X},\boldsymbol{G},\boldsymbol{O},\boldsymbol{A},\boldsymbol{D},\boldsymbol{P}} \sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{N}_i} w_{i,j} x_{i,j} \qquad \text{MSW}$$

$$\begin{aligned}
\text{s.t. } & C_1: \sum_{j \in \mathcal{N}_i} x_{i,j} \le 1, \quad \forall i \in \mathcal{U}; \\
& C_2: \sum_{j \in \mathcal{N}_i} (o_{i,j} - a_{i,j}) x_{i,j} \le 0, \quad \forall i \in \mathcal{U}; \\
& C_3: \text{(1), (3) and (4);} \\
& C_4: \sum_{\substack{i \in \mathcal{U}: \\ a_{i,j} \le t \le d_{i,j}}} f_{i,j} x_{i,j} \le F_j, \quad \forall t \in \mathcal{T},\ j \in \mathcal{M} \cup \text{BS}; \\
& C_5: \sum_{\substack{i \in \mathcal{U}: \\ a_{i,j} \le t \le d_{i,j}}} s_i x_{i,j} \le S_j, \quad \forall t \in \mathcal{T},\ j \in \mathcal{M} \cup \text{BS}; \\
& C_6: \sum_{\substack{i \in \mathcal{U}: \\ g_{i,j} \le t \le o_{i,j}}} \sum_{j \in \mathcal{N}_i} \phi_{i,j} x_{i,j} \le W, \quad \forall t \in \mathcal{T};
\end{aligned}$$

[1] Since the collaborators are heterogeneous in terms of available computation resources and geographical locations, which makes the channel conditions between the collaborators and the requesters different, each requestor values the nearby collaborators differently.

$$
\begin{aligned}
&C_7: \sum_{j\in\mathcal{N}_i}(v_{i,j}-p_{i,j})x_{i,j}\geq 0, \quad \forall\, i\in\mathcal{U};\\
&C_8: \sum_{j\in\mathcal{N}_i}(v_{i,j}-p_{i,j})x_{i,j}\geq \sum_{j\in\mathcal{N}_i}(\widetilde{v}_{i,j}-\widetilde{p}_{i,j})x_{i,j}, \forall\, i\in\mathcal{U};\\
&C_9: x_{i,j}\in\{0,1\},\, o_{i,j}\in\{t_i,\tau_i\},\, g_{i,j}\in\{t_i,\tau_i\},\\
&\quad a_{i,j}\in\{t_i,\tau_i\},\, d_{i,j}\in\{t_i,\tau_i\},\ \forall\, i\in\mathcal{U},\ j\in\mathcal{N}_i.
\end{aligned}
$$

where F_j and S_j denote the maximal CPU frequency and storage capacity of the executor j, respectively, W is the whole bandwidth of the system, and $\mathcal{T}$ is the set of all time slots. Decision variables are $\boldsymbol{X} = \{x_{i,j}\}_{i\in\mathcal{U},j\in\mathcal{N}_i}$, $\boldsymbol{G} = \{g_{i,j}\}_{i\in\mathcal{U},j\in\mathcal{N}_i}$, $\boldsymbol{O} = \{o_{i,j}\}_{i\in\mathcal{U},j\in\mathcal{N}_i}$, $\boldsymbol{A} = \{a_{i,j}\}_{i\in\mathcal{U},j\in\mathcal{N}_i}$, $\boldsymbol{D} = \{d_{i,j}\}_{i\in\mathcal{U},j\in\mathcal{N}_i}$, and $\boldsymbol{P} = \{p_{i,j}\}_{i\in\mathcal{U},j\in\mathcal{N}_i}$. Constraint C_1 ensures that each requester can offload its task to at most one task executor. Constraint C_2 means the transmission process occurs before the computation process for any task. Constraints C_3 represents the time rationality and delay requirement. Constraints C_4 and C_5 indicate constrains on the allocated CPU frequencies and storage resources at any task executor, respectively. Constraint C_6 specifies that the allocated bandwidths cannot exceed W, and constraints C_7 and C_8 are the requirements of IR and IC, respectively. Constraint C_9 defines decision variables $\boldsymbol{G}, \boldsymbol{O}, \boldsymbol{A}, \boldsymbol{D}$, and $\boldsymbol{P}$ to be continuous and $\boldsymbol{X}$ to be binary variables.

Obviously, this formulated offline optimization problem is a mixed integer problem and is usually NP hard [33]. In addition, this formulation requires a complete information on system operation in the future. In the follows, we are going to design a novel online mechanism to find solutions on the fly.

IV. Online Mechanism Design

In this section, we try to design an online mechanism to find solutions to the problem (MSW). Note that since the payments are not in the objective function in (MSW) but only in the constraints C_7 and C_8, we can decouple (MSW) into two subproblems without losing optimality: an allocation subproblem (including task executor selection, resource allocation, and time scheduling) and a payment rule subproblem. In the following, we first reformulate the offline problem, and then solve the allocation problem. After that a corresponding payment scheme will be designed to not only satisfy IC, but also maintain IR.

A. Problem Reformulation

Since constraints C_4 and C_5 in (MSW) have the same structure, we combine them together as

$$C_{10}: \sum_{\substack{i\in\mathcal{U}:\\ a_{i,j}\leq t\leq d_{i,j}}} r^k_{i,j}x_{i,j}\leq R^k_j, \quad \forall t\in\mathcal{T},\, \forall j\in\mathcal{M},\, \forall k\in\mathcal{K},$$

where

$$r^k_{i,j} = \begin{cases} s_i & \text{if } j\in\mathcal{N}_i \text{ and } k=1;\\ f_{i,j} & \text{if } j\in\mathcal{N}_i \text{ and } k=2,\\ 0 & \text{otherwise;}\end{cases} \tag{8}$$

$$R^k_j = \begin{cases} S_j & \text{if } j\in\mathcal{M} \text{ and } k=1;\\ F_j & \text{if } j\in\mathcal{M} \text{ and } k=2;\\ 0 & \text{otherwise;}\end{cases} \tag{9}$$

$\mathcal{K}=\{1,2\}$, and $l_{i,j}=l^1_{i,j}\cup l^2_{i,j}$ denotes all feasible transmission and computation time scheduling pairs from requester i to task executor j with satisfaction of constraints C_2, C_3, and C_9. Let $l^1_{i,j}=\{l^1_{i,j}(1), l^1_{i,j}(2),\cdots\}$ and $l^2_{i,j}=\{l^1_{i,j}(1), l^2_{i,j}(2),\cdots\}$ are sets of all the feasible transmission and computation time scheduling, respectively, and each entry $l^1_{i,j}(\ell)$ or $l^2_{i,j}(\ell)$ indicates the ℓ-th feasible scheduling scheme. Let $\mathcal{L}_{i,j}$ be the index set of all feasible solutions from requester i to task executor j. Note that $\mathcal{L}_{i,j}$ has a potentially exponential number of feasible solutions with respect to the decision variables $\boldsymbol{G}$, $\boldsymbol{O}$, $\boldsymbol{A}$, and $\boldsymbol{D}$.

Then, the allocation problem can be formulated from the original (MSW) as

$$
\begin{aligned}
&\max_{\hat{\boldsymbol{X}}} \sum_{i\in\mathcal{U}}\sum_{j\in\mathcal{N}_i}\sum_{\ell\in\mathcal{L}_{i,j}} w^\ell_{i,j}x^\ell_{i,j} \qquad\qquad \text{EQMSW}\\
&\text{s.t. } C_{11} \sum_{j\in\mathcal{N}_i}\sum_{\ell\in\mathcal{L}_{i,j}} x^\ell_{i,j}\leq 1,\ \forall i\ \in\mathcal{U}\\
&\quad C_{12}: \sum_{i\in\mathcal{U}}\sum_{\ell:t\in l^2_{i,j}(\ell)\in l^2_{i,j}} r^k_{i,j}x^\ell_{i,j}\leq R^k_j, \quad \forall t\in\mathcal{T}, \forall j\in\mathcal{M},\\
&\qquad\qquad k\in\mathcal{K};\\
&\quad C_{13}: \sum_{i\in\mathcal{U}}\sum_{j\in\mathcal{N}_i}\sum_{\ell:t\in l^1_{i,j}(\ell)\in l^1_{i,j}} \phi_{i,j}x^\ell_{i,j}\leq W, \quad \forall t\in\mathcal{T};\\
&\quad C_{14}: x^\ell_{i,j}\in\{0,1\},\ \forall i\in\mathcal{U},\ j\in\mathcal{N}_i,\ \ell\in\mathcal{L}_{i,j}.
\end{aligned}
$$

where $\hat{\boldsymbol{X}} = \{x^\ell_{i,j}, i\in\mathcal{U}, j\in\mathcal{N}_i, \ell\in\mathcal{L}_{i,j}\}$ are new decision variables; $w^\ell_{i,j}=v_{i,j}-c_je^\ell_{i,j}$, where $e^\ell_{i,j}$ is the energy consumption at task executor j when the ℓ-th feasible scheduling scheme is selected. In order to devise an online mechanism with sound CR, we resort to its dual problem. The dual problem of (EQMSW) can be formulated as follows by relaxing the constraint C_{14} into any value between 0 and 1.

$$
\begin{aligned}
&\min_{u,\,\hat{p}} \sum_{i\in\mathcal{U}}u_i+\sum_{t\in\mathcal{T}}W\hat{p}_t+\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{M}}\sum_{k\in\mathcal{K}}R^k_j\hat{p}^k_{j,t} \qquad \text{EQDP}\\
&\text{s.t. } C_{15}: u_i+\sum_{t\in l^2_{i,j}(\ell)}\sum_{k\in\mathcal{K}}r^k_{i,j}\hat{p}^k_{j,t}+\sum_{t\in l^1_{i,j}(\ell)}\phi_{i,j}\hat{p}_t\geq w_{i,j},\\
&\qquad\qquad \forall i\in\mathcal{U},\ j\in\mathcal{N}_i,\ \ell\in\mathcal{L}_{i,j};\\
&\quad C_{16}: u_i\geq 0,\ \hat{p}^k_{j,t}\geq 0,\ \hat{p}_t\geq 0,\ \forall i\in\mathcal{U},\ j\in\mathcal{M}_i,\ k\in\mathcal{K}.
\end{aligned}
$$

where u_i, $\hat{p}_t$ and $\hat{p}^k_{j,t}$ are the dual variables corresponding to the constraints C_{11}, C_{12}, and C_{13}, respectively. Note that the dual variables $\hat{p}^k_{j,t}$ can be interpreted as the marginal price of task executor j's available computation frequencies and storages resources (i.e., $k=1$ or 2) at time slot t, while dual variable $\hat{p}_t$ can be regarded as the marginal price of the available bandwidth in the network. Thus, $\sum\limits_{t\in l^2_{i,j}(\ell)}\sum\limits_{k\in\mathcal{K}}r^k_{i,j}\hat{p}^k_{j,t}$ and $\sum\limits_{t\in l^1_{i,j}(\ell)}\phi_{i,j}\hat{p}_t$ represent the total computation cost and the total transmission cost, respectively. Moreover, u_i can be considered as the utility of requester i. In the following sections, we will apply these observations to design an online mechanism to address problem (MSW).

B. Online Mechanism

In our formulated online mechanism, we need to decide whether to accept a new task upon its arrival and which task executor should be assigned as well as how much the requester should be charged. Our basic idea is that if the BS decides to assign the current requester i's task to task executor j, we increase the unit price of task executor j's resource based on the fact that it will have less resources, and then apply these updated prices to decide the acceptance of future arrived requesters.

1) Allocation Rule: Under the consideration of IC and IR, u_i in constraint C_{13} has to be maximized and greater than zero. In addition, according to the KKT condition [35] in the prima-dual framework, if requester i is accepted (i.e., $x^{\ell}_{i,j} = 1$), we have

$$u_i = w_{i,j} - (\sum_{t\in l^2_{i,j}(\ell)} \sum_{k\in\mathcal{K}} r^k_{i,j}\hat{p}^k_{j,t} + \sum_{t\in l^1_{i,j}(\ell)} \phi_{i,j}\hat{p}_t). \quad (10)$$

Combining these two requirements together, u_i can be written as

$$u_i=\max\{0, \max_{\substack{j\in\mathcal{N}_i,\\ \ell\in\mathcal{L}_{i,j}}}\{w_{i,j} -(\sum_{t\in l^2_{i,j}(\ell)} \sum_{k\in\mathcal{K}} r^k_{i,j}\hat{p}^k_{j,t}+ \sum_{t\in l^1_{i,j}(\ell)} \phi_{i,j}\hat{p}_t)\}\}. \quad (11)$$

From (11), we can design the following allocation rule. Upon the arrival of requester i, we choose a task executor in the set $\mathcal{N}_i$ and a scheduling scheme in set $l_{i,j}$ so that u_i is maximized. We denote such best task executor and the scheduling scheme as j^* and $l_{i,j}(\ell^*)$, respectively. Note that the scheduling scheme $l_{i,j}(\ell^*)$, which maximizes the utility of requester i, is referred to as the optimal scheduling scheme at the collaborator j. If at optimum, (10) is larger than zero, requester i's task is accepted; otherwise, it is rejected. Note that we also refer to the above allocation rule as the acceptance condition in this paper.

2) Payment Design: As indicated before, the marginal prices increase with the acceptance of requesters and the designed updating rule is vital to the achievable competitive ratio of our online auction which will be discussed later. The designed marginal price updating rule should follow the following three requirements: (i) at the beginning of the auction, the price should be set sufficiently low in order to allow the acceptance of coming requesters; (ii) after allocating resources for each accepted requester, prices should be increased rapidly to save resources for the future requesters with high valuations; and (iii) if some resources of any task executor are run out at certain time slot, the prices should be set high enough so that no requesters' tasks can be accepted. By considering all these requirements, for any task executor j, we design the marginal prices updating rule as follows

$$\hat{p}^k_{j,t} = \hat{p}^k_{j,t}(1 + \frac{r^k_{i,j}}{R^k_j}) + \frac{r^k_{i,j}}{\Gamma_{i,j}R^k_j},\ \forall t \in [g_{i,j}, o_{i,j}],\ \forall k \in \mathcal{K}, \quad (12)$$

$$\hat{p}_t = \hat{p}_t(1 + \frac{\phi_{i,j}}{W}) + \frac{\phi_{i,j}}{\Phi_{i,j}W},\ \forall t \in [a_{i,j},\ d_{i,j}], \quad (13)$$

where $\Gamma_{i,j} = \frac{\sum_{k\in\mathcal{K}} r^k_{i,j} I^C_{i,j}}{w_{min}}$, $\Phi_{i,j} = \frac{I^T_{i,j}\phi_{i,j}}{w_{min}}$, and w_{min} is the minimal valuable of $w_{i,j}$, which can be estimated from the historical data, and both $\Gamma_{i,j}$ and $\Phi_{i,j}$ can be calculated based on the outputs of the allocation rule. Thus, the price for a requester i to pay can be determined by

$$\begin{cases} p_{i,j}=p^1_{i,j}+p^2_{i,j}= \sum_{t\in l^2_{i,j}(\ell)} \sum_{k\in\mathcal{K}} r^k_{i,j}\hat{p}^k_{j,t}+ \sum_{t\in l^1_{i,j}(\ell)} \phi_{i,j}\hat{p}_t+e_{i,j}c_j, \\ \qquad\qquad\qquad\qquad \text{if } i \text{ is accepted;} \\ p_{i,j} = 0, \qquad\qquad\quad \text{if } i \text{ is rejected;} \end{cases}$$

3) Scheduling Design: To implement Algorithm 2, the maximization problem in (11) needs to be solved. Since we may confront exponential numbers of feasible solutions, it is inefficient to find the best solution through exclusive searching. To address this issue, we propose a new polynomial time method as follows.

From (11), the original optimization problem can be equivalently converted to one which minimizes the summation of $p^1_{i,j}$, $p^2_{i,j}$, and $e_{i,j}c_j$. Note that since we try to arrange a certain number of time slots to complete the transmission and computation processes for a task, the newly formulated problem for requester i offloading task to the task executor j becomes

$$\begin{aligned} \beta_{i,j} = \min_{\substack{y_j(t), z_j(t),\\ N_{\phi_j}, N_{f_{i,j}}}} & \sum_{t\in[t_i,t_i+\tau_i]} \{\frac{h(t)}{N_{\phi_j}} y_j(t) + (\frac{c_1(t)}{N_{f_{i,j}}} + c_2(t))z_j(t)\} \\ & + \frac{c_3}{N^2_{f_{i,j}}} \\ \text{s.t. } & C_{15} : y_j(t) < z_j(t),\ \forall t \in [t_i, t_i + \tau_i]; \qquad \text{TSP} \\ & C_{16} : \sum_{t\in[t_i,t_i+\tau_i]} y_j(t) = N_{\phi_j}; \\ & C_{17} : \sum_{t\in[t_i,t_i+\tau_i]} z_j(t) = N_{f_{i,j}}; \\ & C_{18} : y_j(t) \in \{0, 1\},\ z_j(t) \in \{0, 1\},\ t \in [t_i, t_i + \tau_i]. \end{aligned}$$

where $h(t) = \frac{s_i\hat{p}_t}{\Delta t \log 2(1+\gamma_{i,j})}$, $c_1(t) = \frac{Q_i\hat{p}^1_{j,t}}{\Delta t}$, $c_2(t) = s_i\hat{p}^2_{j,t}$, and $c_3 = \frac{c_jQ^3_i\xi_j}{\Delta t^2}$ for any pair i and j; $y_j(t)$ and $z_j(t)$ are two new binary scheduling decision variables. If $y_j(t)$ or $z_j(t)$ equals 1, it means requester i transmits the task to task executor j or task executor j executes the task at time slot t, respectively; N_{ϕ_j} and $N_{f_{i,j}}$ denote the total required transmission and computation time slots at task executor j, respectively. Due to the integral decision variables and the nonlinear objective, it's nontrivial to solve problem (TSP) directly. Instead, we decouple it by letting the optimal dividing time slot between transmission period and computation period be $\bar{t}_{i,j} \in [t_i, t_i + \tau_i]$. Then, the scheduling problem (TSP) can be equivalently transformed into two subproblems as

$$\begin{aligned} \beta^1_{i,j} = \min_{y_j(t), N_{\phi_j}} & \sum_{t\in[t_i,\bar{t}_{i,j}]} \frac{h(t)}{N_{\phi_j}} y_j(t) \qquad \text{SubP1} \\ \text{s.t. } & C_{16}, \text{ and } y_j(t) \in \{0, 1\} \end{aligned}$$

$$\begin{aligned} \beta^2_{i,j} = \min_{z_j(t), N_{f_{i,j}}} & \sum_{t\in(\bar{t}_{i,j},t_i+\tau_i]} (\frac{c_1(t)}{N_{f_{i,j}}} + c_2(t))z_j(t) + \frac{c_3}{N^2_{f_{i,j}}} \\ & \qquad \text{SubP2} \\ \text{s.t. } & C_{17}, \text{ and } z_j(t) \in \{0, 1\} \end{aligned}$$

Lemma 1: The optimal solution of subproblem (SubP1) is obtained when $N_{\phi_j} = 1$, $t^* = \arg\min_{t\in[t_i,\bar{t}_{i,j}]} h(t)$.

Proof: The proof of contradiction method is applied to prove our statement. We first sort $h(t)$ during the period of $[\bar{t}_{i,j}, t_i + \tau_i]$ in a non-decreasing order into $h(t^1) \le h(t^2) \le h(t^3) \le \cdots$. According to Lemma 1, we choose $h(t^1)$ as the optimal solution of (SubP1). On the other hand, if there exist $N_{\phi_j} = N$ continuous transmission time slots, for example $h(t^{n_1}), h(t^{n_2}), \cdots, h(t^{n_N})$, whose $\beta^1_{i,j}$ is smaller than $h(t^1)$, then, we have

$$\tfrac{h(t^{n_1})+h(t^{n_2})+\cdots h(t^{n_N})}{N} < h(t^1). \tag{14}$$

However, this contradicts with the fact that $h(t^1) \le h(t^{n_v}), v = 1, 2, \cdots, N$. Thus, our conclusion holds for (SubP1). This completes the proof. ■

Lemma 2: Let $\beta^{2,1}_{i,j}(N) = \sum_{t\in[1,+\infty]} (\frac{c_1(t)}{N} + c_2(t))z_j(t)$ be the value under the optimal scheduling when $N_{f_{i,j}} = N$ and let $\beta^{2,2}_{i,j}(N) = \frac{c_3}{N^2}$. Then, we have $\beta^{2,1}_{i,j}(N)$ is an increasing function with respective to N and there exists at most one intersection point between $\beta^{2,1}_{i,j}(N)$ and $\beta^{2,2}_{i,j}(N)$.

Proof: This statement is obtained by using the analytical approach. We first compare objective values of $\beta^{2,1}_{i,j}(N)$ and $\beta^{2,1}_{i,j}(N + 1)$. Let $t^{n_1}, t^{n_2}, \cdots t^{n_N}$ be the best N numbers of continuous time slots, which means when $N_{f_{i,j}} = N$, the objective of (SubP1) is minimized by selecting these time slots. Likewise, denote $t^{m_1}, t^{m_2}, \cdots t^{m_{(N+1)}}$ as the optimal continuous time slots when $N_{f_{i,j}} = N + 1$. Then, we have

$$\begin{aligned}
&\beta^{2,1}_{i,j}(N) - \beta^{2,1}_{i,j}(N+1) = \frac{\sum\limits_{v=1}^{N} c_1(t^{n_v})}{N} + \sum_{v=1}^{N} c_2(t^{n_v}) \\
&-(\frac{\sum\limits_{v=1}^{N+1} c_1(t^{m_v})}{N+1} + \sum_{v=1}^{N+1} c_2(t^{m_v})) \Rightarrow (N+1)(\beta^{2,1}_{i,j}(N) \\
&-\beta^{2,1}_{i,j}(N+1)) \\
&= (N+1)(\frac{\sum\limits_{v=1}^{N} c_1(t^{n_v})}{N} + \sum_{v=1}^{N} c_2(t^{n_v})) - (\sum_{v=1}^{N+1} c_1(t^{m_v}) \\
&+(N+1)\sum_{v=1}^{N+1} c_2(t^{m_v})) = (N+1)\underbrace{(\frac{\sum\limits_{v=1}^{N} c_1(t^{n_v})}{N} + \sum_{v=1}^{N} c_2(t^{n_v}))}_{\textcircled{1}} \\
&\underbrace{-N(\frac{\sum\limits_{v=1}^{N} c_1(t^{m_v})}{N} + \sum_{v=1}^{N} c_2(t^{m_v}))}_{\textcircled{2}} \\
&\underbrace{-(\frac{Nc_1(t^{m_{N+1}})}{N} + \sum_{v=1}^{N} c_2(t^{m_v}))}_{\textcircled{3}} - (N+1)c_2(t^{m_{N+1}}).
\end{aligned} \tag{15}$$

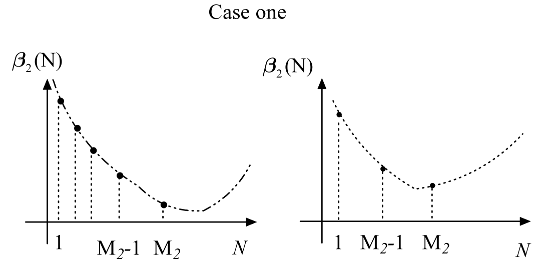

Fig. 2. Relationship between $\beta_2(N)$ and N.

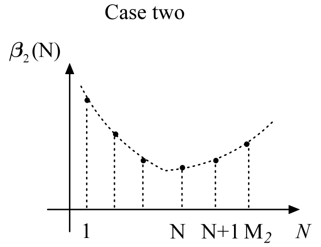

Fig. 3. Relationship between $\beta_2(N)$ and N.

Since ① is the optimal objective value when $N_{f_{i,j}} = N$, we have $(N + 1)\times\textcircled{1} < N\times\textcircled{2}+\textcircled{3}$. Thus, we have $\beta^{2,1}_{i,j}(N) < \beta^{2,1}_{i,j}(N + 1)$, which means $\beta^{2,1}_{i,j}(N)$ is an increasing function with respective to N. Moreover, $\beta^{2,2}_{i,j}(N)$ is a decreasing function with respective to N and $\beta^{2,2}_{i,j}(+\infty) = 0 < \beta^{2,1}_{i,j}(+\infty)$. Thus, $\beta^{2,1}_{i,j}(N)$ and $\beta^{2,2}_{i,j}(N)$ have one intersection point only when $\beta^{2,1}_{i,j}(N) = \beta^{2,2}_{i,j}(N)$. This completes the proof. ■

Based on Lemma 1, the allocated transmission bandwidth for requester i is always $\phi_{i,j} = \frac{s_i}{\Delta t \, \log_2(1+\gamma_{i,j})}$. According to Lemma 2, there must exist a $\overline{N}$ which can minimize the value of $\beta^{2,1}_{i,j}(N)+\beta^{2,2}_{i,j}(N)$. Note that $\beta^{2,1}_{i,j}(N)+\beta^{2,2}_{i,j}(N)$ decreases when $N < \overline{N}$, but increases when $N > \overline{N}$. If there are M_2 available time slots during $(\bar{t}_{i,j}, t_i+\tau_i]$, we apply the following strategies to get the optimal solution of subproblem (SubP2).

- If $\beta^2_{i,j}(1) > \beta^2_{i,j}(M_2 - 1) > \beta^2_{i,j}(M_2)$, we choose $\beta^2_{i,j}(M_2)$ and the corresponding scheduling scheme, denoted as the set π^*, as the optimal solution, as shown in Fig. 2;
- Otherwise, we apply sequential search to compare the values of $\beta^2_{i,j}(N + 1)$ and $\beta^2_{i,j}(N)$ till $\beta^2_{i,j}(N + 1) > \beta^2_{i,j}(N)$. We then choose $\beta^2_{i,j}(N)$ and the corresponding scheduling scheme, denoted as the set π^*, as the optimal solution, as shown in Fig. 3.

Obviously, for the worst case, we only need $\frac{(\overline{N}+1)(2M_2-\overline{N})}{2}+\overline{N}$ comparisons to reach the optimal solution, which is much more computationally efficient compared to the brute force approach. The detailed procedures for solving the scheduling problem are summarized in Algorithm 1. Obviously, Algorithm 1 can find the globally optimal solution for the scheduling problem (TSP).

We summarize the proposed online mechanism integrating allocation rule and payment design in Algorithm 2.

C. Performance Analysis

In this section, we will theoretically analyze our proposed online mechanism in terms of competitive ratio, feasibility of primal and dual solutions, CE, IC, and IR.

Algorithm 1 Online Auction for Scheduling Problem

```
Input: $s_i$, $\Delta t$, $\hat{p}_t$, $\hat{p}^k_{j,t}$, $t_i$, $\tau_i$, and $\gamma_{i,j}$
Output: Optimal schedule $l_{i,j}(\ell)$ and minimum $\beta_{i,j}$ for
        requester $i$ offloading task to requester $j$ or BS.
Initialization;
$l^1_{i,j}(\ell) = \emptyset, l^2_{i,j}(\ell) = \emptyset$, and $\beta_{i,j} = +\infty$;
while $\bar{t}_{i,j} \in [t_i, t_i + \tau_i]$ do
    $t^* = \arg\min_{t \in [t_i, \bar{t}_{i,j}]} h(t)$ and get $\beta^1_{i,j}$ ▷ Solve (SubP1);
    Apply the above strategies for (SubP2) in $[\bar{t}_{i,j}, t_i + \tau_i]$
    and get $\beta^2_{i,j}$ as well as $\pi^*$;
    if $\beta_{i,j} > \beta^1_{i,j} + \beta^2_{i,j}$ then
        $\beta_{i,j} = \beta^1_{i,j} + \beta^2_{i,j}$;
        $l^1_{i,j}(\ell) = l^1_{i,j}(\ell) \cup t^*$ and $l^2_{i,j}(\ell) = l^2_{i,j}(\ell) \cup \pi^*$;
        $l_{i,j}(\ell) = l^1_{i,j}(\ell) \cup l^2_{i,j}(\ell)$;
    Move $\bar{t}_{i,j}$ to the next time slot in $[t_i, t_i + \tau_i]$;
return $l_{i,j}(\ell)$ and $\beta_{i,j}$;
```

Algorithm 2 Online Auction for Collaborative Task Offloading in MEC

```
Input: $w_{i,j}$, $s_i$, $\Delta t$, $\hat{p}_t$, $\hat{p}^k_{j,t}$, $t_i$, and $\tau_i$
Output: Optimal schedule $l_{i,j}(\ell^*)$, $j^*$ and payment $p_{i,j}$.
Initialization;
$x^\ell_{i,j} = 0, \forall i \in \mathcal{U}, \forall j \in \mathcal{N}_i, \ell \in \mathcal{L}_{i,j}$;
$u_i = 0$;      ▷the utility of requester $i$;
$j^* = \emptyset$ and $l_{i,j}(\ell^*) = \emptyset$;
while the arrival of requester i's task do
    for $j \in \mathcal{N}_i$ do
        Run Algorithm 1 to get the best scheduling scheme
        $l_{i,j}(\ell)$ and minimum $\beta_{i,j}$;
        if $w_{i,j} - \beta_{i,j} > u_i$ then
            $u_i = w_{i,j} - \beta_{i,j}$;
            $j^* = j$;
            $l_{i,j}(\ell^*) = l_{i,j}(\ell)$;
    if $u_i > 0$ then
        Accept requester $i$ and set $x^{\ell^*}_{i,j^*} = 1$;
        Allocate the collaborator or BS and implement
        schedule scheme according to $j^*$ and $l_{i,j}(\ell^*)$;
        Charge requester $i$ at price $p_{i,j}$;
        Update $\hat{p}^k_{j,t}$ and $\hat{p}_t$ based on (12) and (13);
    else
        Reject requester $i$ and set $x^\ell_{i,j} = 0$ and $p_{i,j} = 0$;
```

Lemma 3: The competitive ratio of our proposed online mechanism is 3.

Proof: Assume that the requester i offloads its task to task executor j, and we define $\Delta P^{(i)}$ and $\Delta D^{(i)}$ as the increment of objective values in primal and its dual problems after requester i has been served, respectively. Then, we have

$$\begin{aligned}\Delta D^{(i)} &= u_i + \sum_{t\in\mathcal{T}} W\Delta\hat{p}_t + \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} R^k_j \Delta\hat{p}^k_{j,t}\\ &= w_{i,j} - \sum_{a_{i,j}\le t\le d_{i,j}} \phi_{i,j}\hat{p}_t - \sum_{a_{i,j}\le t\le d_{i,j}}\sum_{k\in\mathcal{K}} r^k_{i,j}\hat{p}^k_{j,t} + \sum_{t\in\mathcal{T}} W\Delta\hat{p}_t\\ &\qquad + \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} R^k_j\Delta\hat{p}^k_{j,t}\\ &= w_{i,j} - \sum_{a_{i,j}\le t\le d_{i,j}} \phi_{i,j}\hat{p}_t - \sum_{a_{i,j}\le t\le d_{i,j}}\sum_{k\in\mathcal{K}} r^k_{i,j}\hat{p}^k_{j,t}\\ &\quad + \sum_{a_{i,j}\le t\le d_{i,j}} (\phi_{i,j}\hat{p}_t + \frac{\phi_{i,j}}{\Phi_{i,j}}) + \sum_{a_{i,j}\le t\le d_{i,j}}\sum_{k\in\mathcal{K}} (r^k_{i,j}\hat{p}^k_{j,t} + \frac{r^k_{i,j}}{\Gamma_{i,j}})\\ &= w_{i,j} + \sum_{a_{i,j}\le t\le d_{i,j}} \frac{\phi_{i,j}}{\Phi_{i,j}} + \sum_{a_{i,j}\le t\le d_{i,j}}\sum_{k\in\mathcal{K}} \frac{r^k_{i,j}}{\Gamma_{i,j}}\\ &\le 3w_{i,j} = 3\Delta P^{(i)}.\end{aligned}$$

Let $\mathcal{U}^*$ be the set of the offloaded requesters, and $\overline{P}$ and $\overline{D}$ be solutions of primal and its dual problems by our online mechanism, respectively. Then, we must have

$$\overline{P} = \sum_{i\in\mathcal{U}^*} \Delta P^{(i)} = 3\sum_{i\in\mathcal{U}^*} \Delta D^{(i)} = 3\overline{D}.$$

From the linear dual theory, we have

$$\frac{P^*}{\overline{P}} \le \frac{\overline{D}}{\overline{P}} = 3,$$

where P^* is the optimal solution of primal problem (EQMSW). This completes the proof. ■

Lemma 4: Our proposed online mechanism produces almost feasible solutions to offline problem (EQMSW) if $W \gg 1$; $R^k_j \gg 1, \forall j$; $r^k_{i,j} \ll R^k_j$, and $\phi_{i,j} \ll W, \forall i, j$.

Proof: Let Γ_{max}, Φ_{max}, and w_{max} be the maximum values of $\Gamma_{i,j}$, $\Phi_{i,j}$, and $w_{i,j}$, respectively, and r_{min} and ϕ_{min} be the minimum values of $r^k_{i,j}$ and $\phi_{i,j}$, respectively.

We first show that $\hat{p}^k_{j,t}$ can be bounded by the following expression:

$$\hat{p}^k_{j,t} \ge \frac{(1+\frac{1}{R^k_j})^{\sum_{i\in\mathcal{U}'}\sum_{\ell:t\in l^2_{i,j}(\ell)\in l^2_{i,j}} r^k_{i,j}x^\ell_{i,j}} - 1}{\Gamma_{max}}, \tag{16}$$

where $\mathcal{U}^{'}$ denotes the set of all accepted requesters before requester i. We prove the above inequality through mathematical deduction. Define $\hat{p}^k_{j,t}(i)$ as the value of $\hat{p}^k_{j,t}$ before the arrival of requester i. At beginning, we have $x^\ell_{i,j} = 0 \,\forall j, \ell$ and $\hat{p}^k_{j,t}(1) = 0$, so that inequality (16) holds. We then consider the following two cases:

- Case 1: Requester i is rejected by the BS. In this case, we have $x^\ell_{i,j} = 0$ and $p(i+1) = p(i)$. Obviously, the inequality (16) still holds, which does not affect the validation of $p^k_{j,t}(i+1)$.
- Case 2: Requester i is accepted by the BS. In this case, we have

$$\begin{aligned}\hat{p}^k_{j,t}(i+1) &= \hat{p}^k_{j,t}(i)(1 + \frac{r^k_{i,j}}{R^k_j}) + \frac{r^k_{i,j}}{\Gamma_{i,j}R^k_j}\\ &\ge \hat{p}^k_{j,t}(i)(1 + \frac{r^k_{i,j}}{R^k_j}) + \frac{r^k_{i,j}}{\Gamma_{max}R^k_j}\\ &\ge \frac{(1+\frac{1}{R^k_j})^{\sum_{i\in\mathcal{U}'}\sum_{\ell:t\in l^2_{i,j}(\ell)\in l^2_{i,j}} r^k_{i,j}x^\ell_{i,j}} - 1}{\Gamma_{max}}(1+\frac{r^k_{i,j}}{R^k_j}) + \frac{r^k_{i,j}}{\Gamma_{max}R^k_j}\end{aligned}$$

$$= \frac{(1+\frac{1}{R_j^k})^{\sum\limits_{i\in\mathcal{U}'}\sum\limits_{\ell:t\in l_{i,j}^2(\ell)\in l_{i,j}^2} r_{i,j}^k x_{i,j}^\ell}(1+\frac{r_{i,j}^k}{R_j^k})}{\Gamma_{max}} - \frac{1}{\Gamma_{max}}$$

$$\approx \frac{(1+\frac{1}{R_j^k})^{\sum\limits_{i\in\mathcal{U}'}\sum\limits_{\ell:t\in l_{i,j}^2(\ell)\in l_{i,j}^2} r_{i,j}^k x_{i,j}^\ell}(1+\frac{1}{R_j^k})^{r_{i,j}^k}}{\Gamma_{max}}$$

$$= \frac{(1+\frac{1}{R_j^k})^{\sum\limits_{i\in\mathcal{U}''}\sum\limits_{\ell:t\in l_{i,j}^2(\ell)\in l_{i,j}^2} r_{i,j}^k x_{i,j}^\ell}}{\Gamma_{max}}, \tag{17}$$

where $\mathcal{U}^{''} = \mathcal{U}^{'} \cup i$ and the approximation holds because $R_j^k \gg 1$ and $r_{i,j}^k \ll R_j^k$, and $(1+a)^x \approx 1 + ax$ when a and x are small enough.

Therefore, inequality (16) holds no matter whether requester i is accepted or not. However, $\hat{p}_{j,t}^k(+\infty) < \frac{w_{max}}{r_{min}}(1+1) + 1 = 2\frac{w_{max}}{r_{min}} + 1$ because of the conditions $w_{i,j} > \beta_{i,j}$ and $r_{i,j}^k \ll R_j^k$. By reconsidering the inequality (16), we have

$$\frac{\sum\limits_{i\in\mathcal{U}'}\sum\limits_{\ell:t\in l_{i,j}^2(\ell)\in l_{i,j}^2} r_{i,j}^k x_{i,j}^\ell}{R_j^k} \le \frac{\log(\Gamma_{max}(2\frac{w_{max}}{r_{min}}+1)+1)}{R_j^k \log(1+\frac{1}{R_j^k})}$$
$$\approx \log(\Gamma_{max}(2\frac{w_{max}}{r_{min}}+1)+1), \tag{18}$$

where the last approximation holds when $R_j^k \gg 1$. Inequality (18) indicates that the constraint C_{10} in problem (EQMSW) may be violated by at most $\log(\Gamma_{max}(2\frac{w_{max}}{r_{min}}+1)+1)$.

To verify that the solution meets constraint C_{13} in problem (EQMSW), we can follow the similar procedure to demonstrate that before the arrival of requester i, the value of $\hat{p}_t$ can be bounded as

$$\hat{p}_t(i) \ge \frac{(1+\frac{1}{W})^{\sum\limits_{i\in\mathcal{U}'}\sum\limits_{j\in\mathcal{N}_i}\sum\limits_{\ell:t\in l_{i,j}^1(\ell)\in l_{i,j}^1}\phi_{i,j}x_{i,j}^\ell} - 1}{\Phi_{max}}. \tag{19}$$

However, we have $\hat{p}_t(+\infty) < \frac{w_{max}}{\phi_{min}}(1+1)+1 = 2\frac{w_{max}}{\phi_{min}}+1$ because of the conditions $w_{i,j} > \beta_{i,j}$ and $\phi_{i,j} \ll W$. Combing those inequalities, we have

$$\frac{\sum\limits_{i\in\mathcal{U}'}\sum\limits_{j\in\mathcal{N}_i}\sum\limits_{\ell:t\in l_{i,j}^1(\ell)\in l_{i,j}^1}\phi_{i,j}x_{i,j}^\ell}{W} \le \frac{\log(\Phi_{max}(2\frac{w_{max}}{\phi_{min}}+1)+1)}{W\log(1+\frac{1}{W})}$$
$$\approx \log(\Phi_{max}(2\frac{w_{max}}{\phi_{min}}+1)+1), \tag{20}$$

where the last approximation holds when $W \gg 1$. It indicates that the constraint C_6 in problem (EQMSW) may be violated by at most $\log(\Phi_{max}(2\frac{w_{max}}{\phi_{min}}+1)+1)$. This completes the proof. ■

Lemma 5: Our proposed online mechanism produces a feasible solution to dual problem (EQDP).

Proof: We consider the following two cases.

- Case 1: Requester i is rejected, which means $w_{i,j^*} - \beta_{i,j^*} \le 0$ for the best selected task executor j^* and $u_i = 0$ according to the acceptance condition (11). Thus, constraint C_{15} holds in this case.
- Case 2: Requester i is accepted, which means $u_i = w_{i,j^*} - \beta_{i,j^*} > 0$ for the best selected task executor j^*. Thus, constraint C_{15} still holds in this case.

Therefore, constraint C_{15} in problem (EQDP) always holds no matter whether requester i is accepted or not. This completes the proof. ■

Lemma 6: Our proposed online mechanism runs in polynomial time to get the result.

Proof: The computational complexity of the proposed online mechanism is evaluated in terms of computation times with respect to the number of requesters and collaborators. Recall that our proposed online mechanism consists of Algorithm 2 and Algorithm 1. For Algorithm 1, given that there are total M time slots during the period of $[t_i, t_i + \tau_i]$, the computational complexity of Algorithm 1 can be calculated as $O(M(M - M + 1 + \frac{M(M-1)}{2} + M - 1) = O(M^2\frac{(M+1)}{2})$. Therefore, the computational complexity of Algorithm 2 is $O(|\mathcal{U}| \times |\mathcal{N}_{max}| \times M^2\frac{(M+1)}{2})$. Note that this is the worst case computational complexity. Obviously, Algorithm 2 runs in polynomial time, which completes the proof. ■

Lemma 7: The proposed online mechanism can guarantee truthfulness (IC) and individual rationality (IR).

Proof: We first prove the truthfulness in the requesters' biding values. Note that the marginal prices $\hat{p}_{j,t}^k$ and $\hat{p}_t$ depend only on the past accepted requesters and are independent on the biding values of current requester i. Furthermore, the proposed online mechanism always assigns the requested resource to that requester only when the utility of that requester is maximized among all its bidding values and greater than zero given the current marginal prices. Therefore, our mechanism can be treated as a sequential posted price mechanism [36] or iterative auction [37], where the auctioneer posts the price and the bidders choose the best bidding values to maximize their utilities. In this way, the bidders cannot gain more utilities by misreporting their biding values.

Next, we demonstrate the truthfulness in arrival time t_i. If a requester reports the arrival time $t_i^{'}$ earlier than the actual value (i.e., $t_i^{'} < t_i$), this requester cannot increase its utility or even suffers from the failure to complete its task when $t_i^{'} < t_{i-1}$ or the transmission time is scheduled within the period of $[t_i^{'}, t_i]$. When the requester declares its arrival time later than t_i (i.e., $t_i^{'} > t_i$), the mechanism will find the optimal transmission and computation times after $t_i^{'}$ while in fact, such optimal times may happen in $[t_i, t_i^{'}]$, which results in an increased payment and a decreased utility. Thus, the requesters won't misreport their arrival time.

Third, it is obvious that the requesters won't intend to misreport their offloaded tasks (i.e., T_i) due to the fact that this can incur the failure completion of their tasks.

Finally, we verify the individual rationality. According to the acceptance condition (11), a requester can be accepted only if one of its maximum biddings can lead to a positive utility; otherwise, that requester is rejected and its utility is zero. Hence, our auction satisfies individual rationality. This completes the whole proof. ■

Theorem 1: The proposed online mechanism has a competitive ratio of 3, runs polynomially, and guarantees truthfulness and individual rationality.

TABLE II
MAIN SIMULATION PARAMETERS

Parameter	Value
Cell radius	500 m
Total bandwidth	40 MHz
Transmission power at requesters	1.5 W
Background noise average power	-60 dBm
Total running time	30 minutes
Time slot length	1 second
Task size	Randomly from 10 to 30 MB
CPU cycles coefficient	330 cycles/Byte
Energy consumption coefficient	10^{-26}
Unit energy cost	\$0.1
Valuation	Randomly from $[\$0.1, \$10]$
The maximum delay	Randomly from [5, 15] seconds
Computation capacity of BS	10 GHz
Storage capacity of BS	10 GB
Computation capacity of collaborators	2 GHz
Storage capacity of collaborators	5 GB

Proof: By combing **Lemma 3**, **Lemma 6**, and **Lemma 7**, we can get the above conclusion. This completes the proof. ■

V. NUMERICAL RESULTS

In this section, numerical simulations are conducted to verify the effectiveness of our proposed online mechanism. Since the total social welfare, revenue, and utility of requester are the most important economical metrics and the competitive ratio is also a vital metric to measure an online mechanism, in this section, we will focus on evaluating these two performance metrics with respect to different numbers of requesters and collaborators. In the simulation, the wireless channels between requesters and task executors (i.e., collaborators or the BS) experience Rayleigh fading and all the channel coefficients are zero-mean, circularly symmetric complex Gaussian (CSCG) random variables with variances d^{-v}, where d is the distance between the transmitter and the receiver and $v = 4$. Table II lists the main simulation parameters, some of which have also been employed in [22], [17], [38], and [39]. For comparison purpose, the following three online strategies are also simulated as benchmarks.

- Random online mechanism: For each requester, the BS randomly selects the task executor and randomly schedules the transmission and computation times.
- Greedy online mechanism: Upon the arrival of a requester, the BS chooses the task executor with the maximal valuation as the winner and schedules one time slot for transmission and $\lceil \tau \rceil - 1$ time slots for computation.
- First In First Out (FIFO) online mechanism [40]: Arriving tasks are always accepted with a fixed transmission and computation time schedule till the resources are run out.

Fig. 4 shows the total social welfare achieved by different online mechanisms with respect to different numbers of arrived requesters when there are 30 collaborators, i.e., $|\mathcal{M}| = 30$. From this figure, we can see that the achievable total social welfare increases with the number of requesters. This trend is obvious since with the arrival of more requesters, the BS will admit more before resources are exhausted which results in the increment on the total social welfare. It is worthy noting that our proposed online mechanism outperforms both the random and FIFO online mechanisms, but underperforms the greedy one. This is because the proposed online mechanism tries to minimize the scheduling problem (TSP) so as to maximize the utility of each requester, while the greedy one only attempts to maximize the total social welfare and ignores the maximality of the individual utility. In addition, according to [41], the simple greedy online mechanism cannot guarantee the truthfulness and individual rationality properties.

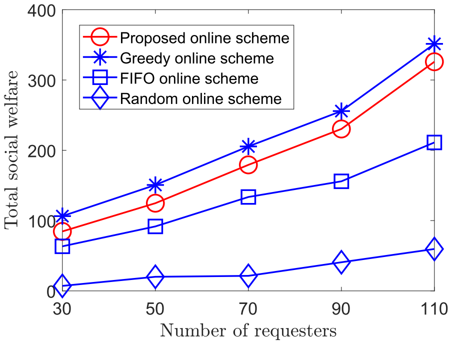

Fig. 4. TSW versus numbers of requesters.

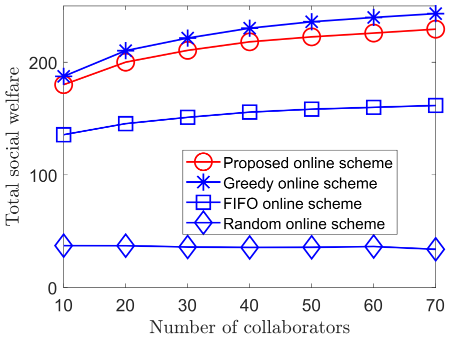

Fig. 5. TSW versus numbers of collaborators.

Fig. 5 reevaluates the total social welfare under various numbers of collaborators. In the simulation, the total number of arrived requesters is fixed at 75, i.e., $|\mathcal{U}| = 75$. It can be seen from the figure that the total social welfare increases with the number of collaborators till reaching a saturation when the number of collaborators is large enough (e.g., 55 in our simulation). This is because with the excessive amount of collaborators, each requester is always served by its most effective collaborator while other collaborators have no effects on the achievable total social welfare. In addition, the total social welfare of random online mechanism is almost a constant. The reason is that this mechanism selects the collaborator in random so that it treats all collaborators equally regardless of how many collaborators exist. Similar to Fig. 4, our proposed online mechanism outperforms both the random and FIFO online mechanisms, but is still inferior to the greedy one. In summary, from both Figs. 4 and 5, we can conclude that although it is not difficult to design an online algorithm with a sound competitive ratio (or larger social welfare), it does be hard to devise an online mechanism which possesses sound competitive ratio, truthfulness, and individual rationality at the

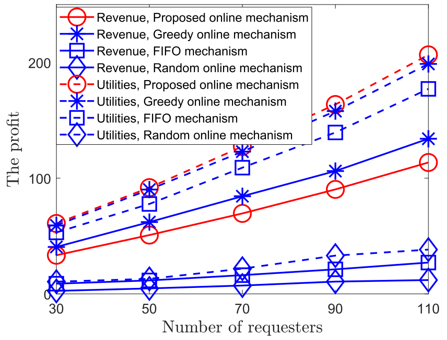

Fig. 6. Profits versus numbers of requesters.

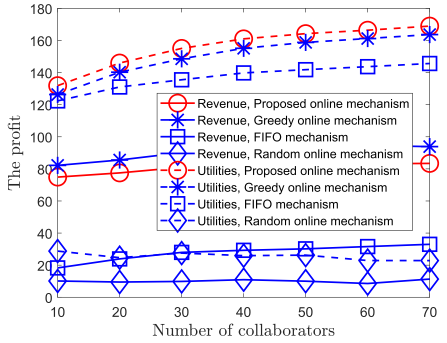

Fig. 7. Profits versus numbers of collaborators.

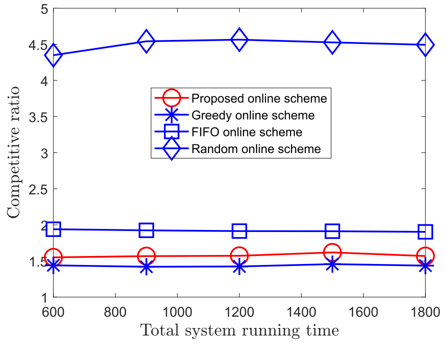

Fig. 8. CR versus total system running time.

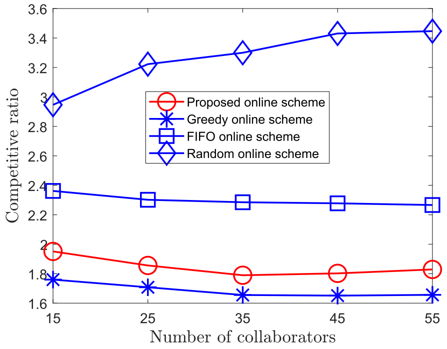

Fig. 9. CR versus numbers of collaborators.

same time. In fact, our proposed online mechanism scarifies a little bit of competitive ratio to achieve other economical properties.

Fig. 6 shows relationship between revenues or utilities and the number of requesters under different online mechanisms. All other simulation parameters are set the same as those in Fig. 4. Revenues are calculated by summing the substraction of each task executor's payment and its cost while utilities are the summation of all requesters. From this figure, we can observe that both revenues and utilities increase with the number of requesters. This is because more requesters can be accepted, which can gain more benefits. Note that since the payment by the greedy online mechanism is larger than the proposed one, the revenue of greedy online mechanism must be higher than that of the proposed one. But, utilities of the proposed online mechanism are larger than all other mechanisms. Moreover, both the differences of revenues and utilities between proposed and greedy online mechanisms are gradually increasing when the number of requesters increases. The reason behind this can be explained as follows. When the number of requesters are small, there are no obvious difference between the proposed and greedy online mechanisms in terms of scheduling scheme due to the fact that since the number of requesters are small enough compared to the system total time slots, the marginal prices at each time slot are small enough and have little effects on total payment. However, such effects increase with the increase numbers of requesters.

Fig. 7 illustrates profits of different mechanisms in terms of both revenues and utilities with respect to the number of collaborators when $|\mathcal{U}| = 75$. It can be observed that profits of the proposed, greedy, and FIFO mechanisms increase first and then keep stable when the number of collaborators becomes large enough, while the profit of the random online mechanism is a constant. The reasons are the same as those to explain the trend in Fig. 5. Moreover, even though the revenue of greedy online mechanism is larger than that of the proposed one, the utilities of requesters are lower than the proposed online mechanism. This further verifies the effectiveness of our proposed online mechanism and the conclusions we have drawn from Fig. 4.

Fig. 8 presents the comparison among different online mechanisms in terms of the competitive ratio by varying the system running time when the number of collaborators is 25. Note that the optimal offline solution is obtained by *Yalmip* optimizer and the payments are based on the *VCG* mechanism [42]. From Fig. 6, we can observe that the CR of our proposed online mechanism is less than 3, which matches our theoretical analyses. Besides, the competitive ratio almost stays unchange with different system running times, which demonstrates that the proposed online truthful mechanism is stable.

Fig. 9 evaluates the performance of different online mechanisms in terms of competitive ratio with respective to different numbers of collaborators when the number of requesters is 40. For the proposed, FIFO, and greedy online mechanisms, their competitive ratios keep less than 3 and slightly decrease till tending to be stable when the number of collaborators becomes large enough. It is because more collaborators can increase the available resources in the system and increase the number of potential alternative collaborators around requesters.

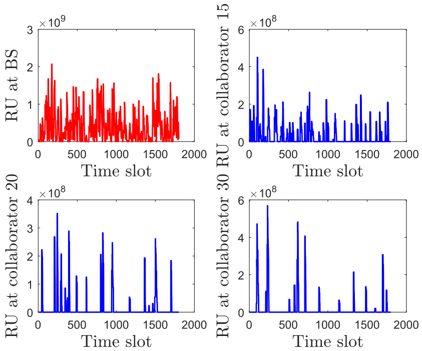

Fig. 10. Utilized computation resource by proposed mechanism.

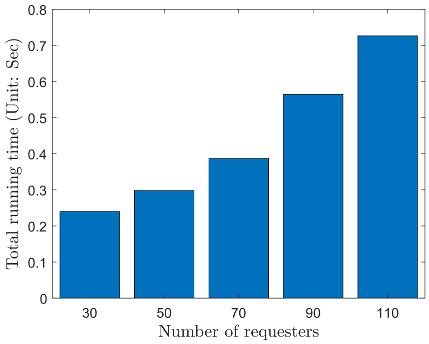

Fig. 11. Time consumption of the whole system.

In contrast, the competitive ratio of the random online mechanism increases (i.e., the worse performance) with the number of collaborators. According to Fig. 5, as the increase of $|\mathcal{M}|$, the social welfare of random online mechanism stays unchanged, while the offline optimal solution increases because more resources are available for allocation. Thus, the competitive ratio of random online mechanism increases.

Fig. 10 depicts the utilization of computation resource of the proposed online mechanism at the BS and some of collaborators with the evolution of time. In our simulation, we set $|\mathcal{N}| = 150$ and $|\mathcal{M}| = 50$. Since there are a lot of collaborators, we randomly choose three collaborators and the BS to observe their computation resource utilizations. As shown in this figure, the maximal computation resources at the BS, collaborators 15, 20, and 30 are around 2.3 GHz, 480 MHz, 380 MHz, and 580 MHz, accordingly. Obviously, those values are less than their provided computation resources, which demonstrates feasibility of our solution.

Fig. 11 reveals the time consumption of the whole system with the number of requesters when there are 30 collaborators, i.e., $|\mathcal{M}| = 30$. Obviously, it is intuitive and reasonable that the total running time for the whole system increases with the number of requesters. What's more, we can observe that the execution time for single task is roughly 5 millisecond, and the total running time for all tasks is only a fraction of second, which further demonstrates the computational efficiency of our proposed online incentive mechanism.

VI. Conclusion

In this paper, online truthful incentive mechanism design for collaborative task offloading in MEC has been studied. By considering each task's specific requirements in terms of data size, delay, and preference, a social-welfare-maximization problem is formulated. After that, an effective online mechanism is developed based on the primal-dual framework to properly select task executors, suitably schedule transmission and computation times, and optimally allocate the transmission and computation resources. Both theoretical and numerical results show that our proposed mechanism can guarantee feasibility, truthfulness, and computational efficiency with competitive ratio of 3.

References

[1] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[2] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing: A key technology towards 5G," Eur. Telecommun. Standards Inst., Sophia Antipolis, France, White Paper 11, 2015.

[3] 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.

[4] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE Trans. Commun.*, vol. 18, no. 3, pp. 1750–1763, Feb. 2019.

[5] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for mobile edge computing," in *Proc. IEEE WiOpt*, May 2018, pp. 1–6.

[6] Y. Cui *et al.*, "Software defined cooperative offloading for mobile cloudlets," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1746–1760, Jun. 2017.

[7] N. Ti and L. Le, "Computation offloading leveraging computing resources from edge cloud and mobile peers," in *Proc. IEEE ICC*, Paris, France, May 2017, pp. 1–6.

[8] Y. Zhang, L. Song, W. Saad, Z. Dawy, and Z. Han, "Contract-based incentive mechanisms for device-to-device communications in cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2144–2155, Oct. 2015.

[9] T. Wang, Y. Sun, L. Song, and Z. Han, "Social data offloading in D2D-enhanced cellular networks by network formation games," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 7004–7015, Dec. 2015.

[10] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6398–6409, Jul. 2018.

[11] Z. Lu, X. Sun, and T. La Porta, "Cooperative data offload in opportunistic networks: From mobile devices to infrastructure," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3382–3395, Dec. 2017.

[12] G. Li and J. Cai, "An online mechanism for crowdsensing with uncertain task arriving," in *Proc. IEEE ICC*, Kansas, MO, USA, May 2018, pp. 1–6.

[13] W. Gong, B. Zhang, and C. Li, "Location-based online task assignment and path planning for mobile crowdsensing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1772–1783, Feb. 2019.

[14] D. Zhang *et al.*, "Near-optimal and truthful online auction for computation offloading in green edge-computing systems," *IEEE Trans. Mobile Comput.*, to be published.

[15] D. Zhao, X.-Y. Li, and H. Ma, "Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 647–661, Apr. 2016.

[16] L. Yang, B. Liu, and J. Cao, "Joint computation partitioning and resource allocation for latency sensitive applications in mobile edge clouds," in *Proc. IEEE Int. Conf. Cloud. Comput.*, Honolulu, CA, USA, Jun. 2017, pp. 246–254.

[17] T. Thinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[18] S. Sardellitti, S. Barbarossa, and G. Scutari, "Distributed mobile cloud computing: Joint optimization of radio and computational resources," in *Proc. IEEE Globecom Workshops*, Austin, TX, USA, Dec. 2014, pp. 1505–1510.

[19] M.-H. Chen, M. Dong, and B. Liang, "Joint offloading decision and resource allocation for mobile cloud with computing access point," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Pudong, China, Mar. 2016, pp. 3516–3520.

[20] Y. H. Yu, J. Zhang, and K. B. Letaief, "Joint subcarrier and CPU time allocation for mobile edge computing," in *Proc. IEEE Globecom*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[21] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2082–2091, Dec. 2017.

[22] H. Zhang, F. Guo, H. Ji, and C. Zhu, "Combinational auction-based service provider selection in mobile edge computing networks," *IEEE Access*, vol. 5, pp. 13455–13464, Jul. 2017.

[23] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 1975–1989, Sep. 2019.

[24] A.-L. Jin, W. Song, and W. Zhuang, "Auction-based resource allocation forsharing cloudlets in mobile cloud computing," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1, pp. 45–57, Jan./Mar. 2015.

[25] M. Liwang, S. Dai, Z. Gao, Y. Tang, and H. Dai, "A truthful reverse-auction mechanism for computation offloading in cloud-enabled vehicular network," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4214–4227, Jun. 2019.

[26] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11339–11351, Dec. 2017.

[27] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[28] B. Fan, H. Tian, L. Jiang, and A. V. Vasilakos, "A social-aware virtual MAC protocol for energy-efficient D2D communications underlying heterogeneous cellular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8372–8385, Sep. 2018.

[29] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, and K. Hugl, "Device-todevice communication as an underlay to LTE-advanced networks," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 42–49, Dec. 2009.

[30] 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.

[31] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

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

[33] J. Borghoff, L. R. Knudsen, and M. Stolpe, "Bivium as a mixed-integer linear programming problem," in *Proc. IMA Int. Conf. Cryptogr. Coding*, in Lecture Notes in Computer Science, vol. 5921. Berlin, Germany: Springer, 2009, pp. 133–152.

[34] N. Buchbinder and J. S. Naor, "The design of competitive online algorithms via a primal–dual approach," *Found. Trends Theor. Comput. Sci.*, vol. 3, nos. 2–3, pp. 263–293, 2009.

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[36] S. Chawla, J. Hartline, D. Malec, and B. Sivan, "Multi-parameter mechanism design and sequential posted pricing," in *Proc. ACM STOC*, 2010, pp. 311–320.

[37] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "A double-auction mechanism for mobile data-offloading markets," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1634–1647, Oct. 2015.

[38] M. H. Chen, B. Liang, and M. Dong, "Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point," in *Proc. IEEE INFOCOM*, May 2017, pp. 1–9.

[39] F. Guo, L. Ma, H. Zhang, H. Ji, and X. Li, "Joint load management and resource allocation in the energy harvesting powered small cell networks with mobile edge computing," in *Proc. IEEE INFOCOM WKSHPS*, Apr. 2018, pp. 299–304.

[40] D. E. Irwin, L. E. Grit, and J. S. Chase, "Balancing risk and reward in a market-based task service," in *Proc. IEEE HPDC*, Jun. 2004, pp. 160–169.

[41] R. Lavi and N. Nisan, "Competitive analysis of incentive compatible on-line auctions," *Theor. Comput. Sci.*, vol. 310, nos. 1–3, pp. 159–180, Jan. 2004.

[42] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *J. Finance*, vol. 16, no. 1, pp. 8–37, 1961.

Gang Li received the B.E. degree in communication engineering from the Wuhan Institute of Technology, Wuhan, China, in 2013, and the M.S. degree in information and communication systems from the Guilin University of Electronic Technology, Guilin, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada. His current research interests include mobile edge computing, cooperative communications, online algorithm, mechanism design, and machine learning. He was a recipient of the Concordia International Tuition Award of Excellence from 2019 to 2020.

Jun Cai received the Ph.D. degree in electrical engineering from the University of Waterloo, ON, Canada, in 2004. From June 2004 to April 2006, he was with McMaster University, Canada, as a Natural Sciences and Engineering Research Council of Canada (NSERC) Post-Doctoral Fellow. From July 2006 to December 2018, he was with the Department of Electrical and Computer Engineering, University of Manitoba, Canada, where he was a Full Professor and a NSERC Industrial Research Chair. Since January 2019, he has been with the Department of Electrical and Computer Engineering, Concordia University, Canada, as a Full Professor and the PERFORM Centre Research Chair. His current research interests include edge/fog computing, ehealth, radio resource management in wireless communications networks, and performance analysis. He was a recipient of the Best Paper Award from CHINACOM in 2013, the Rh Award for outstanding contributions to research in applied sciences in 2012 from the University of Manitoba, and the Outstanding Service Award from the IEEE GLOBECOM 2010. He served as the Technical Program Committee (TPC) Co-Chair for IEEE GREENCOM 2018; the Track/Symposium TPC Co-Chair for the IEEE VTC-Fall 2019, IEEE CCECE 2017, IEEE VTC-Fall 2012, IEEE GLOBECOM 2010, and IWCMC 2008; the Publicity Co-Chair for IWCMC 2010, 2011, 2013, 2014, 2015, 2017, and 2020; and the Registration Chair for QShine 2005. He also served on the Editorial Board of *IET Communications and Wireless Communications* and *Mobile Computing*.