Energy-Efficient Admission of Delay-Sensitive Tasks for Mobile Edge Computing

Xinchen Lyu, Hui Tian, Wei Ni, Yan Zhang, Ping Zhang, and Ren Ping Liu

***Abstract*—Task admission is critical to delay-sensitive applications in mobile edge computing, but technically challenging due to its combinatorial mixed nature and consequently limited scalability. We propose an asymptotically optimal task admission approach which is able to guarantee task delays and achieve $(1-\epsilon)$-approximation of the computationally prohibitive maximum energy saving at a time-complexity linearly scaling with devices. ϵ is linear to the quantization interval of energy. The key idea is to transform the mixed integer programming of task admission to an integer programming (IP) problem with the optimal substructure by pre-admitting resource-restrained devices. Another important aspect is a new quantized dynamic programming algorithm which we develop to exploit the optimal substructure and solve the IP. The quantization interval of energy is optimized to achieve an $[\mathcal{O}(\epsilon), \mathcal{O}(1/\epsilon)]$-tradeoff between the optimality loss and time-complexity of the algorithm. Simulations show that our approach is able to dramatically enhance the scalability of task admission at a marginal cost of extra energy, as compared to the optimal branch and bound method, and can be efficiently implemented for online programming.**

***Index Terms*—Mobile edge computing, Resource allocation, Task admission, Optimization methods**

I. Introduction

Computationally demanding mobile applications, such as face recognition, language processing, online gaming, and eHealth, have been fast developing and increasingly outgrowing the limited capabilities of devices [1]. Offloading and processing these computations at the edge of wireless access networks, e.g., at base stations or gateways, mobile edge computing (MEC) can bridge the gap between the capability limitation of devices and their ever-increasing demands for computations [2], [3]. Co-located at macro/pico/femto base stations, MEC servers are able to conduct and deliver computation services promptly, hence reducing latency and energy consumption which are the key challenges to future wireless networks [4]. Typical characteristics of MEC include low latency, proximity, high bandwidth, mobility support and location awareness [5].

Task admission and resource allocation are critical to MEC, especially in the presence of a large number of delay-sensitive tasks, e.g., face recognition for security applications or online gaming, due to the finite physical bandwidths of wireless channels and limited computational resources at MEC servers. In coupling with allocation of both radio/energy resources for task offloading and computational resources for task processing, task admission is typically a non-deterministic polynomial (NP)-hard combinatorial mixed integer programming (MIP) problem [6]. The computational complexity of task admission would exponentially grow against the number of devices, and become prohibitive in the presence of large numbers of devices (or offloading requests). The scalability and practicality of task admission would degrade.

Earlier works on task admission (also known as scheduling) for MEC, such as [7]–[11], assumed independence among different devices in their admission/offloading decisions under unlimited computational capabilities on cloud platforms. More recent researches have been focused on either offloading decision-makings [12]–[16] or resource allocation [17], [18] among multiple devices, rather than jointly accounting for both. For joint optimizations of offloading decisions and resource allocation, delay-tolerant tasks has been typically assumed [6], [19], [20]; or the feasibility of the problem has been assumed [21]. To the best of our knowledge, efficient joint optimization of offloading decisions and resource allocations has yet to be addressed for delay-sensitive tasks, especially in the case where the number of devices is large, the optimization can be infeasible and admission control would be necessary.

This paper proposes an asymptotically optimal online programming of task admission which can guarantee task delays and achieve $(1-\epsilon)$-approximation of the maximum energy saving at a time-complexity of $\mathcal{O}(NK^2/\epsilon)$ linear to the device number N. (K is the number of subchannels. ϵ is linear to the quantization interval of energy.) This is based on our new discovery that, after the tasks that cannot catch deadlines by local execution are pre-admitted for offloading, the admission of the remaining tasks is integer programming (IP) with the optimal substructure. By relaxing the energy saving as a continuous variable, the subproblems under the substructure can recursively produce the optimal admission schedule at a polynomial complexity depending on the number of subproblems. Another critical contribution is that we optimally discretize the energy saving to holistically control the number of the subproblems, leveraging the optimality loss and complexity at an $[\mathcal{O}(\epsilon), \mathcal{O}(1/\epsilon)]$-tradeoff.

The proposed approach is efficient to implement. Only part of the devices need to report their information while the asymptotic optimality is unaffected. Other devices can evaluate their energy savings of offloading against local execution,

This work was supported by the National Natural Science Foundation of China under Grant 61471060. (Corresponding author: Hui Tian.)

X. Lyu is with Beijing University of Posts and Telecommunications, China, and University of Technology Sydney, Australia.

H. Tian and P. Zhang are with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China.

W. Ni is with Commonwealth Scientific and Industrial Research Organization, Australia.

Y. Zhang is with University of Oslo, Norway.

R. P. Liu is with University of Technology Sydney, Australia.

and spontaneously withhold offloading requests if there is no energy saving or the deadlines would be violated. Evident from extensive simulations, the proposed protocol exhibits an attractive property that the signalling overhead can decrease with the increasing number of devices at no cost of optimality, especially when N is large. This is because the number of devices pre-admitted grows, draining the resources and increasingly preventing other devices from offloading.

DP provides an efficient solver for a class of complex problems which can be partitioned into simpler overlapping subproblems with optimal substructure and solved in sequel. However, existing DP algorithms, such as the Floyd-Warshall algorithm and the Bellman-Ford algorithm [22], are unable to solve the problem of interest. This is due to the fact that the original problem of interest is MIP and does not have the optimal substructure. The solution also involves continuous energy saving and computational resources, which prevents the problem from being partitioned finitely and deterministically. Our contributions of restructuring the problem to IP and comply with the optimal substructure, including pre-admission, discretization of continuous variables, and optimization of discretization intervals, are key to solving the problem.

The rest of this paper is organized as follows. The related works are reviewed in Section II, and the system model is presented in Section III. In Section IV, we formulate the MIP problem of task admission and resource allocation, reformulate it as IP, and propose a quantized DP algorithm for the IP problem. In Section V, we design the asymptotically optimal quantization interval, followed by discussions and extensions in Section VI. Simulation results are provided in Section VII. Conclusions are provided in Section VIII.

II. Related Work

Earlier works on the task admission of MEC, such as [7]–[11], were focused on single-device decision-makings under an implicit assumption of unlimited computational capabilities on cloud platforms. In [7], an application was decided to be offloaded entirely to a cloud or executed at a mobile device. In [8]–[11], applications were partitioned into tasks or code blocks to improve efficiency. In [8], a mobile application was partitioned into a sequence of tasks which were processed sequentially. In [9], partitioned tasks were processed in parallel, and DP was employed to minimize the processing delay under energy constraints. In [10], a heuristic online approach was developed to partition tasks. In [11], a directed acyclic graph was used to represent code blocks, and a genetic algorithm was developed to partition the code blocks.

More recent studies have been focused on either offloading decisions or resource allocation among multiple devices [12]–[16], with no allocation of transmission or computational resources. In [12], under limited cloud resources, both online and offline algorithms were developed to partition tasks for multiple devices. In [13], using queueing theory, offloading decisions were formulated as a non-cooperative game in a three-tier MEC architecture consisting of mobile devices, cloudlets and distant cloud. In [14], [15], offloading decisions were studied in a single cell of WLAN or CDMA with intra-cell interference, where competitions among devices for radio resources were modeled as a non-cooperative game. The utilities of individual game players were designed with emphasis on the stability of the games. Only Pareto-optimal solutions could be achieved, if stabilized, which unnecessarily can be translated to the global optimality in terms of the utility of the entire cell. In [16], an online task offloading algorithm was proposed to minimize energy consumption, where Lyapunov optimization was taken to ensure the incentives of user cooperation in fog computing.

There are only a number of works that have jointly optimized the offloading decisions and resource allocation of multiple devices, typically for delay-tolerant services [6], [19], [20]. In [19], both offline and online approaches were proposed for the joint optimization, where a single task was offloaded to the MEC server while the others were executed locally. In [20], the allocations of computational resources and transmission bandwidths were optimized by exploiting semi-definite programming, and the offloading decisions were generated through randomized rounding. In [6], a heuristic scheme based on a submodular optimization method was proposed to jointly optimize offloading decisions and resource allocations for delay-tolerant tasks.

There are even fewer works for delay-sensitive services [17], [18], [21]. In [17] and [18], transmission and computational resources were jointly optimized to save energy, but there was no attempt to optimize offloading decisions. In [17], a distributed optimization framework was proposed using successive convex optimization, and a closed-form expression for the maximum energy saving was derived in a single-user case. In [18], both independent and joint optimizations of computational and transmission resources were formulated to be non-convex optimization problems, which were reformulated and iteratively solved using minimum mean square error criteria. In [21], joint optimization of offloading and resource allocation was considered, but under a relaxed assumption that the problem was feasible. A suboptimal solution was developed by decoupling offloading decisions and resource allocations. Tasks were incrementally offloaded, and resources were correspondingly allocated by exploiting second-order cone programming iteratively, until either the task deadlines were violated or energy saving diminished. In practice, however, the problem can be increasingly infeasible with the growing number of tasks to be offloaded.

Despite the same objective as in [21], our work is distinctively different from [21] by jointly optimizing offloading decisions and resource allocations without the feasibility assumption. It addresses the challenging problem of admission control yet to be addressed in the case that the problem is infeasible. Our work is also distinct by providing a non-heuristic solution with proved asymptotic optimality.

III. System Model

Fig. 1 shows a multi-user MEC system, where an LTE Base Station (BS) is physically co-located with an MEC server[1].

[1] The proposed approach is a generic MEC framework which is not limited to LTE. Nonetheless, LTE provides a widely accepted and approved embodiment of MEC [5].

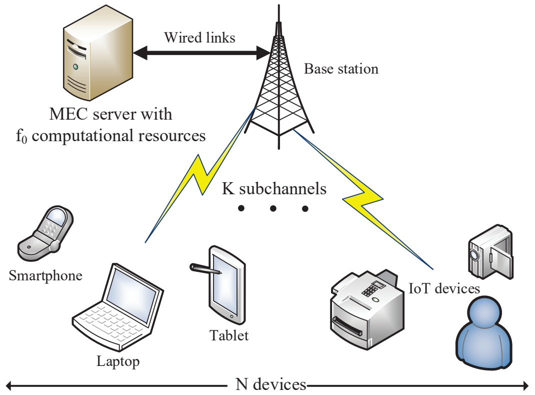


Fig. 1. The multi-user MEC system.

The connection between the MEC server and the BS can be via comparatively delay- and fault-free wired links, such as fiber. With finite computational resources, the MEC server serves the tasks offloaded from a number of devices within the coverage of the BS. The tasks can be processed locally at the device, or offloaded to the MEC server.

The task of device i to be processed can be defined by a triplet $(D_i, C_i, T_i^{\text{req}})$, where D_i specifies the size of the task in bits, C_i is the number of CPU cycles required to accomplish the task, and T_i^{req} is the task deadline, i.e., the maximum delay that the task can tolerate. Assume that every task is atomic, i.e., cannot be further split given strong dependence over different parts of the task [14]–[19]. D_i and C_i, acquired by program profilers, depend on the type of task and are consistent for a specific type [8], [11], [12]. The output of a task is generally much smaller than the task size, and can be returned to the device with negligible transmission delay (e.g., face recognition and language processing [14], [15]).

Let F_i^l denote the local computational capability of device i in cycles per second. T_i^l is the time required to perform the task locally at the device, as given by

$$T_i^l = C_i/F_i^l. \tag{1}$$

The CPU power consumption is widely modeled to be a super-linear function of F_i^l, as given by [14], [15], [23]

$$P_i^l = \alpha(F_i^l)^\gamma, \tag{2}$$

where P_i^l denotes the local power consumption of device i, and α and γ are pre-configured model parameters depending on the chip architecture. Typically, $\alpha = 10^{-11}$ Watt/cycle$^\gamma$ and $2 \leq \gamma \leq 3$ [14], [15], [23].

The energy consumption of device i for *local* computation, denoted by E_i^l, is therefore given by

$$E_i^l = P_i^l \cdot T_i^l = \alpha(F_i^l)^{\gamma-1}C_i. \tag{3}$$

Let f_0 denote the quantity of the available computational resources at the MEC server in cycles per second. In the case that device i is admitted for offloading, the MEC server allocates f_i computational resources, and performs the task on behalf of the device.

With consideration of typical static IoT networks, we assume that the channels are frequency-flat and each device i feeds back its own channel, denoted by h_i, to the BS infrequently. The BS schedules the devices accordingly. The uplink data rate of device i can be given by

$$R_i = W\log_2(1 + p_i h_i/N_0), \tag{4}$$

where W is the channel bandwidth and N_0 is the noise power. With limited number of subchannels, the BS can support at most K concurrent data transmissions at the same time. p_i is the uplink transmission power, and can be pre-configured based on the capability and location of the device, as suggested by 3GPP [24]. Nevertheless, the algorithm developed in this paper can be extended to more complex scenarios, where the channels are frequency-selective and can be selected for different devices to further save energy. However, signalling overhead would grow under this channel-aware scheduling, as all the devices need to feed back instantaneous channels to the BS, as will be discussed in Section VI-C.

For device i, the total delay of remote task processing, denoted by T_i^r, can be given by

$$T_i^r = T_i^t + T_i^e = D_i/R_i + C_i/f_i, \tag{5}$$

where $T_i^t = D_i/R_i$ and $T_i^e = C_i/f_i$ are the uplink transmission time and remote execution time, respectively. Typically, the computational resources allocated to each task at the MEC server, f_i, are up to 10^{10} cycles per second [15]. Taking the example of face recognition, the required processing density is 31680 cycles/bit [25]. On the other hand, the uplink data rate R_i for IoT devices is typically less than 250kbps [24]. The processing speed of about $\frac{10^{10}}{31680} \approx 320$ kbps is similar to the uplink rate. The processing delay at the MEC server is non-negligible, as compared with the transmission delay.

If admitted for remote processing, the energy consumption of device i, denoted by E_i^r, is given by

$$E_i^r = (p_i/\zeta_i)T_i^t = p_i D_i/\zeta_i R_i, \tag{6}$$

where ζ_i is the power amplifier efficiency of device i.

It is the MEC server that makes and advises admission decisions based on the availability of its own resources and the requests of all devices. The request that device i sends consists of information on task and device parameters, such as C_i, D_i, T_i^{req}, p_i, T_i^l and E_i^l.

IV. Energy-efficient Offloading and Resource Optimization

In this section, we propose to minimize the energy consumption of devices under the latency constraints for delay-sensitive tasks. This can be formulated as an MIP problem. Propositions are developed to reformulate the MIP as IP, which can be solved by a new quantized DP algorithm, named Energy-efficient offloading and Resource Optimization Scheme (EROS).

A. Problem Formulation

EROS is designed to minimize the total energy consumption of devices under latency constraints. It may increase energy consumptions at some individual devices which need to give way to others with tighter energy budget or more stringent deadlines, but improve the sustainability of the entire network in the long term. The problem of interest can be formulated as

$$\begin{aligned}\textbf{P:} \min_{\mathbf{s},\mathbf{f}} &\sum_{i\in\mathbf{N}} s_iE_i^r+(1-s_i)E_i^l,\\ \text{s.t.}\ \ &\text{C1: } s_i\in\{0,1\}, \forall i\in\mathbf{N},\\ &\text{C2: } \textstyle\sum_{i\in\mathbf{N}} s_i\le K,\\ &\text{C3: } \textstyle\sum_{i\in\mathbf{N}} f_i\le f_0,\\ &\text{C4: } T_i\le T_i^{\text{req}}\ ,\ \ \forall i\in\mathbf{N},\end{aligned} \tag{7}$$

where $\mathbf{s}$ and $\mathbf{f}$ are the vectors of admission decisions $s_i\in\{0,1\}$, and allocated computational resources f_i, respectively. Particularly, the task of device i is admitted for offloading if $s_i=1$, or rejected otherwise; and the MEC server allocates f_i to the task. $\mathbf{N}$ denotes the set of active devices; T_i and E_i denote the time and the energy consumption for executing the task of device i, respectively, and can be written as

$$T_i \ \ = s_iT_i^r+(1-s_i)T_i^l; \tag{8}$$

$$E_i \ \ = s_iE_i^r+(1-s_i)E_i^l. \tag{9}$$

Here, C1 states that a task can be either executed locally or offloaded for remote processing. C2 specifies the constraint of concurrent offloaded tasks due to limited frequency subchannels. C3 ensures that the total computational resources assigned are no more than the available computational capability. C4 is the latency constraints for task deadlines.

Note that $\mathbf{P}$ is NP-hard MIP [26]. This is because the admission decision $\mathbf{s}$ is binary while the resource allocation decision $\mathbf{f}$ is continuous. The MIP problem is typically solvable, e.g., by using branch and bound method or exhaustive search, but at prohibitive NP time-complexity. To circumvent this impasse, the following two new propositions are first put forth to decouple $\mathbf{f}$ from C3 and C4 and transform $\mathbf{P}$ to an IP problem which exhibits the optimal substructure of DP.

Proposition 1. *Resource-restrained devices with $C_i/f_i^l > T_i^{\text{req}}$ are pre-admitted for offloading to satisfy task deadlines.*

Define $\mathbf{N}_r=\{i\mid T_i^l>T_i^{\text{req}}\}$. Devices in $\mathbf{N}_r$ are too resource-restrained to accomplish tasks on their own before deadline, and hence pre-admitted for remote processing according to Proposition 1. The MEC server allocates the minimum computational resources to meet the deadline, as specified in the following.

Remark 1. *For a resource-restrained device i, the remote computational resources are pre-allocated, as given by*

$$f_i=s_if_i^{\min}=s_iC_i/(T_i^{\text{req}}-D_i/R_i). \tag{10}$$

Proof. For offloaded task, C4 indicates that $s_iT_i^r\le T_i^{\text{req}}$, i.e., $s_i(D_i/R_i+C_i/f_i)\le T_i^{\text{req}}$. Rearranging the inequality, we have

$$f_i\ge s_if_i^{\min}, \tag{11}$$

where $f_i^{\min}=C_i/(T_i^{\text{req}}-D_i/R_i)$ for notational simplicity. Note that the MEC server is designed to allocate the minimum computational resources to the offloaded traffic here, since the energy consumption of device i is independent of f_i. □

In the case that **P** is feasible, both the computational and transmission resources are sufficient to satisfy the deadlines of all devices. Under this assumption, the remaining computational resources and frequency subchannels available at the MEC server are given by

$$\tilde{f}_0=f_0-\textstyle\sum_{i\in\mathbf{N}_r} f_i^{\min}\ge 0;\ \tilde{K}=K-N_r\ge 0. \tag{12}$$

Proposition 2. *If all the devices are capable of accomplishing tasks by their own, i.e., $T_i^l\le T_i^{\text{req}}$, $\forall i\in\mathbf{N}$, constraints C3 and C4 are equivalent to C5: $\sum_{i\in\mathbf{N}} s_if_i^{\min}\le f_0$.*

Proof. Since $T_i^l\le T_i^{\text{req}}$ is satisfied for any device $i\in\mathbf{N}$ by substituting (11) into C3, we can obtain the necessary condition of C3 and C4, as given by

$$f_0\ge\textstyle\sum_{i\in\mathbf{N}} f_i\ge\sum_{i\in\mathbf{N}} s_if_i^{\min}. \tag{13}$$

On the other hand, C5 can also be proved to be the sufficient condition of C3 and C4. If C5 is satisfied, the minimum resources $f_i=s_if_i^{\min}$ are assigned to the task of device i to fulfill C3 and C4. Therefore, we conclude that if $T_i^l\le T_i^{\text{req}}$ for $\forall i\in\mathbf{N}$, C3 and C4 are equivalent to C5. □

Proposition 2 reveals that the MIP problem (7) can be reformulated to an IP problem by replacing C3 and C4 with C5, on the condition that all devices are capable of local task execution. From Proposition 1, resource-restrained devices in $\mathbf{N}_r$ can be pre-admitted for offloading. Proposition 2 holds for the remaining devices, and therefore the remaining problem can be recast as IP.

Let $\mathbf{N}_u=\mathbf{N}\setminus\mathbf{N}_r=\{1,2,\dots,N_u\}$ collect the devices that are capable of accomplishing tasks on their own before deadline, and $\mathbf{s}_u=\{s_i|i\in\mathbf{N}_u\}$ collect the offloading decisions for the devices in $\mathbf{N}_u$. Devices in $\mathbf{N}_u$ satisfy the condition required in Proposition 2. Applying (13), the energy minimization problem under the latency constraints can be reformulated as an IP problem, as given by

$$\begin{aligned}\textbf{P1:} \max_{\mathbf{s}_u} &\sum_{i\in\mathbf{N}_u} s_iE_i^s,\\ \text{s.t.}\ \ &\text{C1: } s_i\in\{0,1\}, \forall i\in\mathbf{N}_u,\\ &\text{C2: } \textstyle\sum_{i\in\mathbf{N}_u} s_i\le\tilde{K},\\ &\text{C5: } \textstyle\sum_{i\in\mathbf{N}_u} s_if_i^{\min}\le\tilde{f}_0,\end{aligned} \tag{14}$$

where the objective of minimizing the total energy is equivalent to maximizing the energy saving $E_i^s=E_i^l-E_i^r$ through remote processing. Analogous to (10), the remote computational resource schedule is $f_i=s_if_i^{\min}$.

In the case that **P** is infeasible, the available transmission or computational resources at the MEC server and devices can by no means satisfy the deadlines of all devices. Some of the resource-restrained devices $i\in\mathbf{N}_r$ that need to offload to meet their deadlines, as specified in Proposition 1, have to be denied for offloading. Then, **P** becomes to select a subset of the resource-restrained devices and maximize energy saving, as given by

$$\begin{aligned}\textbf{P1':} \max_{\mathbf{s}_r} &\sum_{i\in\mathbf{N}_r} s_iE_i^s,\\ \text{s.t.}\ \ &\text{C1: } s_i\in\{0,1\}, \forall i\in\mathbf{N}_r,\\ &\text{C2: } \textstyle\sum_{i\in\mathbf{N}_r} s_i\le K,\\ &\text{C5: } \textstyle\sum_{i\in\mathbf{N}_r} s_if_i^{\min}\le f_0,\end{aligned} \tag{15}$$

where $\mathbf{s}_r = \{s_i | i \in \mathbf{N}_r\}$ denotes the admission decisions of devices in $\mathbf{N}_r$. The computational resources can also be allocated based on Remark 1, i.e., $f_i = s_i f_i^{\min}$.

Both inherited from **P**, **P1** and **P1'** account for the feasible and infeasible scenarios, respectively. Given the same structure and the same goal of selecting the most energy-efficient subset of devices to offload, **P1** and **P1'** can be solved by a unified solver (with the only difference of input parameters) which is to be developed in the rest of this section. For illustration convenience, the solver is described against **P1** in the paper, but can be readily used for **P1'**.

B. Quantized DP Algorithm

We note that **P1** yields the optimal substructure of DP, and can be partitioned into overlapping subproblems. The solution for **P1** can be efficiently constructed from the solutions for its subproblems by using DP techniques. Specifically, the solution for minimizing the energy of the first i devices can be that either device i offloads its task for remote processing or processes the task locally, given the solutions to the subproblems for the first $(i-1)$ devices.

Let $\phi_i(e,l)$ denote the minimum value of the subproblem defined in (16), i.e., the minimum computational resources for the first i devices while saving e units of energy and offloading $l \in \{0,\ldots,\tilde{K}\}$ tasks:

$$\phi_i(e,l) = \min_{\mathbf{s}_u} \left\{ \sum\nolimits_{j=1}^{i} s_j f_j^{\min} \Big| \sum\nolimits_{j=1}^{i} s_j E_j^s = e, \sum\nolimits_{j=1}^{i} s_j = l \right\}. \tag{16}$$

According to Bellman equation [27], $\phi_i(e,l)$ can be solved recursively based on the results of the preceding subproblems $\phi_{i-1}(e,l)$, as given by

$$\phi_i(e,l) = \min\{\phi_{i-1}(e,l), \phi_{i-1}(e-E_i^s, l-1) + f_i^{\min}\}. \tag{17}$$

The solution for $\phi_i(e,l)$ is chosen between local task execution, i.e., $\phi_{i-1}(e,l)$, or task offloading for remote processing, i.e., $\phi_{i-1}(e-E_i^s, l-1) + f_i^{\min}$. The Bellman equation exploits the optimal-substructure property, and reduces the time-complexity by finding the solution from the subproblem of the shortest size [27].

The admission decision for device i in the solution to subproblem $\phi_i(e,l)$, denoted by $s_i(e,l) \in \{0,1\}$, is given by

$$s_i(e,l) = \begin{cases} 1, \text{ if } \phi_i(e,l) = \phi_{i-1}(e-E_i^s, l-1) + f_i^{\min}; \\ 0, \text{ otherwise.} \end{cases} \tag{18}$$

We note that e can take 2^i possible discrete values for the i-th $(i = 1, \cdots, N_u)$ subproblem in (16). It can be computationally prohibitive to enumerate the possible discrete values of e in (18), especially in the case N is large. To eliminate the computationally prohibitive enumeration, we propose to first relax e to be a continuous variable $(0 \leq e \leq \overline{E})$ which can be properly initialized so that the final optimal value of e takes one of the 2^{N_u} possible discrete values. $\overline{E} = \sum_{i \in \mathbf{N}_u} \max(E_i^s, 0)$ is the upper bound of energy saving for **P1**. Specifically, we initialize $\phi_i(e,l) = \infty$ except that $\phi_i(0,0) = 0$. If $e = e'$ does not belong to the 2^i possible values, $\sum_{j=1}^{i} E_j^s \neq e'$, subproblem (16) is inactive, and $\Phi_i(e', l)$ is not updated and remains ∞. The superordinate subproblems of $\phi_i(e', l)$ are ∞, larger than their counterparts of $\phi_i(e", l)$ with $e"$ taking one of the 2^i possible discrete values. e' cannot be part of the final optimal solution.

Algorithm 1 Energy-efficient offloading and Resource Optimization Scheme (EROS)

```
    Pre-admit Resource-restrained Devices
if $T_i^l > T_i^{\text{req}}$ then
    Offload and schedule resources based on Proposition 1
end if
    Quantized Dynamic Programming
Discretize: $e_i^s = q_\delta(E_i^s)$, $\forall i \in \mathbf{N}_u$
Initialize: $\phi_i(e,l) = \infty$ except that $\phi_i(0,0) = 0$
for Each device $i = 1$ to $N_u$ do
    for $l = 0$ to $\tilde{K}$, $e = 0$ to $\widehat{e}$ do
        $\phi_i(e,l) = \min\{\phi_{i-1}(e,l), \phi_{i-1}(e-e_i^s, l-1) + f_i^{\min}\}$
        Record $s_i(e,l)$ by (18)
    end for
end for
Find the optimal solution $e^*$ by (21)
    Backward Induction
Initialize: $e = e^*/\delta$ and $l = l^*$
for $i = N_u$ down to 1 do
    Task admission: $s_i = s_i(e,l)$
    Trace backward: $e = e - s_i E_i^s$, $l = l - s_i$
end for
    Task Admission
Devices offload tasks according to the result of $s_i$
Schedule computational resources: $f_i = s_i f_i^{\min}$
```

Despite the computationally prohibitive enumeration is avoided, the continuous relaxation of $e \in [0, \overline{E}]$ would lead to an infinite number of subproblems. Nevertheless, the number of subproblems is much flexible and controllable, as compared to the rigid enumeration. To control the number of subproblems and improve the tractability of (18), we propose to further discretize the energy saving e, and restrain the number of subproblems to be finite. (We also optimize the quantization interval to holistically leverage the optimality loss and complexity at an $[\mathcal{O}(\epsilon), \mathcal{O}(1/\epsilon)]$-tradeoff, as to be articulated in Sections V and VI-A.) The uniform quantizer of e is given by

$$q_\delta(e) = k, \text{ if } (k-1)\delta < e \leq k\delta, \tag{19}$$

where δ is the quantization interval. The energy saving of each device can also be discretized, and let $e_i^s = q_\delta(E_i^s)$ denote the quantized energy saving of device i. As a result, the upper bound of quantized total energy saving for **P1**, denoted by $\widehat{e}$, can be given by

$$\widehat{e} = \lceil \overline{E}/\delta \rceil + \tilde{K} \tag{20}$$

where $\lceil \overline{E}/\delta \rceil$ corresponds to the quantized upper bound of total energy saving. Note that, according to (19), the proposed quantizer can overestimate the energy saving by no more than δ, i.e., $0 \leq \delta e_i^s - E_i^s < \delta$. Given the constraint of $\tilde{K}$ available subchannels in **P1**, the MEC server cannot admit more than $\tilde{K}$ devices for offloading. Thus, the quantization error due to the discretization of e_i^s cannot exceed $\delta\tilde{K}$.

The number of subproblems $\phi_i(e,l)$ is the product of the numbers of devices, N_u, the number of available subchannels, $\tilde{K}$, and the upper bound of quantized energy saving, $\widehat{e}$. There are $N_u\tilde{K}\widehat{e}$ subproblems in total. After solving these subproblems for the total of N_u devices, the optimal solution can be given by

$$e^* = \delta \max_{l=0,\ldots,\tilde{K}} \{e \mid \phi_{N_u}(e,l) \leq \tilde{f}_0\}, \tag{21}$$

where e^* is the maximum energy saving by the proposed quantized DP algorithm. Let l^* denote the number of offloaded tasks corresponding to e^*.

Backward induction has been widely used to solve DP problems and can determine a sequence of optimal actions by reasoning backwards [28]. It starts by first assessing the last decision in the optimal solution, i.e., $s_{N_u}(e^*/\delta, l^*)$, and then uses the outcome to determine the second-to-last decision. This continues until the optimal decisions are decided for all the devices. The proposed energy-efficient offloading and resource optimization scheme is summarized in Algorithm 1.

V. Asymptotically Optimal Quantization Interval

There is a tradeoff between the time-complexity and the optimality loss of Algorithm 1 due to the quantization of energy; see (19). Particularly, narrowing the quantization interval δ reduces the optimality loss, but increases the time-complexity. In this section, we quantify the tradeoff by inferring the upper and lower bounds of the solution for **P1**, where the linear programming (LP) relaxation of **P1** is carried out, as given by

$$\begin{aligned} \textbf{P2:} \max_{\mathbf{s}_u} & \sum_{i\in\mathbf{N}_u} s_i E_i^s, \\ \text{s.t. } & \text{C2 and C5}, \\ & \text{C6: } s_i \in [0,1], \forall i \in \mathbf{N}_u. \end{aligned} \tag{22}$$

Here, the binary constraint C1 is relaxed to be C6. Clearly, **P2** gives the upper bound for **P1** in terms of energy saving.

The Lagrangian problem of **P2** can be written as

$$L(\lambda,\mu) = \max_{\mathbf{s}_u\in\text{C6}} \sum_{i\in\mathbf{N}_u} s_i E_i^s + \lambda(\tilde{K} - \sum_{i\in\mathbf{N}_u} s_i) + \mu(\tilde{f}_0 - \sum_{i\in\mathbf{N}_u} s_i f_i^{\min}). \tag{23}$$

The Lagrangian problem $L(\lambda,\mu)$ is separable, and can be restructured as

$$L(\lambda,\mu) = \lambda\tilde{K} + \mu\tilde{f}_0 + \sum\nolimits_{i\in\mathbf{N}_u} L_i(\lambda,\mu), \tag{24}$$

where

$$L_i(\lambda,\mu) = \max_{s_i\in[0,1]} s_i(E_i^s - \lambda - \mu f_i^{\min}). \tag{25}$$

As a result, the Lagrangian function can be maximized if and only if $L_i(\lambda,\mu)$ is maximized for all $i = 1, 2, \cdots$. The maximization of $L_i(\lambda,\mu)$ can be efficiently solved as

$$s_i^*(\lambda,\mu) = \begin{cases} 1, \text{ if } \theta(i,\lambda,\mu) > 0; \\ 0, \text{ if } \theta(i,\lambda,\mu) < 0, \end{cases} \tag{26}$$

where $\theta(i,\lambda,\mu) = E_i^s - \lambda - \mu f_i^{\min}$ for notational simplicity.

Strong duality holds in LP problems [29]. By substituting (26) into (23), the dual problem of (22) can be formulated, as given by

$$\min_{\lambda>0,\mu>0} \lambda\tilde{K} + \mu\tilde{f}_0 + \sum\nolimits_{i\in\mathbf{N}_u} s_i^*(\lambda,\mu)(E_i^s - \lambda - \mu f_i^{\min}), \tag{27}$$

where the optimal Lagrangian multipliers λ^* and μ^*, subject to a hyperplane search problem, can be obtained through multi-dimensional search at a linear complexity of $\mathcal{O}(N_u)$ [30].

According to (26) and the optimal Lagrangian multipliers, $\mathbf{N}_u$ can be divided into three subsets: $\mathbf{N}_u^+ = \{i \mid \theta(i,\lambda^*,\mu^*) > 0\}$, $\mathbf{N}_u^0 = \{i \mid \theta(i,\lambda^*,\mu^*) = 0\}$, and $\mathbf{N}_u^- = \{i \mid \theta(i,\lambda^*,\mu^*) < 0\}$. From (26), clearly, $s_i = 1$ for $i \in \mathbf{N}_u^+$; and $s_i = 0$ for $i \in \mathbf{N}_u^-$.

For the evaluation of the upper bound of **P1**, among all the devices $i \in \mathbf{N}_u^0 \neq \emptyset$ and $\theta(i,\lambda^*,\mu^*) = 0$, one of the devices $r^0 = \arg\max_{r^0\in\mathbf{N}_u^0}\{s_{r^0}E_{r^0}^s\}$ with $s_{r^0} = (\tilde{f}_0 - \sum_{j\in\mathbf{N}_u^+} f_j^{\min})/f_{r^0}^{\min} \in (0,1)$ can be selected to optimize **P2**. This is the case where some resources remain available at the MEC server after all devices in $\mathbf{N}_u^+$ are accepted for offloading, but the remaining resources cannot satisfy in whole any unsatisfied offloading request (from the devices in $\mathbf{N}_u^0$). Given the resource, the device which can save the most energy among the unselected devices, i.e., device r^0, is selected to offload part of its task, thereby maximizing the total energy saving in the absence of the binary constraint on $s_j \in \{0,1\}$. This provides the upper bound for **P1**, as given by

$$e^{\text{LP}} = \sum\nolimits_{i\in\mathbf{N}_u^+} E_i^s + s_{r^0}E_{r^0}^s. \tag{28}$$

For the evaluation of the lower bound of **P1**, $s_i = 0$ can be taken for any device $i \in \mathbf{N}_u^0 \neq \emptyset$, and hence the lower bound is given by $e_f = \sum_{i\in\mathbf{N}_u^+} E_i^s$. This is the case where, after the devices in $\mathbf{N}_u^+$ are admitted, the remaining available resources are just wasted. Since this solution is integer but not optimized, it can save no more energy than the optimal solution to **P1**, and hence provides the lower bound.

The relationship among the lower bound, e_f, the optimal solution to **P1**, e^{opt}, and the LP upper bound, e^{LP}, is revealed in the following Lemma.

Lemma 1. $e_f \leq e^{\text{opt}} \leq e^{\text{LP}} \leq 2e_f$.

Proof. Note that e_f is a lower bound for **P1**, while the LP relaxation provides the upper bound e^{LP}. We can obtain that $e_f \leq e^{\text{opt}} \leq e^{\text{LP}}$. Besides, $e^{\text{LP}} = \sum_{i\in\mathbf{N}_u^+} E_i^s + s_{r^0}E_{r^0}^s \leq 2\max\{\sum_{i\in\mathbf{N}_u^+} E_i^s, E_{r^0}^s\} = 2e_f$. Then, we prove $e_f \leq e^{\text{opt}} \leq e^{\text{LP}} \leq 2e_f$. □

Following Lemma 1, the quantization interval δ can be adjusted to achieve $(1-\varepsilon)$-approximation of the optimum e^{opt} for any $\varepsilon > 0$, as dictated in the following Lemma.

Lemma 2. *The proposed quantized DP algorithm can achieve $(1-\varepsilon)$-approximation of the optimum for any $\varepsilon > 0$, by setting the quantization interval $\delta = e_f\varepsilon/\tilde{K}$.*

Proof. Let $\mathbf{s}_u^{\text{opt}}$ and $\mathbf{s}_u^*$ denote the optimal admission decisions for **P1** and the decision obtained by the proposed quantized DP algorithm, respectively. Let $p(\mathbf{s}) = \sum_i s_iE_i^s$ and $q(\mathbf{s}) = \sum_i s_ie_i^s$ denote the original and quantized energy saving of the admission decision $\mathbf{s}$, respectively. Then, the optimal energy saving can be given by $e^{\text{opt}} = p(\mathbf{s}_u^{\text{opt}})$ and the solution of EROS is $\overline{e^*} = p(\mathbf{s}_u^*)$.

According to (19), we have

$$\delta(e_i^s - 1) \leq E_i^s < \delta e_i^s. \tag{29}$$

Hence, we can obtain that $p(\mathbf{s}_u^{\rm opt}) < \delta q(\mathbf{s}_u^{\rm opt})$ and $p(\mathbf{s}_u^*) \geq \delta[q(\mathbf{s}_u^*) - |\mathbf{s}_u^*|]$, and therefore, we have

$$e^{\rm opt} - \overline{e^*} = p(\mathbf{s}_u^{\rm opt}) - p(\mathbf{s}_u^*) < \delta[q(\mathbf{s}_u^{\rm opt}) + |\mathbf{s}_u^*| - q(\mathbf{s}_u^*)]. \quad (30)$$

Note that $q(\mathbf{s}_u^*) \geq q(\mathbf{s}_u^{\rm opt})$ holds, since the proposed EROS produces the optimal solution for the problem after energy quantization. Thus, we also obtain

$$\delta[q(\mathbf{s}_u^{\rm opt}) + |\mathbf{s}_u^*| - q(\mathbf{s}_u^*)] \leq (e_f \varepsilon / \tilde{K})\, |\mathbf{s}_u^*| \leq e_f \varepsilon \leq e^{\rm opt} \varepsilon. \quad (31)$$

From (30) and (31), we have $e^{\rm opt} - \overline{e^*} < e^{\rm opt}\varepsilon$. Hence, for any $\varepsilon > 0$, the proposed quantized DP algorithm can achieve $(1-\varepsilon)$-approximation of the optimum, i.e., $\overline{e^*} > (1-\varepsilon)e^{\rm opt}$. □

VI. Discussions and Extensions

In the proposed EROS, devices send offloading requests to the MEC server, and offload their tasks according to the result of task admission. Signalling is streamlined, and reduced. In this section, we analyze the complexity, discuss fairness and task partitioning of EROS, extend the proposed EROS to frequency-selective subchannels, and illustrate the impact of inter-cell interference.

A. Complexity Analysis

The following Lemma exhibits the tradeoff between the performance and time-complexity of the proposed EROS.

Lemma 3. *EROS is able to achieve $(1-\varepsilon)$-approximation of the optimum at the complexity of $\mathcal{O}(NK^2/\varepsilon)$.*

Proof. Recall that N_u and $\tilde{K}$ denote the numbers of remaining devices and subchannels after pre-admission by Proposition 1, respectively. The time-complexity of EROS depends on the number of subproblems to be solved. As mentioned in Section IV-B, the number of subproblems is $N_u \tilde{K} \widehat{e}$, where the time-complexity for each subproblem using (17) is $\mathcal{O}(1)$. The time-complexity of backward induction is $\mathcal{O}(N_u)$ [28]. Thus, the overall time-complexity of EROS is $\mathcal{O}(N_u \tilde{K} \widehat{e})$.

We can further tighten the upper bound of quantized energy saving in (20) by replacing $\overline{E}$ with the LP upper bound, $e^{\rm LP}$, as given by $\widehat{e} = \lceil e^{\rm LP}/\delta \rceil + \tilde{K}$. Therefore, we have

$$\mathcal{O}(N_u \tilde{K} \widehat{e}) = \mathcal{O}(N_u \tilde{K}^2 + N_u \tilde{K} e^{\rm LP}/\delta). \quad (32)$$

From Lemma 2, we show that $(1-\varepsilon)$-approximation of the optimum can be achieved by using $\delta = e_f \varepsilon/\tilde{K}$. By substituting this into (32), the time-complexity of EROS is $\mathcal{O}(N_u \tilde{K}^2 + (e^{\rm LP}/e_f) N_u \tilde{K}^2/\varepsilon) = \mathcal{O}(N_u \tilde{K}^2 + N_u \tilde{K}^2/\varepsilon) = \mathcal{O}(N_u \tilde{K}^2/\varepsilon)$. Since the pre-admission using Proposition 1 reduces the number of devices to be assessed, the complexity of EROS is $\mathcal{O}(NK^2/\varepsilon)$. □

Additional measures can be taken to further reduce the complexity and overhead. Upon the receipt of $\tilde{f}_0$, each device in $\mathbf{N}_u$ can check the following condition to determine whether to send offloading requests.

Proposition 3. *If $(T_i^r)_{\min} = T_i^t + C_i/\tilde{f}_0 > T_i^{\rm req}$ or $E_i^r \geq E_i^l$, device i executes its task locally.*

Proposition 3 describes two cases. In the first case, even allocating all the remaining resources $\tilde{f}_0$ at the MEC server to device i cannot satisfy the task deadline, i.e., $(T_i^r)_{\min} = T_i^t + C_i/\tilde{f}_0 > T_i^{\rm req}$. In the second case, offloading would not save energy for the device, i.e., $E_i^r \geq E_i^l$. In both cases, the device is pre-denied and chooses to execute its task locally.

Based on Propositions 1 and 3, signalling can be reduced and streamlined while the asymptotic optimality of the system is preserved. Particularly, devices with limited resources and tight deadlines are given priority to send offloading requests to, and get satisfied by, the MEC server; see Proposition 1. The MEC server then broadcasts the remaining resources, based on which devices can spontaneously decide to process tasks locally and withhold requests, if the remaining resources is insufficient to meet their deadlines; see Proposition 3. Only the rest of the devices send offloading requests to the MEC server which (i.e. the server) conducts the proposed quantized DP algorithm to admit devices and allocate resources accordingly.

Lemma 3 dictates an $[\mathcal{O}(\varepsilon), \mathcal{O}(1/\varepsilon)]$-tradeoff between the optimality loss and time-complexity of the proposed quantized DP algorithm running at the MEC server. This gives the MEC server an opportunity to reduce the energy consumption of the network by leveraging its hardware capability. In practice, an MEC server can choose the smallest ϵ value based on its capability, thereby attaining the minimum achievable energy consumption of the system.

B. Fairness and Task Partitioning

The proposed approach can be readily extended to provide fairness between devices in terms of energy saving in the long term. By exploiting the idea of proportional fairness [31], a coefficient $\frac{1}{\kappa_i}$ can be multiplied to the energy consumption of each device in the objective of **P**, i.e., $\min_{\mathbf{s},\mathbf{f}} \sum_{i \in \mathbf{N}} \frac{1}{\kappa_i}[s_i E_i^r + (1-s_i)E_i^l]$. κ_i is the time-average of the past energy saving of device i. Device i with small κ_i is given priority to offload tasks, achieving fairness in the long term. At every instant, the optimal substructure of Bellman equation can be preserved, and the reformulated problem can be readily solved by using the proposed quantized DP algorithm.

In a different context of task partitioning, a task can be partitioned into atomic subtasks (e.g., in a tree structure). Some of the atomic subtasks can be offloaded, and offloading needs to be holistically designed to ensure the consistency (i.e., the correct order) of task processing. However, existing studies, such as [7]–[11], implicitly assumed unlimited computational and transmission capabilities, and did not schedule between multiple devices. To this end, there is no comparable algorithm to the algorithm proposed in this paper.

In our earlier work [32], we partitioned a delay-sensitive task of a single device to be processed against limited resources in the most energy-efficient manner. This has potential to be implemented in conjunction with the proposed algorithm to support task partitioning. Particularly, each device can partition its own task into atomic subtasks, and specify the deadlines of the subtasks. By using the partitioning technique in [32], the deadlines can be designed to guarantee the integrity of task processing, given available resources. The proposed quantized DP algorithm can be used to asymptotically optimally schedule the atomic subtasks and assign resources.

Given the inherent independence between task partitioning and scheduling in this design, we can still separately focus on the scheduling approach developed in this paper. More closely coupled designs of task partitioning and scheduling are nontrivial, have yet to be developed, and will be our future work.

C. Channel-aware Scheduling

The proposed algorithm has the potential to be extended to more complex scenarios, where the channels are frequency-selective, instantaneously measured at the devices, fed back to the BS, and selected for different devices to further reduce energy consumption.

Define $s_i \in \{0, \cdots, K\}$ as such that device i locally executes its task if $s_i = 0$, or offload the task to the MEC server via subchannel s_i. The constraints C1 and C2 of problem **P** can be accordingly reformulated as

$$\begin{aligned} &\text{C1': } s_i \in \{0, \cdots, K\}, \forall i, \\ &\text{C2': } s_i \neq s_j, \forall s_i \neq 0, \; i \neq j; \end{aligned} \tag{33}$$

where C2' indicates that a subchannel must be exclusively occupied by a single device at a time.

The changes of the constraints do not affect Proposition 1, where the resource-restrained devices $\mathbf{N}_r$ are pre-admitted for offloading. We note that the minimum computational resources (i.e., $f_i^{\min}$) in Proposition 2 now depend on the subchannel that device i is allocated, and needs to be updated to $f_{i,l}^{\min}$, i.e., the minimum computational resources of device i using the specific subchannel l:

$$f_{i,l}^{\min} = C_i / (T_i^{\text{req}} - D_i / R_i^l), \tag{34}$$

where R_i^l is the transmit rate of device i at subchannel l. C5 in problem **P1** can be updated by

$$\text{C5': } \sum\nolimits_{i \in \mathbf{N}} s_i f_{i,s_i}^{\min} \leq f_0. \tag{35}$$

Unlike **P1**, the channel allocation for the pre-admitted devices $\mathbf{N}_r$ are now coupled with the admission and channel allocation for the remaining undetermined devices $\mathbf{N}_u$. Given frequency-selective channels, the problem of interest **P3** becomes

$$\begin{aligned} \textbf{P3: } &\max_{\mathbf{s}} \sum\nolimits_{i \in \mathbf{N}} s_i E_{i,s_i}^s, \\ &\text{s.t. } s_i \neq 0, \; i \in \mathbf{N}_r, \\ &\quad \text{C1', C2' and C5'}, \end{aligned} \tag{36}$$

where E_{i,s_i}^s denotes the energy saving of device i by offloading its task through subchannel s_i. The channels of the pre-admitted devices in $\mathbf{N}_r$ are selected together with the other devices, and the admission of $\mathbf{N}_r$ is guaranteed by enforcing the constraint $s_i \neq 0, \; i \in \mathbf{N}_r$.

Let $\mathbf{O} = \{o_1, \cdots, o_K\}$ stand for the channel occupation status of the K subchannels and $o_i \in \{0, 1\}$, i.e., subchannel i is unoccupied if $o_i = 0$, or occupied otherwise. Define $\phi_i(e, \mathbf{O})$ as the minimum computational resources for the first i devices while e units of energy is saved and the subchannel status is $\mathbf{O}$. We notice that the optimal substructure still holds by replacing the original subproblem $\phi_i(e, l)$ with $\phi_i(e, \mathbf{O})$ in problem **P3**.

According to Bellman equation [27], $\phi_i(e, \mathbf{O})$ can be presented in a recurrence expression based on the results of the preceding subproblems $\phi_{i-1}(e, \mathbf{O})$, as given by

$$\phi_i(e, \mathbf{O}) = \min \left\{ \begin{array}{l} \phi_{i-1}(e, \mathbf{O}), \text{ if } i \notin \mathbf{N}_r \\ \phi_{i-1}(e - E_{i,l}^s, \mathbf{O}(l) = 0) + f_{i,l}^{\min}; \forall o_l = 1 \end{array} \right\}, \tag{37}$$

where $\mathbf{O}(l) = 0$ stands for the vector $\mathbf{O}$ with $o_l = 0$. Also, the channel allocation can be recorded by

$$s_i(e, \mathbf{O}) = \left\{ \begin{array}{l} l, \text{ if } \phi_{i-1}(e, \mathbf{O}) = \phi_{i-1}(e - E_{i,l}^s, \mathbf{O}(l) = 0) + f_{i,l}^{\min}; \\ 0, \text{ otherwise.} \end{array} \right. \tag{38}$$

The maximum energy saving is $e^* = \delta \max_l \{e \mid \phi_N(e, \mathbf{O}) \leq f_0\}$, and the channel-aware scheduling can be obtained also through backward induction, as done in Algorithm 1.

However, signalling overhead would grow under the channel-aware scheduling, as all the devices need to feed back their instantaneous channels so that the BS can schedule the devices in the channel-aware manner. Moreover, the computational complexity would also grow, since the channel selection is integer programming coupled with the device selection in a multiplicative manner in the proposed algorithm. The number of subproblems grows from $NK\widehat{e}$ subproblems $\phi_i(e, l)$ in frequency-flat channels to $N2^K\widehat{e}$ subproblems $\phi_i(e, \mathbf{O})$ in frequency-selective channels. Consequently, the time-complexity grows from $\mathcal{O}(NK^2/\epsilon)$ to $\mathcal{O}(NK2^K/\epsilon)$.

D. The Impact of Inter-cell Interference

The proposed asymptotically optimal approach can be applied in the presence of inter-cell interference. Interference coordination and fractional frequency reuse are typically adopted for inter-cell interference mitigation [33]. Based on the Central Limit Theorem, the residual inter-cell interference can be modeled to be Gaussian, given a typically large number of transmitters beyond the cell of interest [34] and [35]. (4) can be rewritten as $R_i^l = W \log_2(1 + \frac{p_i h_i^l}{N_0 + N_I^l})$ to account for the Gaussian interference per subchannel, where R_i^l and h_i^l are the achievable data rate and the channel gain of device i in subchannel l, and N_I^l is the interference power measured at the BS in the subchannel. In the case of a flat-fading channel with i.i.d. Gaussian interference per subchannel, the resultant problem involving R_i^l (not R_i) fully complies with the original interference-free setting, and can be solved by running the proposed algorithm, i.e., Algorithm 1. In the case of a frequency-selective channel with independent and non-identically distributed (i.n.d.) Gaussian interference per subchannel, the resultant problem fully complies with the channel-aware variation of the original interference-free setting, and can be also readily solved by using the proposed approach in the same way as discussed in Section VI-C.

VII. Simulation Results

In this section, Monte Carlo simulations are carried out to evaluate the proposed algorithm. The number of subchannels is $K = 20$, and the bandwidth per subchannel is $W = 180$kHz. The total number of devices N is up to 25; unless otherwise specified. These devices are uniformly distributed in a cell

TABLE I
SIMULATION PARAMETERS

Parameters	Assumptions
Macrocell Radius	250m
Number of Subchannels	20
Bandwidth per Subchannel	180kHz
Pathloss from Device to Macro BS	$128.1 + 37.5\log_{10}(r)$
Thermal Noise Density	-174dBm/Hz
Transmit Power	23dBm
Lognormal Shadowing Standard Deviation	10dB
Input Data Size D	85kB
Required Number of CPU Cycles C	1000Million cycles
Local Computational Capability F_i^l	0.5GHz−1.5GHz
Latency Request $T^{\rm req}$	$\{1, 1.5\}$s
Remote Computational Capability f_0	15GHz

with radius of 250m. Consider complex applications like face recognition [25], of which the input data size is $D = 85$kB and the required number of CPU cycles is $C = 1000$ million cycles. The local computational capability F_i^l is uniformly distributed within $[0.5, 1.5]$ GHz. Other parameters used in the simulations are summarized in Table I [36]. 5000 independent runs are conducted for each data point.

Apart from the proposed task admission algorithm (i.e., EROS), we also simulate the following algorithms for comparison purpose.

1) **Branch and Bound (B&B):** After reformulating the MIP problem P to the IP problem **P1**, we take the B&B algorithm [37] to achieve the optimal solutions for **P**. B&B is a classical and popular solver for discrete and combinatorial optimization problems, conducting a structured enumeration of candidate solutions following a tree structure [37]. In the case of **P1**, the B&B method can be set up to assess s_j ($j = 1, \cdots, N_u$), following a binary tree. When assessing a particular s_j, the upper and lower bounds of **P1** are achieved by taking the binary values of s_i, $i = 1, \cdots, j$, from each of the remaining branches, relaxing s_k to be continuous within $[0, 1]$ for $k = j + 1, \cdots, N_u$, and solving the relaxed problems with linear programming. The branches with upper bounds lower than the lower bound of some other branches are removed – bounding, since those branches offer no prospect of the optimal solution. By this means, the optimal solution can be achieved by enumerating a potentially reduced number of candidate solutions. Unfortunately, in the worst-case scenario where no branches can be removed until the final stage of assessing the last layer, all candidate solutions are enumerated from the root along every branch of the binary tree. As a consequence, the worst-case complexity of the B&B method can be as high as exhaustive search.
2) **All Request Admission Algorithm (ARAA):** The MEC server accepts all the offloading requests, and computational resources are equally allocated. In the case that the wireless bandwidth is insufficient to admit all the devices, K out of the devices are randomly admitted for remote processing.
3) **Local execution (Local):** There is no offloading. All tasks are executed locally.

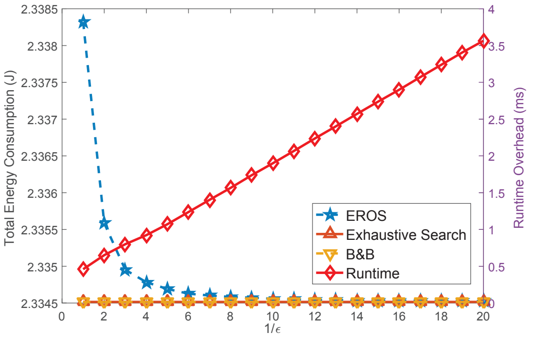


Fig. 2. The $[\mathcal{O}(\varepsilon), \mathcal{O}(1/\varepsilon)]$-tradeoff between the optimality loss and time-complexity.

We have also simulated the relaxed version of EROS which allows partial offloading by dropping the binary constraint C1, where a task can be arbitrarily partitioned; see [4]. This is due to the fact that there is no other comparable partial offloading approach, as mentioned in Section VI-B. Unfortunately, the relaxed algorithm supporting partial offloading, referred to as "Partial", can violate the integrity of tasks that cannot be partitioned in many cases. Moreover, the relaxed algorithm is less challenging than EROS. Without the binary constraint C1, partial offloading can be straightforwardly implemented by using the standard Matlab linear programming toolbox. The energy saving of partial offloading can also be substantially overestimated, especially in the case of large tasks, tight deadlines, or limited resources, as will be shown in Figs. 3 and 4.

Fig. 2 shows the tradeoff between the optimality loss and time-complexity of the proposed EROS, as $1/\varepsilon$ varies from 1 to 20. Here, $N = 20$ and $T^{\rm req} = 1$s. We see that the time-complexity of EROS grows linearly with $1/\varepsilon$, while the total energy consumption of devices decreases with $1/\varepsilon$ and approaches to the optimum achieved by exhaustive search. The $[\mathcal{O}(\varepsilon), \mathcal{O}(1/\varepsilon)]$-tradeoff is confirmed, as stated in Lemma 3. B&B achieves the minimum energy consumption, as can be validated by comparing to the results of exhaustive search. EROS achieves nearly the minimum energy consumption achieved by B&B when $1/\varepsilon \geq 10$. For this reason, in the following simulations, we set $(1-\varepsilon) = 0.9$, i.e., $1/\varepsilon = 10$. It is worth mentioning that both 2.3385J and 2.3345J are the energy consumptions under different values of $\epsilon = 1$ and $\epsilon = 0.05$. Their difference is not energy saving.

Fig. 3 shows the average energy consumptions of EROS, Partial, B&B, ARAA and Local, where $N = 20$ and $T^{\rm req}$ ranges from 1s to 3s. The energy savings of the proposed EROS compared with the existing solutions are evaluated. With the growth of $T^{\rm req}$, the energy consumptions of EROS, Partial, and B&B first decrease and then stabilize at 0.075 Joule per device when $T^{\rm req} \geq 2$s. This is because increasingly relaxed deadlines allow a growing number of devices to be selected for offloading, thereby increasingly exploiting diversity

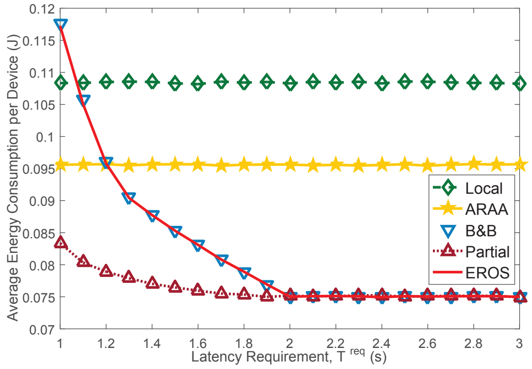


Fig. 3. The comparison of devices' average energy consumption as $T^{\rm req}$ increases from 1s to 3s, where $N = 20$.

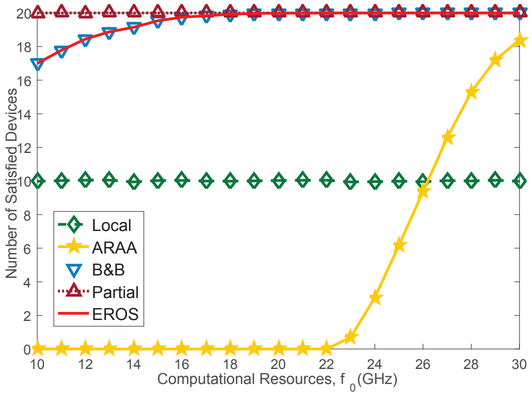
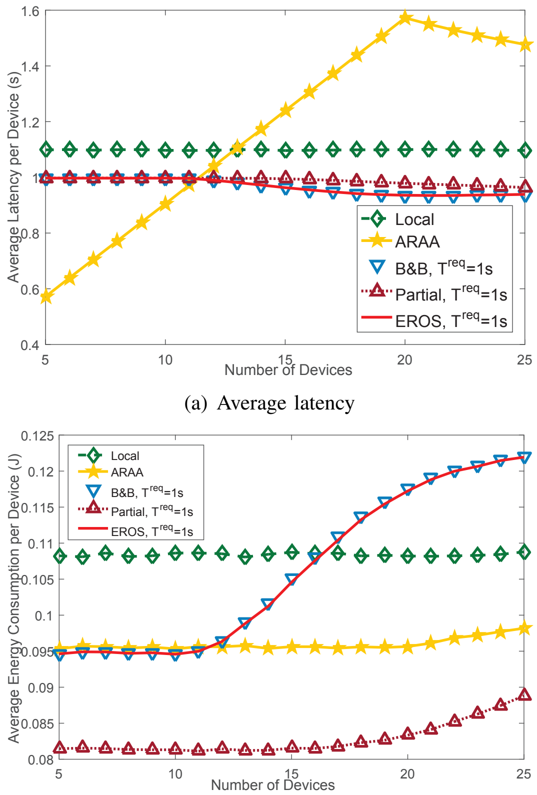


Fig. 4. The comparison of the number of satisfied devices as f_0 increases from 10GHz to 30GHz, where $N = 20$ and $T^{\rm req} = 1$s.



(a) Average latency

(b) Average energy consumption

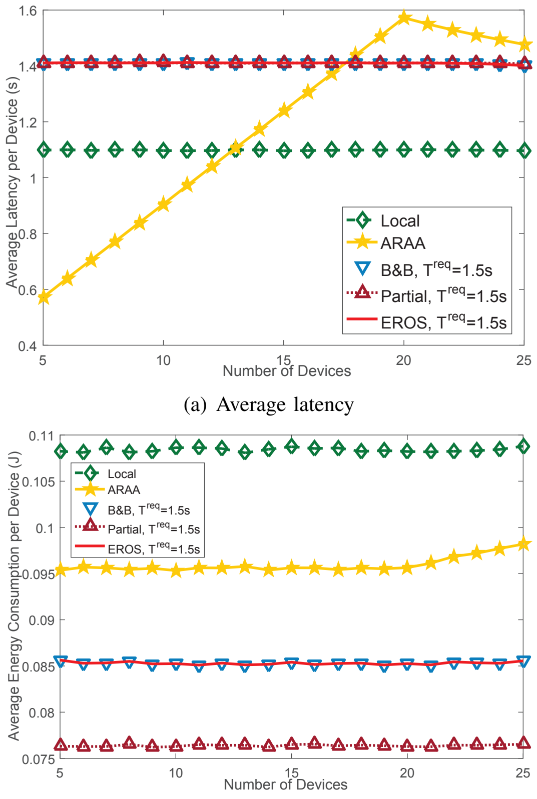
Fig. 5. The comparison of devices' average latency and energy consumption where $T^{\rm req} = 1$s. EROS and B&B increase energy consumptions to satisfy the stringent deadline.

and reducing energy consumption. EROS and B&B can save up to 31% of energy as compared with Local. Compared with EROS and B&B, Partial could dramatically save 28% more energy, since it allows part of the most energy-efficient tasks to be offloaded, but violates the integrity of the tasks. When the deadlines are so loose that almost all devices can be selected for offloading, the energy consumption stops decreasing and stabilizes. The loose deadlines can also help eliminate the difference between atomic tasks and partial offloading, since tasks can be offloaded in whole and integrity does not degrade.

Fig. 4 shows the number of devices with satisfied deadlines achieved by EROS, Partial, B&B, ARAA and Local, as f_0 increases from 10GHz to 30GHz, where $N = 20$ and $T^{\rm req} = 1$s. EROS and B&B can satisfy the deadlines of all devices when $f_0 \geq 17$ GHz. However, when $f_0 < 17$GHz, the numbers of devices satisfied by EROS and B&B slightly drop to 17 as f_0 decreases to 10GHz. This is the infeasible scenario of (15) due to the insufficient computational and transmission resources. On the other hand, Partial can satisfy all deadlines irrespective of f_0, by overlooking task integrity and continuously leveraging both local and remote computational resources. In contrast, ARAA cannot satisfy any deadline when $f_0 \leq 22$GHz, and satisfies only up to 18 devices when $f_0 = 30$GHz. Local can always satisfy only half of the deadlines due to the uniform distribution of local computational capacity.

Fig. 5 compares the average latency and energy consumption between EROS, Partial, B&B, ARAA and Local, where $T^{\rm req} = 1$s. This is the case where the deadlines are stringent, and effective task admission is critical to meet the deadlines by leveraging local and remote computational capabilities. We see in Fig. 5 that EROS, Partial, and B&B can either save energy or reduce latency, compared with ARAA and Local. In Fig. 5(a), EROS and B&B provide nearly identical latency, both satisfying the deadline. Partial can also meet the deadline. The other algorithms all violate the deadline $T^{\rm req} = 1$. The average latency of ARAA grows linearly with N when N is small to medium (i.e., $N \leq 20$); and declines when N is large (i.e., $N > 20$), as the result of the increasing number of devices executing tasks locally with the average latency of 1.1s (c.p., 1.6s for remote processing). ARAA also violates $T^{\rm req} = 1$ for $N > 13$.

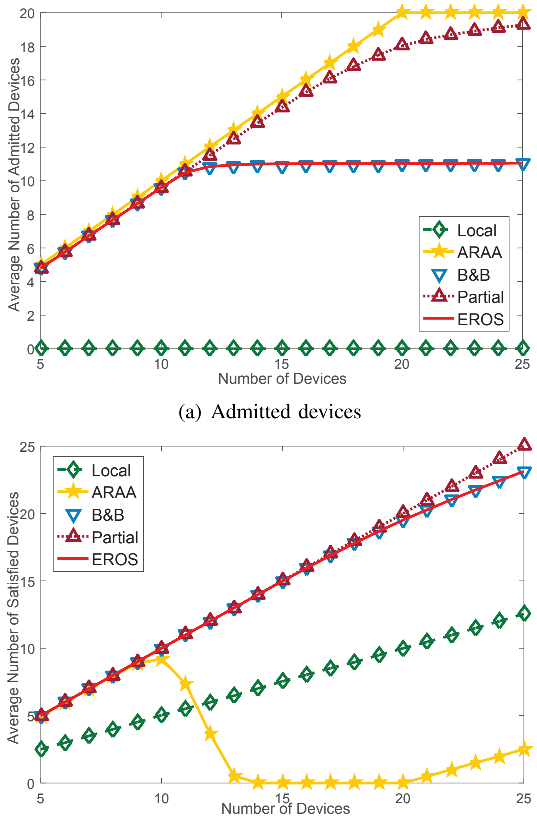
In Fig. 5(b), we see that the proposed EROS and B&B provide indistinguishably close energy consumptions, validating the asymptotic optimality of EROS. We also see the energy consumptions of EROS and B&B are low and stable when N is small, i.e., $N < 12$, and grow as N increases from 12 to 20. The growths of energy consumptions slow down, when N is large, i.e., $N > 20$. This is because EROS and B&B exploit the increasing diversity pertaining to the growing number of devices, thereby saving more energy. As shown in Fig. 3, Partial could substantially save energy by breaching task integrity, but would undergo the increase of energy consumption when $N \geq 17$ as in EROS and B&B.



(a) Average latency

(b) Average energy consumption

Fig. 6. The comparison of devices' average latency and energy consumption where $T^{\text{req}} = 1.5$s. EROS and B&B can substantially save energy.



(a) Admitted devices

(b) Satisfied devices

Fig. 7. The comparison of the numbers of admitted and satisfied devices where $T^{\text{req}} = 1$s. Without task admission, ARAA satisfies no task deadlines.

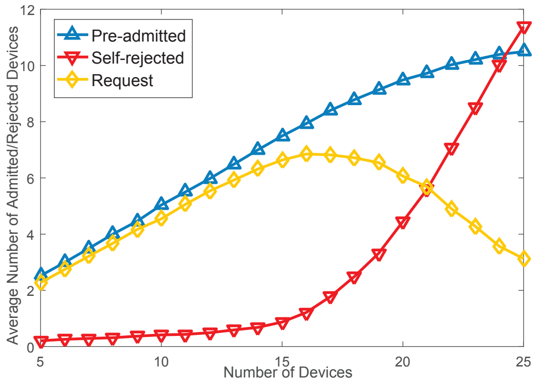


Fig. 8. The comparison of the numbers of pre-admitted and pre-denied devices where $T^{\text{req}} = 1$s. The number of devices to be assessed is consistent with the result in Fig. 9.

In contrast, the energy consumptions of ARAA and Local are relatively stable, and can be lower than that of EROS and B&B when N is large, at the cost of unsatisfied deadlines, as discussed in Fig. 5(a).

Fig. 6 plots the average latency and energy consumption of the algorithms where $T^{\text{req}} = 1.5$s. This is the case where the deadlines are relatively loose. We can see that EROS, Partial, and B&B can substantially save energy consumptions given the loose deadline. Their energy consumptions are much lower than ARAA and Local, since ARAA and Local admit devices independently of deadlines.

Fig. 7 shows the numbers of admitted and satisfied devices under EROS, Partial, B&B, ARAA and Local, where $T^{\text{req}} = 1$s. We see that B&B and EROS admit at most 11 devices for offloading, while satisfying the deadlines of almost all tasks. This is because the limited resources at the MEC server cannot accommodate more tasks concurrently under the stringent deadlines. Partial can admit more devices for offloading than EROS and B&B, and satisfy all deadlines, but the number of offloaded devices is still less than $K = 20$. This is because partial offloading must not violate the physical constraint of K subchannels. In contrast, Local does not admit any devices, as shown in Fig. 7(a). The number of satisfied devices is proportional to the probability that a device can satisfy its deadline locally, and therefore grows linearly with the number of devices, as shown in Fig. 7(b). ARAA always admits as many devices as possible without consideration on deadlines, as shown in Fig. 7(a). As a consequence, the satisfaction of the devices degrades rapidly, as shown in Fig. 7(b). Only in the case that $N > 20$, a small number of devices that are not admitted may satisfy their deadlines through local processing. The increase of such devices is proportional to the probability that a device can satisfy its deadline locally, and therefore yields the same slope as Local in Fig. 7(a).

Fig. 8 plots the numbers of the devices that can be pre-admitted, the devices that spontaneously decide to process tasks locally and not to send offloading requests, and the devices that send requests to be selected for offloading, as N increases. The total number of the three types of device is N. It

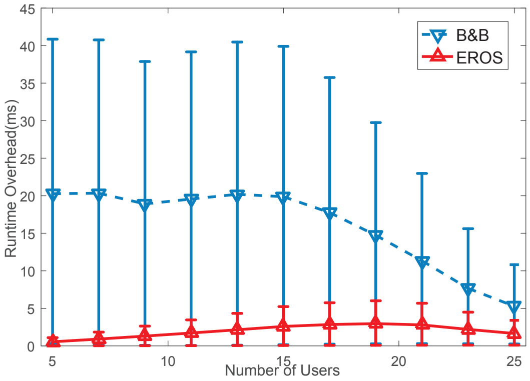


Fig. 9. The comparison of runtime between EROS and B&B where $T^{\rm req} = 1$s. EROS is superior in terms of efficiency and stability.

is worth pointing out that the number of subchannels K limits the number of devices concurrently offloading, not the number of devices in total. As shown in the figure, both the numbers of the devices pre-admitted and the devices withholding requests increase with N, while the number of the devices that request to be scheduled first increases and then declines. This is because, by pre-admitting any resource-restrained devices and pre-rejecting those which can neither save energy nor meet deadlines by offloading, the number of devices that need to feed back channels and task information can be substantially reduced. The feedbacks can even reduce with the growth of N, since the number of resource-restrained devices that need to be pre-admitted for offloading grows, hence increasingly draining the available computational resources and stopping devices from offloading and feeding back.

Fig. 9 compares runtime between B&B and EROS, where $T^{\rm req} = 1$s and the confidence interval is 95%. We can see that the proposed EROS is superior in terms of efficiency and stability. Both the average and variance of the runtime of EROS are substantially lower than those of B&B, respectively. In contrast, B&B is neither computationally scalable nor reliable, and provides limited value in practice. We also see the concavity of the average runtime, i.e., the runtimes of EROS and B&B first increase and then decline as N grows, as can be evidenced by Fig. 8.

VIII. Conclusion

In this paper, we formulated task admission and resource allocation to minimize the total energy consumption of MEC while guaranteeing the latency requirements of devices. This problem was reformulated as an integer programming problem by pre-admitting resource-restrained devices. A quantized DP algorithm was proposed to solve the integer programming problem at a polynomial complexity $\mathcal{O}(NK^2/\varepsilon)$. We also meticulously designed the quantization interval of energy saving to achieve the asymptotic optimality of the proposed algorithm with an $[\mathcal{O}(\varepsilon), \mathcal{O}(1/\varepsilon)]$-tradeoff between the optimality loss and time-complexity. Simulation results corroborate that, superior in efficiency and stability, the proposed scheme is able to save energy indistinguishably close to the maximum energy saving.

References

[1] 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 *IEEE Symp. on Computers and Communications (ISCC)*, 2012, pp. 59–66.

[2] W. Zhang, Y. Wen, J. Wu, and H. Li, "Toward a unified elastic computing platform for smartphones with cloud support," *IEEE Netw.*, vol. 27, no. 5, pp. 34–40, 2013.

[3] X. Lyu, W. Ni, H. Tian, R. P. Liu, X. Wang, G. B. Giannakis, and A. Paulraj, "Optimal schedule of mobile edge computing for internet of things using partial information," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2606–2615, Nov 2017.

[4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[5] "Mobile-edge computing introductory technical white paper," *White Paper, Mobile-edge Computing (MEC) industry initiative*, 2014.

[6] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Tech.*, vol. 66, no. 4, pp. 3435–3447, April 2017.

[7] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, September 2013.

[8] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 81–93, Jan 2015.

[9] Y.-H. Kao, B. Krishnamachari, M.-R. Ra, and F. Bai, "Hermes: Latency optimal task assignment for resource-constrained mobile computing," in *2015 IEEE INFOCOM*, April 2015, pp. 1894–1902.

[10] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan, "Odessa: enabling interactive perception applications on mobile devices," in *Mobisys*. ACM, 2011, pp. 43–56.

[11] Z. Cheng, P. Li, J. Wang, and S. Guo, "Just-in-time code offloading for wearable computing," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 1, pp. 74–83, March 2015.

[12] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Trans. on Comput.*, vol. 64, no. 8, pp. 2253–2266, Aug 2015.

[13] V. Cardellini, V. D. N. Personé, V. Di Valerio, F. Facchinei, V. Grassi, F. L. Presti, and V. Piccialli, "A game-theoretic approach to computation offloading in mobile cloud computing," *Mathematical Programming*, pp. 1–29, 2013.

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

[15] 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, October 2016.

[16] 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.

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

[18] K. Wang, K. Yang, and C. Magurawalage, "Joint energy minimization and resource allocation in c-ran with mobile cloud," *IEEE Trans. Cloud Comput.*, 2017.

[19] W. Labidi, M. Sarkiss, and M. Kamoun, "Joint multi-user resource scheduling and computation offloading in small cell networks," in *2015 IEEE WiMob*, Oct 2015, pp. 794–801.

[20] M. H. Chen, M. Dong, and B. Liang, "Joint offloading decision and resource allocation for mobile cloud with computing access point," in *2016 IEEE ICASSP*, March 2016, pp. 3516–3520.

[21] J. Cheng, Y. Shi, B. Bai, and W. Chen, "Computation offloading in cloud-ran based mobile cloud computing system," in *2016 IEEE ICC*, May 2016, pp. 1–6.

[22] J. A. Bondy, *Graph Theory With Applications*. Oxford, UK, UK: Elsevier Science Ltd., 1976.

[23] X. Lin, Y. Wang, Q. Xie, and M. Pedram, "Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment," *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 175–186, March 2015.

[24] "Evolved universal terrestrial radio access (e-utra); user equipment (ue) procedures in idle mode," 3GPP TS 36.304 V14.2.0, Tech. Rep., 2017.

[25] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: making smartphones last longer with code offload," in *Mobisys*. ACM, 2010, pp. 49–62.

[26] Y. Pochet and L. A. Wolsey, *Production planning by mixed integer programming*. Springer Science & Business Media, 2006.

[27] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[28] T. H. Cormen, *Introduction to algorithms*. MIT press, 2009.

[29] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[30] N. Megiddo, "Linear programming in linear time when the dimension is fixed," *J. ACM*, vol. 31, no. 1, pp. 114–127, Jan. 1984.

[31] T. B. L. E. L. R. Ramjee, "Generalized proportional fair scheduling in third generation wireless data networks," in *2006 IEEE INFOCOM*, 2006.

[32] X. Lyu and H. Tian, "Adaptive receding horizon offloading strategy under dynamic environment," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 878–881, May 2016.

[33] Y. L. Lee, T. C. Chuah, J. Loo, and A. Vinel, "Recent advances in radio resource management for heterogeneous lte/lte-a networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2142–2180, Fourthquarter 2014.

[34] H. V. Poor and S. Verdu, "Probability of error in mmse multiuser detection," *IEEE Trans. Inf. Theory*, vol. 43, no. 3, pp. 858–871, May 1997.

[35] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[36] E. U. T. R. Access, "Further advancements for E-UTRA physical layer aspects," 3GPP TR 36.814, Tech. Rep., 2010.

[37] R. S. Garfinkel and G. L. Nemhauser, *Integer programming*, vol. 4.

Xinchen Lyu received the B.E. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2014. He is currently pursuing the cotutelle Ph.D. degree at BUPT and University of Technology Sydney. His research interests include mobile edge computing and radio resource management.

Hui Tian received her M.S. in Micro-electronics and Ph. D degree in circuits and systems both from Beijing University of Posts and Telecommunications, China, in 1992 and 2003, respectively. Currently, she is a professor at BUPT, the Network Information Processing Research Center director of State Key Laboratory of Networking and Switching Technology and the MAT director of Wireless Technology Innovation Institute (WTI). Her current research interests mainly include radio resource management, cross layer optimization, M2M, cooperative communication, mobile social network, and mobile edge computing.

Wei Ni received the B.E. and Ph.D. degrees in Electronic Engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. Currently he is a Senior Scientist, and Team and Project Leader at CSIRO, Australia. He also holds honorary positions at the University of New South Wales (UNSW), Macquarie University (MQ) and the University of Technology Sydney (UTS). Prior to this he was a postdoctoral research fellow at Shanghai Jiaotong University (2005-2008), Research Scientist and Deputy Project Manager at the Bell Labs R&I Center, Alcatel/Alcatel-Lucent (2005-2008), and Senior Researcher at Devices R&D, Nokia (2008-2009). His research interests include optimization, game theory, graph theory, as well as their applications to network and security.

Dr Ni serves as Editor for Hindawi Journal of Engineering since 2012, secretary of IEEE NSW VTS Chapter since 2015, Track Chair for VTC-Spring 2016 and 2017, and Publication Chair for BodyNet 2015. He also served as Student Travel Grant Chair for WPMC 2014, Program Committee Member of CHINACOM 2014, TPC member of IEEE ICC'14, ICCC'15, EICE'14, and WCNC'10.

Yan Zhang is currently full Professor at the Department of Informatics, University of Oslo, Norway. He received a Ph.D. degree in School of Electrical & Electronics Engineering, Nanyang Technological University, Singapore. He is an Associate Technical Editor of IEEE Communications Magazine, an Editor of IEEE Transactions on Green Communications and Networking, IEEE Communications Surveys & Tutorials, and the IEEE Internet of Things Journal, and an Associate Editor of IEEE Access. He has served as Chair for a number of conferences, including IEEE GLOBECOM 2017, IEEE VTC-Spring 2017, IEEE PIMRC 2016, IEEE CloudCom 2016, IEEE ICCC 2016, IEEE CCNC 2016, IEEE SmartGridComm 2015, and IEEE CloudCom 2015. He has served as a TPC member for numerous international conference including IEEE INFOCOM, IEEE ICC, IEEE GLOBECOM, and IEEE WCNC. His current research interests include next-generation wireless networks leading to 5G, and green and secure cyber-physical systems (e.g., smart grid, healthcare, and transport). He is an IEEE Vehicular Technology Society Distinguished Lecturer. He is also a Senior Member of IEEE ComSoc, IEEE CS, IEEE PES, and IEEE VT Society. He is a Fellow of IET.

Ping Zhang received the M.S. degree in electrical engineering from Northwestern Polytechnical University, Xian, China, in 1986 and the Ph.D. degree in electric circuits and systems from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1990. He is currently a Professor with BUPT and the Director of the State Key Laboratory of Networking and Switching Technology, China. His research interests include cognitive wireless networks, fourth-generation mobile communication, fifth-generation mobile networks, communications factory test instruments, universal wireless signal detection instruments, and mobile Internet.

Dr. Zhang is the Executive Associate Editor-in-Chief on information sciences of the Chinese Science Bulletin, a Guest Editor of the IEEE Wireless Communications Magazine, and an Editor of China Communications. He received the First and Second Prizes from the National Technology Invention and Technological Progress Awards, as well as the First Prize Outstanding Achievement Award of Scientific Research in College.

Ren Ping Liu is a Professor at the School of Electrical and Data Engineering in University of Technology Sydney, where he leads the Network Security Lab in the Global Big Data Technologies Centre. He is also the Research Program Leader of the Digital Agrifood Technologies in Food Agility CRC, a government/research/industry initiative to empower Australia's food industry through digital transformation. Prior to that he was a Principal Scientist at CSIRO, where he led wireless networking research activities. He specialises in protocol design and modelling, and has delivered networking solutions to a number of government agencies and industry customers. Professor Liu was the winner of Australian Engineering Innovation Award and CSIRO Chairman medal. His research interests include Markov analysis and QoS scheduling in WLAN, VANET, IoT, LTE, 5G, SDN, and network security. Professor Liu has over 100 research publications, and has supervised over 30 PhD students.

Professor Liu is the founding chair of IEEE NSW VTS Chapter and a Senior Member of IEEE. He served as Technical Program Committee chair and Organising Committee chair in a number of IEEE Conferences. Ren Ping Liu received his B.E.(Hon) and M.E. degrees from Beijing University of Posts and Telecommunications, China, and the Ph.D. degree from the University of Newcastle, Australia.